



**Eva Harb, BSc**

## **Group Recommendations for Holiday Destinations**

An application supporting group decision processes to plan common tours  
using recommender technologies deployed on Facebook

### **Master's Thesis**

to achieve the university degree of  
Master of Science  
Master's degree programme: Computer Science

submitted to

**Graz University of Technology**

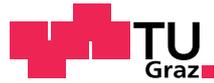
Supervisor

Univ.-Prof. Dipl.-Ing. Dr.techn. Alexander Felfernig  
Institute of Software Technology (IST),  
Graz University of Technology

Graz, May 2015

© Copyright 2015 by Eva Harb





**Eva Harb, BSc**

## **Gruppenempfehlungen für Urlaubsreisen**

Eine Applikation, die Gruppenentscheidungsprozesse für die Planung gemeinsamer Urlaube mittels Recommender-Technologien auf Facebook unterstützt

### **Masterarbeit**

zur Erlangung des akademischen Grades

Diplom-Ingenieurin  
Masterstudium Informatik

eingereicht an der

**Technischen Universität Graz**

Betreuer

Univ.-Prof. Dipl.-Ing. Dr.techn. Alexander Felfernig  
Institut für Softwaretechnologie (IST),  
Technische Universität Graz

Graz, Mai 2015

© Copyright 2015, Eva Harb



## Abstract

Recommender systems for groups have become more important lately. They suit the purpose of finding a solution to a problem within a group, so that all members are satisfied. An example would be to recommend a suitable route for a holiday, containing interesting sightseeing destinations for each member and representing a good compromise for all of them. Social networks such as Facebook are used more frequently nowadays to plan conjoint events within a group. As far as known by the author, there is no application integrated in Facebook so far, which provides a holiday planning system based on recommender systems.

Finding suitable sightseeing destinations and a suitable route for the whole group is a great challenge, because the individual members usually have different interests. A further challenge constitutes the designing of a route algorithm, which dynamically computes a route of several days' duration by taking into account a couple of constraints.

Most of the recent holiday planners based on recommender systems are constructed for single persons. Furthermore, opening times are hardly ever taken into consideration in routing algorithms. With most such algorithms, computing a route of several days' duration is not possible or start- and endpoints respectively -regions of all daily part routes must be specified in advance by the user. As far as known by the author, no holiday planning system based on recommender systems for groups exists so far, representing a complete planning system with which it is possible to select sightseeing destinations taking into account the interests of all group members, as well as to compute a route fitting to the needs of all members, where start- and endpoints of each day do not need to be specified in advance.

Hence, in this work such a system which resolves all the mentioned weak points was designed, developed and integrated in the social network Facebook as an application. The developed "Holiday Recommender" suggests suitable sightseeing destinations to each group member based on the member's interests, which can then be brought in to the travel list as a proposal by the member. Each of the sightseeing destinations, brought in that way, can then be evaluated by each member. In the next step, three suitable routes are calculated using different decision strategies, which contain the most suitable sightseeing destinations optimised by evaluation, distance, and opening time. Furthermore, the routes are path- and time-optimised. Finally, the group can select one of the recommended routes.

Due to the fact that fitting start- and endpoints of all daily part routes are calculated by the algorithm itself (with exception of the startpoint of the first day and the endpoint of the last day), the problem of the daily part routes optimisation was solved. Thereby, the selection of the sightseeing destinations for the route can be optimised much better, because fixed start- and endpoints of each day would constitute a restriction of flexibility of the overall route.

Several test groups have evaluated the "Holiday Recommender", whereby the usability of the application has been verified.



## Kurzfassung

Recommender Systeme für Gruppen haben in letzter Zeit immer mehr an Bedeutung gewonnen. Sie dienen dazu, innerhalb einer Gruppe eine Lösung für ein Problem zu finden, mit der alle Mitglieder zufrieden sind. Ein Beispiel wäre, für einen Urlaub eine geeignete Route vorzuschlagen, die für jedes Gruppenmitglied interessante Sehenswürdigkeiten enthält und die für alle Mitreisenden einen guten Kompromiss darstellt. Soziale Netzwerke wie Facebook werden heutzutage immer häufiger benutzt, um gemeinsame Events innerhalb einer Gruppe zu planen. Soweit der Autorin bekannt ist, gibt es noch keine in Facebook integrierte Applikation, die einen Urlaubsplaner auf Basis von Recommender Systemen anbietet.

Geeignete Sehenswürdigkeiten und eine passende Urlaubsrouten für die ganze Gruppe zu finden ist eine große Herausforderung, da die einzelnen Gruppenmitglieder meist unterschiedliche Interessen haben. Eine weitere Herausforderung stellt der Entwurf eines Routenalgorithmus dar, der dynamisch eine Reiseroute über mehrere Tage unter Beachtung einiger Nebenbedingungen berechnen kann.

Die meisten bisherigen Urlaubsplaner auf Empfehlungsbasis sind für Einzelpersonen ausgelegt. Auch werden bei Routenalgorithmus selten Öffnungszeiten berücksichtigt. Meistens ist bei diesen eine Berechnung über mehrere Tage nicht möglich oder es müssen die Start- und Endorte, bzw. -gebiete, aller Tagesteilstrecken vom Benutzer im Vorhinein angegeben werden. Soweit der Autorin bekannt ist, gibt es noch keinen Urlaubsplaner auf Empfehlungsbasis für Gruppen, der ein vollständiges Planungssystem darstellt, das es ermöglicht, sowohl Sehenswürdigkeiten unter Berücksichtigung der Interessen aller Gruppenmitglieder auszuwählen als auch eine Reiseroute zu berechnen, die auf die Bedürfnisse aller Mitglieder zugeschnitten ist, wobei Start- und Endpunkte jedes Tages nicht im Vorhinein angegeben sein müssen.

In dieser Arbeit wurde daher ein solches System, das alle genannten Schwachstellen behebt, entwickelt, entwickelt und als Applikation in das soziale Netzwerk Facebook integriert. Der entwickelte "Holiday Recommender" schlägt für jedes Gruppenmitglied aufgrund dessen persönlicher Interessen passende Sehenswürdigkeiten vor, die das Gruppenmitglied dann als Vorschlag in die Reiseliste einbringen kann. Jede so eingebrachte Sehenswürdigkeit kann anschließend von jedem Mitglied bewertet werden. Im folgenden Schritt werden unter Verwendung verschiedener Entscheidungsstrategien drei passende Reiserouten berechnet, die je nach Bewertungen, Entfernungen und Öffnungszeiten optimiert die passendsten Sehenswürdigkeiten enthalten und zudem weg- und zeitoptimiert sind. Zuletzt kann sich die Gruppe gemeinsam für eine der empfohlenen Routen entscheiden.

Dadurch, dass die Tagesanfangs- und Tagesendpunkte vom Algorithmus selbst passend bestimmt werden (mit Ausnahme des Startpunktes des ersten Tages und des Endpunktes des letzten Tages), wurde das Problem der Optimierung der Tagesteilstrecken gelöst. Dadurch kann auch die Auswahl der Sehenswürdigkeiten für die Route viel besser optimiert werden, da fixe Anfangs- und Endorte jedes Tages auch eine Einschränkung der Flexibilität der Gesamtroute darstellen würden.

Mehrere Testgruppen haben den "Holiday Recommender" evaluiert, wodurch die Benutzerfreundlichkeit der Applikation bestätigt wurde.



## **Statutory Declaration**

*I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.*

\_\_\_\_\_

Place

\_\_\_\_\_

Date

\_\_\_\_\_

Signature

## **Eidesstattliche Erklärung**

*Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.*

\_\_\_\_\_

Ort

\_\_\_\_\_

Datum

\_\_\_\_\_

Unterschrift



# Danksagung

Zuerst möchte ich mich bei meinem Betreuer, Univ.-Prof. Dipl.-Ing. Dr.techn. Alexander Felfernig für seine Unterstützung, sein Feedback und seine Geduld, die er mir während der Arbeit entgegen brachte, bedanken. Ebenfalls bedanke ich mich für das Feedback von Dipl.-Ing. Gerald Ninaus und Dipl.-Ing. Mag. Florian Reinfrank.

Ein spezieller Dank gilt auch Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Keith Andrews für das  $\LaTeX$ -Grundgerüst [Andrews, 2012b], das für diese Arbeit verwendet wurde sowie für die in diesem Gerüst enthaltene Hilfestellung zur Verfassung einer guten Diplomarbeit.

Herrn Dipl.-Ing. Gerald Ninaus und Herrn Dipl.-Ing. Martin Stettinger vom Institut für Softwaretechnologie danke ich für die Einrichtung und zur Verfügungsstellung eines Serverplatzes, auf dem die Software zur Testung des Systems laufen konnte.

Weiters danke ich Sascha Nemecek für die Hilfestellungen in  $\LaTeX$  sowie allen Testpersonen, die an der Evaluierung der Applikation “Holiday Recommender” teilgenommen haben für ihre Zeit, ihr Engagement und ihre Verbesserungsvorschläge.

Zum Schluss möchte ich mich noch bei meinem Freund, meinen Freunden und meiner Familie für ihre mentale Unterstützung und ihr Verständnis bedanken.

Eva Harb  
Graz, Mai 2015



# Vorwort

Ich möchte den folgenden Personen und Organisationen für die zu Verfügungsstellung ihres Materials danken:

- Diese Arbeit wurde mit Hilfe des  $\LaTeX$ -Grundgerüsts von Ao.Univ.-Prof. Dipl.-Ing. Dr.techn. Keith Andrews [Andrews, 2012b] erstellt.
- Der CKEditor von Frederico Knabben [Knabben, 2015] wurde in die Applikation eingebunden, um Reisenotizen verwalten zu können.
- Das Menü, die Buttons, die Comboboxen, die Dialogboxen, die Slider, die Date-Picker und weitere Funktionen der Applikation wurden mit Hilfe von jQuery [jQuery Foundation, 2015] erstellt.
- Die Sterne, um in der Applikation Bewertungen abgeben zu können, wurden von Openclipart [2015], User “klainen” adaptiert.
- Das “X”, um in der Applikation Sehenswürdigkeiten von der Reiseliste löschen zu können, wurde von Openclipart [2015], User “dagobert83” adaptiert.
- Das Kalenderbild, das in der Applikation auf der Seite der Reisedaten rechts in den Kalendereingabefeldern angezeigt wird, wurde von Openclipart [2015], von “Francesco ’Architetto’ Rollandin” adaptiert.
- Das “Bitte-warten-gif”, das dem Benutzer in der Applikation während der Suche nach Sehenswürdigkeiten und während der Berechnung der Routen angezeigt wird, wurde mit Hilfe von Ajaxload [Croissant, 2015] erstellt.
- Der grundlegende Fragebogen, der den Testpersonen ausgehändigt wurde, um die Benutzerfreundlichkeit der Applikation zu bewerten, wurde von John Brooke [Brooke, 1996] entworfen.
- Die in der Applikation angezeigten Karten und die automatische Vervollständigung der Ortseingaben sowie auch die vorgeschlagenen Sehenswürdigkeiten und deren Details wurden von Google [Google API, 2013] zur Verfügung gestellt.
- Die Distanzen zwischen allen Sehenswürdigkeiten sowie dem Start- und Endort, die benötigt wurden, um in der Applikation die Routen zu berechnen, wurden von MapQuest [MapQuest API, 2014] zur Verfügung gestellt.
- Das Online-Forum “Stack Overflow” [Stackoverflow, 2015] hat sich bei der Entwicklung der Applikation als sehr hilfreich erwiesen.



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Aktuelle Trends im Internet . . . . .	1
1.2	Aktuelle Trends im Tourismus . . . . .	2
1.3	Aktueller Trend der Gruppennutzung des Internets . . . . .	2
1.4	Motivation . . . . .	3
1.5	Problemstellung . . . . .	3
1.6	Das Routenproblem . . . . .	4
1.7	Zielsetzung . . . . .	6
<b>2</b>	<b>Themenverwandte Arbeiten</b>	<b>9</b>
2.1	Grundlagen . . . . .	9
2.2	Gruppenempfehlungen unter Berücksichtigung sozialer Komponenten . . . . .	11
2.3	Identifikation von Gruppen . . . . .	14
2.4	Empfehlungsapplikationen auf Facebook . . . . .	15
2.4.1	Empfehlungsapplikationen für Einzelpersonen . . . . .	15
2.4.2	HappyMovie - Eine Empfehlungsapplikation für Gruppen . . . . .	16
2.4.3	GroupFun - Eine Empfehlungsapplikation für Gruppen . . . . .	17
2.5	Urlaubsplaner auf Empfehlungsbasis im Web . . . . .	19
2.5.1	e-Tourism . . . . .	19
2.5.2	Your Tour . . . . .	21
2.5.3	City Trip Planner . . . . .	22
2.5.4	INTRIGUE . . . . .	23
2.5.5	Travel Decision Forum . . . . .	23
2.6	Routenberechnung . . . . .	26
2.6.1	Personalisierte Berechnung und Empfehlung einer Route . . . . .	26
2.6.2	Methoden zur Routenberechnung . . . . .	27
<b>3</b>	<b>Beschreibung der Applikation aus Benutzersicht</b>	<b>31</b>
3.1	Grundlegende Funktionsweise und Überblick . . . . .	31
3.2	Detaillierte Beschreibung der Einzelkomponenten . . . . .	32
3.2.1	Startseite . . . . .	32
3.2.2	Verwaltungs-Hauptseite . . . . .	33
3.2.3	Reisedaten . . . . .	34
3.2.4	Persönliche Interessen . . . . .	36
3.2.5	Sehenswürdigkeiten . . . . .	38
3.2.5.1	Reisefliste . . . . .	40
3.2.5.2	Sehenswürdigkeiten vorschlagen lassen . . . . .	44
3.2.5.3	Sehenswürdigkeiten hinzufügen . . . . .	46
3.2.6	Reiserouten . . . . .	46
3.2.7	Notizen . . . . .	51
3.2.8	Szenario . . . . .	52

<b>4 Technische Beschreibung der Applikation</b>	<b>57</b>
4.1 Technische Grundlagen . . . . .	57
4.2 Empfehlen von Sehenswürdigkeiten . . . . .	57
4.3 Empfehlen von Routen . . . . .	61
4.3.1 Entscheidungsstrategien . . . . .	62
4.3.2 Inputdaten und Vorbedingungen . . . . .	63
4.3.3 Übersicht der Funktionsweise des Algorithmus' . . . . .	64
4.3.4 Der Einfügeprozess . . . . .	68
4.3.5 Der Mischprozess . . . . .	73
<b>5 Evaluierung der Applikation</b>	<b>79</b>
5.1 Aufbau des Tests . . . . .	79
5.2 Durchführung des Tests . . . . .	80
5.3 Feedback-Fragebogen . . . . .	81
5.4 Ergebnis des Tests . . . . .	83
<b>6 Konklusion</b>	<b>85</b>
6.1 Ergebnis . . . . .	85
6.2 Einschränkungen der Applikation . . . . .	87
6.3 Weiterentwicklung für die Zukunft . . . . .	88
<b>Abbildungsverzeichnis</b>	<b>91</b>
<b>Quellcodeverzeichnis</b>	<b>93</b>

# Kapitel 1

## Einleitung

Die Menschen in der heutigen Zeit haben in unseren Breitengraden oft höhere Ansprüche in verschiedensten Hinsichten wie Ansprüche an Gegenstände, Dienstleistungen oder persönlichen Lebenserwartungen, als noch vor einigen Jahrzehnten. Das liegt mit Sicherheit auch daran, dass die Anzahl an verschiedenen Möglichkeiten und die Anzahl der Auswahl bezüglich allem ständig wächst. Auch gibt es einen spürbaren Trend in die Richtung, Produkte und Dienstleistungen individuell auf den Kunden abzustimmen beziehungsweise den Kunden die Möglichkeit zu geben, selbst mitgestalten zu können. So kann man heute beispielsweise mühelos T-Shirts mit eigenen Motiven bedrucken lassen oder sein eigenes, individuelles Fotobuch zusammenstellen. Da die Auswahl an Produkten wie auch die Ansprüche der Menschen mittlerweile groß sind, ist es oft schwierig, das am besten passende Produkt zu finden, sodass die Kunden zufrieden sind.

### 1.1 Aktuelle Trends im Internet

Auch im Internet ist dieser Trend deutlich spürbar. Das Internet wird heutzutage immer häufiger genutzt und es gibt mittlerweile nahezu keine Alltagssituation mehr, wo nicht ein Programm, eine Webseite oder eine Applikation Unterstützung bieten kann. Seit dem Web 2.0 steht auch die Interaktivität zwischen Mensch und Internet stark im Vordergrund. Aufgrund der höheren Ansprüche der Menschen und des wachsenden Bedürfnisses der Menschen nach individuell angepassten Angeboten steigt die Individualisierung des Internetangebotes stark an. Durch die steigende Vernetzung der Dinge und dem breiter gefächerten Angebot erleidet der Benutzer im Internet oft eine wahre Informationsüberflutung.

Um diesem Problem Abhilfe zu schaffen, sowie dem Wunsch des Kunden nach individuell abgestimmten Produkten nachzukommen, wurden sogenannte "Recommender Systeme", also Empfehlungssysteme, die zum Ziel haben, aus einer Menge von Daten diese herauszusuchen, die für den individuellen Benutzer relevant sind, entwickelt. Wie Jannach u. a. [2011] erwähnen, hatten die meisten Internetbenutzer bereits auf die eine oder andere Art Kontakt mit solchen Empfehlungssystemen. Sei es unangefordert, wie zum Beispiel bei diversen Online-Shops, wo man automatisch Artikel angezeigt bekommt, die einen ebenfalls interessieren könnten, wenn man sich ein Produkt ansieht oder aufgefordert, indem man zum Beispiel bewusst einem System persönliche Interessen bekannt gibt, um personalisierte Ergebnisse zu erhalten.

Mit der Einführung des Web 2.0 haben auch soziale Netzwerke wie zum Beispiel "Facebook" großes Aufsehen erlangt [Symeonidis, Ntempos und Manolopoulos, 2014]. Diese bieten neben vielen Informationen auch vor allem soziale Kontaktmöglichkeiten an. Viele Menschen, die ein gemeinsames Interesse haben, planen gemeinsame Events auf Facebook. Auch organisieren sich viele Gruppen über soziale Netzwerke. Symeonidis, Ntempos und Manolopoulos [2014] bemerken in ihrer Arbeit, dass diese Netzwerke Daten enthalten, die aus ihnen ausgelesen und verwendet werden können, um persönliche Empfehlungen bezüglich Produkten, Benutzern oder digitalen Inhalten machen zu können. Solche Daten sind

## 1. Einleitung

zum Beispiel Alter, Geschlecht, Beziehungsstatus, Wohnort und persönliche Interessen. Auch soziale Netzwerke haben demzufolge ein großes Potenzial, Empfehlungssysteme zum Einsatz zu bringen.

### 1.2 Aktuelle Trends im Tourismus

Auch in der Tourismusbranche ist der aktuelle Trend dahingehend, dass die Nachfrage nach Individualurlauben steigt und sich 08/15-Massentouren nicht mehr allzu großer Beliebtheit erfreuen. Urlauber wollen ihre Reise selbst gestalten und diese speziell an ihre Bedürfnisse und Vorlieben anpassen. Reisebüros bieten zwar mittlerweile vermehrt Individualreisen an, wobei aber auch diese meist nicht ausreichend auf die einzelnen Personen abgestimmt sind.

Wie Souffriau und Vansteenwegen [2010] zusammenfassen, gibt es Reiseführer, die allgemeine Tour-Vorschläge durch eine bestimmte Stadt oder Region beinhalten. Diese Touren sind aber auf die Interessen der Mehrheit abgestimmt und nehmen auf die speziellen, individuellen Bedürfnisse von verschiedenen Personen keine Rücksicht [Cheverst u. a., 2000]. Dunlop u. a. [2004] erklären, dass Reiseführer keine temporalen Informationen über zum Beispiel temporäre Ausstellungen in Museen, Sehenswürdigkeiten, die vorübergehend wegen einer Renovierung geschlossen sind oder aktuelle Theaterprogramme anbieten können. Kramer, Modsching und Hagen [2006] haben die Vielfalt von gesammelten Touristeninteressenprofilen analysiert und herausgefunden, dass diese überraschend unterschiedlich sind, weswegen es sinnvoll ist, personalisierte Touren anstatt von allgemeinen Touren anzubieten.

Wenn der Urlaubsort erst einmal feststeht, kann man viele Seiten im Internet finden, die Informationen über Sehenswürdigkeiten einer Stadt, Aktivitäten, die man während der Reise machen kann, Restaurants und vieles mehr anbieten [Castillo u. a., 2008]. Aber auch diese Informationen sind wie auch in den Reiseführern meist statisch und werden allen Benutzern gleich dargestellt. Des Weiteren ist die Menge an verfügbaren Informationen oft sehr groß, sodass sie die Benutzer selbst filtern und entscheiden müssen, welche Teile für sie daraus interessant sind und welche nicht. Solche allgemeinen Tourismuseiten bieten keine Touren oder Zeitpläne, die an die Bedürfnisse und Interessen des Benutzers angepasst sind. Aus Benutzersicht ist es daher sinnvoll, ein Empfehlungssystem zu haben, das automatisch vorschlägt, welche Sehenswürdigkeiten für den jeweiligen Benutzer in einer bestimmten Stadt interessant zu sehen wären. Dabei muss ein täglicher Plan auf Basis des Benutzerprofils berechnet werden, der dem Benutzer anzeigt, welche Sehenswürdigkeiten wann besichtigt werden sollen und wie er von einem Ort zum nächsten kommt.

Wie Sebastia u. a. [2009b] anmerken, existiert heute ein wachsendes Interesse an Empfehlungssystemen im Tourismusbereich im Internet, da immer mehr Menschen urlaubsbezogene Webseiten besuchen, um Informationen zu ihrem geplanten Urlaub zu erhalten.

### 1.3 Aktueller Trend der Gruppennutzung des Internets

Seit einiger Zeit erleben wir eine Entwicklung der Computer dahingehend, vom ursprünglichen Desktop-PC wegzugehen [Jameson und Smyth, 2007]. Dies führt dazu, dass von den neuen Systemen nicht mehr nur Einzelpersonen, sondern zunehmend auch Gruppen angesprochen werden. Wanddisplays, Informationsterminals, PDAs und Mobiltelefone können von Menschen auf einfache Art benutzt werden, um miteinander zu agieren. Und auch mit dem traditionellen PC ermöglicht sich für die Benutzer eine große Vielfalt an Wegen, miteinander zu kommunizieren und gemeinsam Projekte durchzuführen. Aus diesen Gründen kann man ein weiteres Wachstum des Trendes dahingehend erwarten, dass Empfehlungssysteme für Gruppen immer mehr gefragt werden.

Viele soziale Aktivitäten, wie ein gemeinsamer Kinobesuch, ein gemeinsamer Videoabend, eine Grillfeier oder ein Urlaub werden mittlerweile gerne über das Internet organisiert. Dabei gibt es viele Möglichkeiten, wie eine Aktivität aussehen kann, und je unterschiedlicher die Gruppenmitglieder sind, desto schwieriger ist es, sich auf eine Lösung zu einigen. Nicht nur Zeit, Ort und andere Rahmenbe-

dingungen sind zu klären, sondern auch die Einzelheiten: welcher Film soll angesehen werden, welche Speisen landen am Grill? Um so etwas selbst gemeinsam über das Internet zu organisieren, sind viele Interaktionen zu tätigen und viele Kompromisse zu finden, was einen erheblichen Zeitaufwand bedeutet und auch ein sehr mühsamer Prozess ist.

Empfehlungssysteme für Gruppen wurden entwickelt, um Gruppen von Menschen zu helfen, sich auf eine gemeinsame Aktivität oder ein Element zu einigen [Quijano-Sánchez, Recio-García und Díaz-Agudo, 2011], [Stettinger, 2013]. Heutzutage werden für gewöhnlich soziale Netzwerke benutzt, um Events oder Aktivitäten in eine Gruppe von Menschen zu organisieren. Es ist mittlerweile sehr üblich geworden, Facebook oder andere Online-Communities zum Vorschlagen einer Aktivität zu nutzen und dann mehrere Freunde zu dieser einzuladen. Daher sind soziale Netzwerke eine ideale Umgebung um Empfehlungssysteme einzusetzen.

## 1.4 Motivation

Zusammenfassend kann man folgende Trends in der heutigen Zeit beobachten:

- Weg von Massenware hin zu individuellen Einzellösungen
- Das Internet wird immer häufiger genutzt, auch zur Organisation von Gruppenprojekten
- Soziale Netzwerke erfreuen sich immer größerer Beliebtheit
- Informationen zu Reisen werden gerne aus dem Internet gewonnen
- Die Informationsüberflutung im Internet macht Empfehlungssysteme immer notwendiger

Aufgrund dieser Entwicklungen ist es sinnvoll, webbasierende Empfehlungssysteme für Urlaubsplanungen anzubieten. In den meisten Fällen reist man nicht alleine, sondern in einer Gruppe mit anderen Menschen. Das kann sowohl die eigene Familie sein wie auch einige Freunde oder Arbeitskollegen. Wie schon erwähnt, organisieren sich Gruppen gerne über soziale Netzwerke. Eines der bekanntesten davon ist Facebook, weswegen Facebook die ideale Plattform ist, um ein Empfehlungssystem für Gruppenreisen anzubieten. Da die einzelnen Gruppenmitglieder meist sehr unterschiedliche Interessen bezüglich ihres Urlaubes haben, ist es wichtig, Gruppen bei der Entscheidung der Sehenswürdigkeitenauswahl zu unterstützen und eine geeignete Route zu berechnen, sodass alle Mitglieder zufrieden sind.

## 1.5 Problemstellung

Im Rahmen dieser Arbeit wurde der “Holiday Recommender” entwickelt, eine Facebook-Applikation, die einer Gruppe von Personen mittels Empfehlungssystemen bei der Planung eines gemeinsamen Urlaubes hilft. Es ist eine große Herausforderung, so ein System zu entwickeln, da es viele Faktoren und eine Vielzahl an Anforderungen gibt, die bei der Lösung berücksichtigt werden müssen. Gerade bei Empfehlungssystemen für Gruppen ist es schwierig, alle Mitglieder zufrieden zu stellen und die Bedürfnisse aller zu decken bzw. einen guten Kompromiss, mit dem alle leben können, zu finden. Bei Empfehlungssystemen für Gruppen gibt es keine bestimmte beste Lösung, die man einfach berechnen kann und dessen Vorgehensweise man für alle verschiedenen Gebiete oder auch alle verschiedenen Gruppen anwenden kann. Es ist wichtig, den Kontext, in dem das Empfehlungssystem zum Einsatz kommen soll, genau zu analysieren und auch die Zielpersonen grob einschätzen zu können. Es gibt viele verschiedene Strategien, um eine passende Entscheidung für eine Gruppe zu finden. Hier spielt zum Beispiel auch die Größe der Gruppe eine Rolle. Wie Boratto, Carta und Satta [2010] erläutern, sollte bei großen Gruppen eher eine Durchschnittsbewertung der von den Mitgliedern bewerteten Elemente berechnet werden, bei kleineren Gruppen ist eher auf jenes Mitglied Rücksicht zu nehmen, das mit einem Element überhaupt nicht einverstanden ist. Des Weiteren spielt es eine Rolle, wie gut sich die Mitglieder untereinander kennen,

## 1. Einleitung

sowie auch welche Persönlichkeiten sie haben und wie ihre Bereitschaft ist, auf andere Rücksicht zu nehmen. Auch sollte man bedenken, ob es Personen mit besonderen Bedürfnissen in der Gruppe gibt, oder Experten, dessen Bewertungen den anderen besonders wichtig sind.

Auch entscheidend ist, welche Informationen man bereits über die Benutzer hat und inwiefern man diese auswerten und nutzen kann. Man unterscheidet dabei zwischen impliziten und expliziten Informationen. Bei Facebook-Applikationen kann auf das Profil der Benutzer zugegriffen werden, wobei man verschiedene Daten wie Alter, Geschlecht, Wohnort usw. auslesen und implizit Folgerungen schließen kann. Bei Online-Shops ist auslesbar, welche Produkte der Benutzer bereits angesehen oder gekauft hat, um implizit seine Interessen zu kennen, während man bei manchen anderen Anwendungen vielleicht überhaupt keine Daten hat und die benötigten daher explizit vom Benutzer anfordern muss, indem man ihn Angaben zu seinen Interessen machen oder Datensätze bewerten lässt. Ebenso wie Inputdaten unterschiedlich sein können, können auch die gewünschten Outputs unterschiedlich sein. So werden bei manchen Systemen viele Objekte vorgeschlagen, bei anderen nur ein einziges.

Sollen mehrere Objekte für eine Gruppe vorgeschlagen werden, gibt es verschiedene Möglichkeiten um diese zu finden [Jameson und Smyth, 2007]:

- **unabhängig voneinander**

Dies könnte eventuell unfair sein, da ein Mitglied gesamt gesehen vielleicht benachteiligt wird.

- **abhängig von einander**

Dabei wird eine Sequenz von Objekten als eigenes Empfehlungsproblem gesehen. Dies ist fair, aber es ist nicht möglich, Ergebnisse von früheren Entscheidungen mitzuberücksichtigen.

- **individuell mit Rücksicht auf andere Entscheidungen**

Ein Mittelweg, bei dem die Objekte zuerst unabhängig von einander generiert werden. Wenn jedoch auffällt, dass ein Mitglied benachteiligt wurde, bekommt dieses Mitglied mehr Gewicht, wobei die Ergebnisse von gerade entstandenen Entscheidungen mit berücksichtigt werden. Dies ist sehr komplex und das Ergebnis ist der Gruppe schwer zu erklären.

Man muss sich auch überlegen, welche weiteren Faktoren in den Entscheidungsprozess mit einfließen. Empfiehlt man zum Beispiel Sehenswürdigkeiten oder Restaurants, so muss man, wie Łazaruk u. a. [2012] anmerken, bei der Entscheidung auch die Entfernungen mit berücksichtigen. Es wäre nicht sinnvoll, ein perfekt passendes Restaurant in Australien vorzuschlagen, wenn der Benutzer gerade in Österreich ist. Delgado und Davidson [2002] erklären, dass Empfehlungen bezüglich Reisen nicht nur aufgrund der Benutzerinteressen gemacht werden dürfen, sondern auch Informationen über die Umgebung, wie zum Beispiel Distanzen zwischen den Sehenswürdigkeiten, Reiseart (Auto, Fahrrad), Jahreszeit, Öffnungszeiten der Sehenswürdigkeiten usw. mit in den Entscheidungsprozess einfließen müssen. Beim "City Trip Planner" [Vansteenwegen u. a., 2011], der nicht im Voraus plant, sondern direkt vor Ort eine passende Tour findet, fließen auch noch Wetter und überfüllte Orte mit ein. Auch ein bestimmtes Budget-Limit könnte man noch mit berücksichtigen.

## 1.6 Das Routenproblem

Bei einem Urlaub in einer Stadt oder einer Region können nicht alle dort vorhandenen Sehenswürdigkeiten besucht werden, da sich das zeitlich nicht ausgehen würde. Es müssen also passende Sehenswürdigkeiten für die Gruppe ausgewählt werden. Dazu muss zu jeder Sehenswürdigkeit eine Bewertung vorliegen, die das Interesse der gesamten Gruppe an dieser Sehenswürdigkeit ausdrückt, wobei dieses Interesse nach mehreren Entscheidungsstrategien, die in Kapitel 2.2 genauer vorgestellt werden, ermittelt werden kann.

Wie in Kapitel 1.5 erwähnt wurde, gibt es viele Faktoren und Bedingungen, die bei der Berechnung einer Route mit berücksichtigt werden müssen. Zum einen müssen die passendsten Sehenswürdigkeiten

ausgewählt werden, die sich in der Urlaubszeit ausgehen, zum anderen muss eine Route gefunden werden, die diese Sehenswürdigkeiten in bestmöglicher Reihenfolge beinhaltet, wobei alle Bedingungen wie Urlaubsdauer, Besichtigungszeiten, Öffnungszeiten, Aufenthaltsdauer, Entfernungen der Sehenswürdigkeiten, Interesse an den Sehenswürdigkeiten usw. berücksichtigt werden müssen.

Diese Auswahl der Sehenswürdigkeiten gemeinsam mit dem Routenproblem wird oft “Touristen Trip Design Problem” [Vansteenwegen und Oudheusden, 2007] genannt, das eine Erweiterung des Orientierungsproblems ist und ein schwieriges kombinatorisches Optimierungsproblem darstellt [Vansteenwegen u. a., 2011]. Dem Orientierungsproblem wiederum liegt das “Problem des Handelsreisenden” zugrunde.

Beim “Problem des Handelsreisenden” muss ein Handelsvertreter in kürzester Zeit alle vorgegebenen Orte besuchen (das heißt, er muss die kürzeste Strecke finden), wobei er beim Depot startet, jeden Ort exakt ein Mal besucht und wieder zum Depot zurückkehrt. Golden, Levy und Vohra [1987] haben bewiesen, dass dieses Problem NP-Hart ist, weswegen eine Berechnung einer optimalen Lösung in polynomieller Zeit nicht möglich ist. Eine solche Berechnung steigt mit der Anzahl an Orten exponentiell an. Das “Subset-Tour-Problem des Handelsreisenden” ist eine Spezialform, bei der nicht alle Orte besucht werden müssen, das aber ebenfalls NP-Hart ist [Mittenthal und Noon, 2007]. Unter der Berücksichtigung des mathematischen Modells des Problems ist es sehr schwierig, wenn nicht unmöglich, eine optimale Lösung zu finden [Mittenthal und Noon, 2007].

Beim Orientierungsproblem bekommt jeder Teilnehmer einen Kompass und eine Landkarte, auf der die zu besuchenden Orte (Kontrollpunkte) eingezeichnet sind [Sun und Lee, 2004]. Die Teilnehmer haben nur eine begrenzte Zeit zur Verfügung, in der sie möglichst viele Kontrollpunkte besuchen und am Ende einen gegebenen Endpunkt erreichen müssen. Sie müssen dabei nicht alle Kontrollpunkte besuchen, jedoch darf jeder Kontrollpunkt höchstens ein Mal besucht werden. Jeder Kontrollpunkt hat ein Gewicht, und die Aufgabe ist es, in der gegebenen Zeit einen möglichst hohen Punktwert zu erreichen. Jeder Teilnehmer muss also ein Optimierungsproblem lösen, um seine Route zu planen, auszuwählen, welche und wie viele Kontrollpunkte er besucht und sicherzustellen, dass dabei die gegebene Zeit gut ausgenutzt, aber nicht überschritten wird. Beim “Problem des Handelsreisenden” werden also die Kosten minimiert, die gebraucht werden, um alle Orte zu besuchen, während beim Orientierungsproblem die Gesamtpunkte unter Einhaltung der Nebenbedingungen maximiert werden.

Um einer Gruppe eine geeignete Route mit passenden Sehenswürdigkeiten vorschlagen zu können, muss also das “Touristen Trip Design Problem” gelöst werden. Dazu müssen die Koordinaten der Sehenswürdigkeiten, die Koordinaten des Start- und Endortes sowie alle Distanzen zwischen ihnen bekannt sein. Ebenfalls muss die Besichtigungszeit jeder Sehenswürdigkeit bekannt sein, wie auch die Zeiträume, in denen die Gruppe Besichtigungen vornehmen möchte. Dann können noch weitere Nebenbedingungen hinzugefügt werden wie Öffnungszeiten, Reiseart und viele andere, die das Problem zusätzlich komplexer machen. Des Weiteren stellt die Berechnung einer mehrtägigen Route eine Herausforderung dar.

Wie schon erwähnt, steigt bei einer Berechnung einer exakten Lösung die Berechnungszeit mit der Anzahl der zu besuchenden Orte bereits beim “Problem des Handelsreisenden” exponentiell an. Soll das Orientierungsproblem gelöst werden, wobei auch Öffnungszeiten und generelle Zeiten, in denen Besuche möglich sind, berücksichtigt werden, würde eine exakte Berechnung noch länger dauern und wäre daher völlig ungeeignet, wie Righini und Salani [2009] anmerken, da selbst mit sehr viel Rechenpower eine Berechnung in einer annehmbaren Zeit nicht möglich wäre. Es muss daher eine Näherungslösung gefunden werden.

Der berechnete Routenplan ist immer ein Kompromiss zwischen den persönlichen Interessen der Gruppe und den Bedingungen, die erfüllt werden müssen, wie zum Beispiel den räumlichen und zeitlichen Bedingungen [Soo und Liang, 2001]. Es muss also eine abgestimmte, passende Lösung gefunden werden, die den Bedingungen und den Wünschen der Gruppe gerecht wird.

### 1.7 Zielsetzung

Die Zielsetzung bei der Entwicklung des “Holiday Recommenders” ist es, eine Gruppe von Personen, die gemeinsam einen Urlaub machen will, dabei zu unterstützen, geeignete Sehenswürdigkeiten zu finden sowie drei Routen zu generieren, die für die Gruppe passend sein könnten. Die Benutzer geben dazu neben den Reisedaten auch ihre Interessen an verschiedenen Kategorien von Sehenswürdigkeiten bekannt, woraufhin für jedes Gruppenmitglied extra, je nach dessen Interesse, Sehenswürdigkeiten vorgeschlagen werden, die das Gruppenmitglied dann für die gemeinsame Reise vorschlagen kann. Es können dabei aber auch eigene Sehenswürdigkeiten eingebracht werden. Nachdem jedes Mitglied so Sehenswürdigkeiten eingebracht hat, werden diese von allen Mitgliedern bewertet. Danach werden drei Routen nach drei verschiedenen Entscheidungsstrategien berechnet, von denen die Gruppe dann gemeinsam eine auswählt. Bei den drei Routen wird einmal die Durchschnittsbewertung der Gruppe für eine Sehenswürdigkeit berücksichtigt, einmal das glücklichste Mitglied und einmal das unglücklichste Mitglied pro Sehenswürdigkeit.

Die Routenberechnung stellt dabei eine besondere Herausforderung dar, da der “Holiday Recommender” Routen über mehrere Tage berechnen kann, wobei die Start- und Endorte der Tagesteilstrecken nicht bekannt sein müssen, immer unterschiedlich sein können und vom System selbst passend gewählt werden. Soweit der Autorin bekannt ist, ist diese Besonderheit bislang einzigartig. Des Weiteren ist das System nicht auf wenige Orte beschränkt, wie das bei einigen anderen Systemen der Fall ist, die meist auch nur für Einzelpersonen ausgelegt sind, sondern es können Orte in der ganzen Welt besucht werden, was durch die Vernetzung des “Holiday Recommenders” mit Google und MapQuest ermöglicht wird. Aus den von der Gruppe eingebrachten Sehenswürdigkeiten werden diejenigen automatisch ausgesucht, die aufgrund der Bewertungen und der Entfernung am relevantesten sind. Auch wird dabei auf Öffnungszeiten geachtet, sodass ein Museum, das sonntags geschlossen hat, nicht an einem Sonntag vorgeschlagen wird. Kommt man vor Beginn der Öffnungszeiten bei der Sehenswürdigkeit an, ist eine Wartezeit möglich, wobei der dadurch entstandene Zeitverlust im Optimierungsalgorithmus berücksichtigt wird. Die Öffnungszeiten einer Sehenswürdigkeit können dabei täglich variieren.

Um das System nicht unnötig kompliziert zu machen, wurde auf eine genaue Zeitangabe in der Reiseroute, wann welche Sehenswürdigkeit besucht werden soll verzichtet. Es ist in den meisten Fällen nicht möglich, einen Reiseplan exakt einzuhalten, weswegen eine genaue Zeitangabe nicht sinnvoll erscheint. Auch die Angabe der Öffnungszeiten beschränkt sich nach außen hin nur auf die Angabe der Tage, an denen die Sehenswürdigkeit besucht werden darf. Der interne, im Rahmen dieser Arbeit entwickelte Routenalgorithmus kann allerdings mit exakten Öffnungszeiten umgehen und kann ebenfalls auch Zeitpläne erstellen, die genaue Uhrzeitangaben beinhalten.

Im “Holiday Recommender” sind also zwei Empfehlungssysteme integriert. Eines, das geeignete Sehenswürdigkeiten für Einzelpersonen vorschlägt und ein zweites, das geeignete Routen nach drei verschiedenen Entscheidungsstrategien berechnet und selbst je nach Strategie unter Einhaltung der Nebenbedingungen diejenigen Sehenswürdigkeiten auswählt, die für die Gruppe am besten passen.

Ein wichtiger Punkt beim Entwurf des “Holiday Recommenders” ist, auf die Benutzerfreundlichkeit des Systems zu achten, da eine Applikation nur dann verwendet werden wird, wenn sich die Benutzer schnell mit dem System zurechtfinden und die Handhabung einfach und intuitiv ist.

Die Aufteilung dieser Arbeit gliedert sich folgendermaßen:

In Kapitel 2 wird zuerst ein Überblick über die Grundlagen gegeben, die für Empfehlungsalgorithmen wichtig sind, woraufhin einige Arbeiten, die dem “Holiday Recommender” ähnlich sind oder sich in einem Teilbereich mit ihm überschneiden, vorgestellt werden. Dabei wird jeweils der Zusammenhang mit dem “Holiday Recommender” erläutert.

In Kapitel 3 wird die entwickelte Facebook-Applikation “Holiday Recommender” aus Benutzersicht beschrieben. Dabei wird ein genauer Einblick mittels Text und Bild gewährt, wie die Benutzer die Applikation bedienen können.

Aus technischer Sicht wird die entwickelte Facebook-Applikation “Holiday Recommender” in Kapitel 4 beleuchtet. Dabei werden kurz die grundsätzliche Funktionsweise sowie ausführlicher die wesentlichen Algorithmen, die den “Holiday Recommender” prägen, präsentiert.

Kapitel 5 widmet sich der Evaluierung des “Holiday Recommenders”. Die Benutzerfreundlichkeit dieser Applikation wurde im Zuge der Arbeit getestet und ausgewertet. Hier werden der Testaufbau, die Durchführung sowie die Ergebnisse präsentiert.

Zuletzt werden die erzielten Ergebnisse sowie die Probleme des “Holiday Recommenders” in Kapitel 6 diskutiert. Des Weiteren wird in diesem Kapitel eine Übersicht über sinnvolle Ideen zur Weiterentwicklung des “Holiday Recommenders” geboten.



# Kapitel 2

## Themenverwandte Arbeiten

Im Bereich “Recommender Systems” wurde auf Ebene von Reiseplanungen sowie Gruppenentscheidungen schon viel geforscht und entwickelt. Zuerst wird in diesem Kapitel ein Überblick über die Grundlagen gegeben. Dann werden einige Arbeiten, die dem “Holiday Recommender” ähnlich sind oder sich in einem Teilbereich mit ihm überschneiden, vorgestellt.

### 2.1 Grundlagen

Grundsätzlich gibt es mehrere Ansätze, Empfehlungssysteme zu konstruieren. Die bekanntesten davon sind:

- **Kollaborative Empfehlungen**

Hierbei werden Elemente gefiltert, die aufgrund der Bewertungen anderer Personen empfohlen werden können. Wenn viele Menschen etwas hoch bewerten, ist es wahrscheinlich, dass es auch der Person gefällt, für die eine Empfehlung erstellt werden soll. [Schafer u. a., 2007]

- **Inhaltsbasierte Empfehlungen**

Hierbei geht man von der Annahme aus, dass Elemente, die einem anderen sehr ähnlich sind, ähnlich bewertet werden würden. Wenn einem eine bestimmte Webseite über Tomatensauce gefällt, ist es wahrscheinlich, dass ihm auch eine andere Webseite über Tomatensauce gefällt [Schafer u. a., 2007]. Ähnlichkeiten werden dabei aus Kategorien, Genren und Schlüsselwörtern geschlossen [Jannach u. a., 2011].

- **Wissensbasierte Empfehlungen**

Sie kommen zum Einsatz, wenn Elemente vorgeschlagen werden sollen, deren Bewertungen schnell veralten (zB bei Technik), nur wenige Bewertungen vorhanden sind und der Benutzer genaue Anforderungen hat. Ein Beispiel wäre der Kauf einer Digitalkamera, wobei der Benutzer genaue Anforderungen an Preis, Auflösung, Zoom, Filmfunktion usw. hat [Jannach u. a., 2011].

- **Hybride Empfehlungen**

Hierbei werden verschiedene Stärken der vorigen drei Systeme miteinander kombiniert. Verschiedene Teile der Systeme können dabei kombiniert werden, um eines der Grundsysteme zu verbessern [Jannach u. a., 2011].

Inhaltsbasierte Empfehlungen können gemacht werden, ohne dass Bewertungen vorliegen müssen, während kollaborative Empfehlungen ohne Wissen über Eigenschaften auskommen [Schafer u. a., 2007]. Wissensbasierte Systeme brauchen keine vorliegenden Bewertungen, jedoch ein genaues Wissen über Eigenschaften der Elemente [Jannach u. a., 2011]. Die besten Ergebnisse können erzielt werden, wenn Teile und damit Stärken der Systeme miteinander kombiniert werden [Jannach u. a., 2011].

## 2. Themenverwandte Arbeiten

Um der Reisegruppe im “Holiday Recommender” Sehenswürdigkeiten vorzuschlagen, wird ein hybrides System verwendet, das die Vorteile von kollaborativen und inhaltsbasierten Empfehlungen vereint. Zunächst wird dabei nach Sehenswürdigkeiten gesucht, die einer bestimmten Kategorie entsprechen, an der der Benutzer interessiert ist. Aus den gefundenen Sehenswürdigkeiten werden dann diejenigen vorgeschlagen, die von Google-Nutzern am besten bewertet wurden.

Um ein Empfehlungssystem für Gruppen zu entwerfen, müssen vier Teilaufgaben gelöst werden [Jameson und Smyth, 2007]:

- **Erlangen von Informationen über die Interessen der einzelnen Mitglieder**

Zum einen können die Informationen explizit von den Gruppenmitgliedern eingeholt werden, indem einige Punkte bewertet werden müssen oder Interessen an bestimmten Kategorien oder Vergleichsdaten angegeben werden. Zum anderen kann man auch implizite Schlüsse aus Daten ziehen, die schon vorhanden sind.

- **Empfehlungen berechnen**

Die Empfehlungen werden generell anhand der Interessen der Gruppenmitglieder berechnet. Dabei gibt es verschiedene Methoden und Entscheidungsstrategien, die je nach Zusammensetzung der Gruppe erfolgreicher oder weniger erfolgreich sind und im Rahmen dieses Kapitels noch ausführlicher diskutiert werden. Grundsätzlich müssen die Empfehlungen so berechnet werden, dass alle Mitglieder zufrieden sind, sie als fair angesehen werden und sie nachvollziehbar sind.

- **Erklären der Ergebnisse**

Die Mitglieder wollen verstehen, wie die Empfehlung zustande kam und inwiefern die eigenen Interessen dabei berücksichtigt wurden, bzw. die Interessen anderer Mitglieder. Möglichkeiten wären, eine Analyse für die Individuen anzugeben oder für die ganze Gruppe oder für Subgruppen der Gruppe.

- **Unterstützung bei der Findung der finalen Entscheidung**

Dieser Punkt fällt bei Systemen weg, die eine einzige Lösung errechnen und präsentieren. Sind mehrere Möglichkeiten zur Auswahl, ist die einfachste Entscheidung, ein einziges Mitglied über die finale Lösung entscheiden zu lassen. Wird gemeinsam in der Gruppe innerhalb des Systems entschieden, können Hilfen wie animierte Persönlichkeiten oder weiteres Abstimmen über die angebotenen Empfehlungen ins System eingebunden werden.

Auch beim “Holiday Recommender” wurden diese vier Punkte in der Planungsphase überlegt. Um Informationen über die Interessen der Benutzer, an welchen Sehenswürdigkeiten sie interessiert sind zu erlangen, wurden neun Kategorien zur Bewertung bereitgestellt. Empfehlungen werden einmal für Sehenswürdigkeiten angeboten, wobei diese Empfehlungen für Einzelpersonen sind und einmal werden am Ende drei Routen empfohlen, aus denen die Gruppe eine wählen kann. Wie diese drei Routen zustande kamen, wird im Überblick über die jeweilige Route in der Applikation kurz erläutert. Um die Ergebnisse für die vorgeschlagenen Sehenswürdigkeiten zu erklären, wurde die dazugehörige Kategorie, aus der die Sehenswürdigkeit gewählt wurde, angegeben. Um eine Entscheidung, welche der drei angebotenen Routen gewählt werden soll zu erleichtern, können die drei Routen von allen Mitgliedern bewertet und kommentiert werden. Auf die genauen Details wird sowohl in diesem Kapitel als auch in den Folgekapiteln eingegangen.

Grundsätzlich gibt es zwei Methoden, um Empfehlungen für Gruppen zu errechnen [Boratto, Carta und Satta, 2010]:

- Individuelle Empfehlungen verschmelzen um daraus eine Empfehlung für die Gruppe zu machen
- Individuelle Präferenzen verschmelzen um daraus ein Gruppenprofil zu erstellen und dann Empfehlungen für die Gruppe zu machen

## 2.2. Gruppenempfehlungen unter Berücksichtigung sozialer Komponenten

Beim "Holiday Recommender" bewerten die Benutzer die eingebrachten Sehenswürdigkeiten, woraus nach drei verschiedenen Entscheidungsstrategien drei Routen berechnet werden. Es werden dabei also die individuellen Bewertungen der Sehenswürdigkeiten durch eine Entscheidungsstrategie miteinander verschmolzen. Daraus werden im Rahmen des Routenplaners drei Empfehlungen für die Gruppe errechnet, wobei zusätzlich zu den verschmolzenen Bewertungen auch die Entfernungen der Sehenswürdigkeiten sowie andere Nebenbedingungen eine Rolle spielen. Bei einer Reise ist es sinnvoll, den Benutzern aufgrund ihrer individuellen Interessen entsprechende Sehenswürdigkeiten vorzuschlagen. Ob diese auch wirklich den Geschmack des jeweiligen Benutzers treffen, kann der Benutzer dann selbst entscheiden. Ebenso hat jedes Mitglied die Möglichkeit, eigene Sehenswürdigkeiten einzubringen. Dabei wird gewährleistet, dass die Sehenswürdigkeiten, die für die Route grundsätzlich in Betracht gezogen werden, auch wirklich für die Gruppe relevant sind und auch, dass keine wichtigen Sehenswürdigkeiten fehlen. Die Qualität der Sehenswürdigkeiten wird dadurch optimiert. Durch diese Anforderungen ist es sinnvoller, alle Benutzer die eingebrachten Sehenswürdigkeiten bewerten zu lassen und diese Bewertungen zu verschmelzen, anstatt aufgrund der einzelnen Interessen ein Gruppenprofil zu erstellen, da dann die Selektion und Einbringung der Sehenswürdigkeiten durch die Mitglieder nicht mehr möglich wäre. Man könnte zwar anhand des Gruppenprofils Sehenswürdigkeiten für alle vorschlagen, die dann von den Benutzern aussortiert und ergänzt werden können, jedoch ist es sinnvoller, hierfür Empfehlungen für Einzelpersonen zu machen, da dadurch das Einbringen von Sehenswürdigkeiten für die jeweiligen Mitglieder interessanter und logischer ist.

## 2.2 Gruppenempfehlungen unter Berücksichtigung sozialer Komponenten

Masthoff [2004] hat in einem Experiment herausgefunden, dass eine Gruppe von Menschen, die sich gemeinsam ein paar Filme, die sie gemeinsam anschauen wollen, aussuchen, dabei gewissen Strategien folgen, um sicherzustellen, dass die Wahl fair ausfällt und dabei auch keine Person unglücklich gemacht wird. Da in einer Gruppe je nach deren sozialer Zusammensetzung andere Entscheidungskriterien schlagend werden, ist es wichtig, einige soziale Komponenten im Entscheidungsfindungsprozess zu berücksichtigen.

Gartrell u. a. [2010] verbessern Gruppenentscheidungssysteme dadurch, dass sie nicht nur die inhaltsbezogenen Interessen der einzelnen Gruppenmitglieder berücksichtigen, sondern auch die sozialen Eigenschaften der Mitglieder in das Entscheidungssystem miteinbeziehen. Das heißt, dass die einzelnen Bewertungen der Mitglieder nicht unbedingt gleich gewichtet werden, sondern spezielle Mitglieder das Ergebnis stärker beeinflussen können. Wie Amer-Yahia u. a. [2009] erläutern, gehen die Meinungen zu den einzelnen zu bewertenden Punkten in der Gruppe oft stark auseinander. Daher sollte auch eine Art Standardabweichung, die in den einzelnen Punkten besteht, in die Entscheidungsfindung miteinbezogen werden.

Um eine richtige Entscheidungsstrategie, also eine passende Formel, nach der die Entscheidung für die Gruppe berechnet wird, zu finden, müssen als erstes die Charakteristiken der Gruppen analysiert werden. Drei Gruppendeskriptoren sind besonders wichtig [Gartrell u. a., 2010]:

- **Deskriptor des Fachwissens**

Ist ein Experte in der Gruppe, werden sich die anderen Mitglieder von ihm beeinflussen lassen. Zum einen sind Experten oft redogewandter und haben mehr Überzeugungskraft auf ihrem Gebiet, zum anderen vertrauen die anderen dem Rat, den sie von einem, der sich auskennt, bekommen, wenn sie es selbst nicht besser wissen. Die Stimme eines Experten sollte daher im Entscheidungsprozess höher gewichtet werden.

- **Sozialer Deskriptor**

Wie gut sich die Gruppenmitglieder untereinander kennen, beeinflusst ebenfalls die Entscheidungsfindung. So sind Gruppenmitglieder, die sich untereinander sehr gut kennen und durch eine

## 2. Themenverwandte Arbeiten

enge Freundschaft oder Partnerschaft verbunden sind, eher bereit, die eigenen Interessen etwas in den Hintergrund zu stellen, um einer anderen Person dafür eine Freude zu machen. Sind die Mitglieder nicht verbunden miteinander, oder kennen sich womöglich gar nicht, erreicht man die größte Zufriedenheit dadurch, dass jede Stimme jedes Mitgliedes der Gruppe gleich viel zählt.

- **Deskriptor der Verschiedenheit**

Auch die Verschiedenheit der einzelnen Gruppenmitglieder beeinflusst die Entscheidungsfindung. Die Geschmäcker der Mitglieder sind oft unterschiedlich und so unterscheiden sich ihre Bewertungen der einzelnen Punkte auch oft stark voneinander. Wenn unterschiedliche Meinungen herrschen, hängt die Entscheidung davon ab, wie stark sie jeweils zu den einzelnen Punkten auseinander gehen.

Meiner Meinung nach ist es ebenso wichtig, Personen mit besonderen Bedürfnissen in der Gruppe zu berücksichtigen. So sollten ihre Stimmen, gleich wie bei Experten, eine höhere Gewichtung bekommen. Befindet sich zum Beispiel eine Person in der Gruppe, die im Rollstuhl sitzt oder deren Bein gebrochen ist und ein anderes Mitglied hat einen Ausflug in unwegsames Gelände vorgeschlagen, wird man der Bewertung der gehandicapten Person mehr Gewicht zukommen lassen, wenn diese Person diesen Ausflug sehr schlecht bewertet, da sie dorthin nicht mitkommen wird können. Ebenso sollte die Bewertung eines Kindes, das sich keinen Horrorfilm anschauen möchte, höher gewichtet werden, wenn es um eine Filmauswahl geht. Ardissono u. a. [2003] haben ein Empfehlungssystem für Sehenswürdigkeiten entwickelt, bei dem im Entscheidungsfindungsprozess die Bewertungen von Kindern und behinderten Personen stärker gewichtet werden.

Hat man die Gruppe identifiziert, kann man sich für eine bestimmte Entscheidungsstrategie entschließen. Die wichtigsten Entscheidungsstrategien sind [Gartrell u. a., 2010]:

- **Strategie des durchschnittlichen Willens (Average Satisfaction)**

Sie ist die naheliegendste, intuitivste Entscheidungsstrategie, die am besten dann eingesetzt wird, wenn die Gruppenmitglieder untereinander nicht sehr verbunden sind, da sie die fairste ist. Jede Bewertung wird hier für jeden Punkt gleich gewertet. Die Wertungen werden für jeden Punkt aufsummiert und durch die Anzahl der Wertungen dividiert; es wird also für jeden Punkt eine Durchschnittswertung errechnet. Beispiel: Liegen zu einem Punkt die Bewertungen 2, 4 und 5 vor, errechnet sich der Gesamtwert 3,66.

- **Strategie des geringsten Leides eines Mitgliedes (Minimum Misery)**

Hierbei wird auf jene Personen Rücksicht genommen, die an einem bestimmten Punkt überhaupt keinen Gefallen finden. Diese Strategie kommt am besten zum Einsatz, wenn die einzelnen Gruppenmitglieder sehr miteinander verbunden sind. Als Gesamtwertung für einen Punkt wird dazu einfach die schlechteste Bewertung dieses Punktes herangezogen. Alle anderen Bewertungen desselben Punktes sind für das Ergebnis damit irrelevant. Beispiel: Liegen wieder zu einem Punkt die Bewertungen 2, 4 und 5 vor, ergibt sich der Gesamtwert 2, wenn niedrigere Zahlen schlechtere Bewertungen bedeuten.

- **Strategie der größten Freude eines Mitgliedes (Maximum Satisfaction)**

Hierbei wird auf jene Personen Rücksicht genommen, die an einem Punkt besonderen Gefallen finden. Diese Strategie kommt am besten zum Einsatz, wenn sich ein Gruppenmitglied besonders für einen Punkt begeistert und dieser auch für die anderen akzeptabel wäre. Als Gesamtwertung für einen Punkt wird dazu einfach die beste Bewertung dieses Punktes herangezogen. Alle anderen Bewertungen desselben Punktes sind für das Ergebnis damit irrelevant. Beispiel: Liegen wieder zu einem Punkt die Bewertungen 2, 4 und 5 vor, ergibt sich der Gesamtwert 5, wenn niedrigere Zahlen schlechtere Bewertungen bedeuten.

## 2.2. Gruppenempfehlungen unter Berücksichtigung sozialer Komponenten

Weitere Entscheidungsstrategien sind [Masthoff, 2004]:

- **Strategie der Stimmenmehrheit (Plurality Voting)**  
Hier hat jedes Mitglied eine Stimme, die es dem am meisten bevorzugten Punkt gibt. Der Punkt mit den meisten Stimmen gewinnt. Beispiel: Person A stimmt für Punkt 1, Person B für Punkt 3, Person C für Punkt 3 und Person D für Punkt 2. Dann gewinnt Punkt 3 vor Punkt 1 und 2, die gemeinsam auf Platz 2 liegen.
- **Borda Zählung (Borda Count)**  
Jedes Mitglied reiht alle Punkte in einer Liste vom bevorzugtesten bis hin zum am wenigsten Gewollten. Entsprechend der Reihung werden für jeden Punkt Punkte vergeben. Die pro Mitglied erreichten Punkte werden dann aufsummiert. Beispiel: Reihung von Person A: 1, 3, 2, 4, Reihung von Person B: 3, 2, 1, 4, Reihung von Person C: 3, 1, 2, 4, Reihung von Person D: 4, 3, 2, 1. Platz 4 bekommt jeweils 0 Punkte, Platz 3 jeweils 1 Punkt, Platz 2 jeweils 2 Punkte und Platz 1 jeweils 3 Punkte. Für Punkt 1 errechnet sich der Wert  $3+1+2+0 = 6$ , für Punkt 2 der Wert  $1+2+1+1 = 5$ , für Punkt 3 der Wert  $2+3+3+2 = 10$  und für Punkt 4 der Wert  $0+0+0+3 = 3$ , womit Punkt 3 vor Punkt 1, 2 und 4 gewinnt.
- **Copeland Regel (Copeland Rule)**  
Die Bewertung eines Punktes errechnet sich durch die Anzahl der Fälle, wo der Punkt einen anderen schlägt, minus der Anzahl der Fälle, wo der Punkt gegenüber den anderen verliert. Beispiel: Es gelte die Reihung aus dem Beispiel der Borda Zählung. Punkt 1 ist zu Punkt 2 neutral, da Punkt 1 von Person A und C gegenüber Punkt 2 vorgezogen wird, von Person B und D aber nicht. Punkt 1 verliert gegenüber Punkt 3, da Punkt 3 von Person B, C und D gegenüber Punkt 1 vorgezogen wird und nur von Person A nicht. Punkt 1 schlägt Punkt 4, da Punkt 1 von Person A, B und C gegenüber Punkt 4 vorgezogen wird, nur von Person D nicht. Für Punkt 1 ergibt sich also:  $0-1+1 = 0$ . Analog ergibt sich für Punkt 2:  $0-1+1 = 0$ , für Punkt 3:  $1+1+1 = 3$  und für Punkt 4:  $-1-1-1 = -3$ . Somit gewinnt Punkt 3 vor den gleich platzierten Punkten 1 und 2. Punkt 4 erhält den letzten Platz.
- **Befürwortendes Stimmen (Approval Voting)**  
Die Mitglieder stimmen für jeden Punkt, mit dem sie einverstanden wären. Für jeden Punkt wird dann aufsummiert, wie viele Mitglieder mit ihm einverstanden wären. Beispiel: Person A ist mit Punkt 1, 3 und 2 einverstanden. Person B mit Punkt 3 und 2, Person C mit Punkt 3 und 1 und Person D nur mit Punkt 4. Punkt 1 erhält demnach 2 Stimmen, Punkt 2 ebenfalls 2 Stimmen, Punkt 3 erhält 3 Stimmen und Punkt 4 nur eine Stimme. Somit gewinnt Punkt 3 vor den gleich platzierten Punkten 1 und 2. Punkt 4 erhält den letzten Platz.
- **Strategie des Durchschnittes ohne Leid (Average Without Misery Strategy)**  
Eine neue Liste wird erstellt, wo für jeden Punkt die durchschnittliche Bewertung der Mitglieder vermerkt wird. Die Punkte, die von mindestens einem Mitglied schlecht bewertet wurden, werden dabei nicht auf die neue Liste gesetzt. Beispiel: Angenommen, man kann jeden Punkt mit 1 bis 5 Sternen bewerten, wobei 1 der schlechtesten Bewertung und auch 2 einer eher schlechten Bewertung entspricht. Dann wird kein Punkt in die neue Liste mit aufgenommen, der von zumindest einer Person mit weniger als 3 Sternen bewertet wurde. Punkt 1 wird von Person B mit nur 2 Sternen bewertet und Punkt 4 wird von Person B und C mit nur einem Stern bewertet, weswegen die Punkte 1 und 4 nicht in die neue Liste übernommen werden. Punkt 2 hat von den vier Personen die Bewertungen 3, 4, 4 und 3 bekommen, Punkt 3 die Bewertungen 4, 5, 5 und 3. Die neue Liste enthält nun auf Platz 1 Punkt 3 mit einem Durchschnittswert von 4,25 und auf Platz 2 Punkt 2 mit einem Durchschnittswert von 3,5.
- **Fairness Strategie (Fairness Strategy)**  
Von jedem Mitglied werden die beliebtesten Punkte ausgewählt. Sind Punkte gleich bewertet, wird die Meinung der andern Mitglieder mit betrachtet. Beispiel: Person A darf beginnen und hat Punkt 1 am besten bewertet, weswegen Punkt 1 an erster Stelle auf die Ergebnisliste gesetzt

## 2. Themenverwandte Arbeiten

wird. Person B hat Punkt 3 und 2 mit voller Punktezahl bewertet. Daher müssen die Bewertungen der anderen mit einbezogen werden: da Punkt 3 gegenüber Punkt 2 bei den anderen Personen besser abschneidet, rückt Punkt 3 auf Platz 2 in der Ergebnisliste. Durch Person C erhält Punkt 2 den dritten Platz und Person D hat Punkt 4 am besten bewertet weswegen dieser auf den nächsten Platz kommt. Gäbe es noch weitere Punkte, würde eine neue Runde eröffnet werden, bei der vorzugsweise die letzte Person aus der Vorrunde beginnen dürfte.

- **Strategie der respektiertesten Person (Most Respected Person Strategy)**

Die Bewertungen des am meisten respektierten Mitgliedes werden herangezogen. Hat dieses Punkte gleich bewertet, wird für eine Reihung die Bewertung der anderen mit einbezogen. Beispiel: Sei Person B die dominierende Person. Diese habe die Punkte 1 bis 4 mit den Bewertungen 2, 5, 5 und 1 versehen. Für Punkt 2 und 3 müssen für eine Reihung die Bewertungen der anderen Personen betrachtet werden, da diese beiden von der dominierenden Person mit der gleichen Bewertung (5) versehen wurden. Da Punkt 3 von allen anderen Personen gegenüber Punkt 2 besser oder gleich gut bewertet wurde, ergibt sich folgende Reihung: Punkt 3, Punkt 2, Punkt 1 und Punkt 4.

Gartrell u. a. [2010] haben in ihrer Arbeit eine Formel entworfen, die nach Berücksichtigung der oben genannten Deskriptoren eine der drei wichtigsten Entscheidungsstrategien auswählt und die weiters durch die Deskriptoren und der gewählten Entscheidungsstrategie eine für die Gruppe angepasste Ergebnisberechnung erreicht. Beim "Holiday Recommender" ist die Charakteristik der Gruppe oft schwer zu identifizieren, da viele unterschiedliche Gruppen gemeinsam auf Urlaub fahren können. Zumeist wird es eine Gruppe aus guten Freunden oder eine Familie sein, jedoch können es auch Arbeitskollegen sein, die einander nur flüchtig kennen und keine soziale Bindung untereinander haben, die gemeinsam eine Firmenreise bestreiten. Auch wird es völlig unterschiedlich sein, ob Experten oder Personen mit besonderen Bedürfnissen mit in der Gruppe sind. Für den "Holiday Recommender" wurde daher entschieden, drei Routen zu generieren, wovon eine den durchschnittlichen Willen der Gruppe, eine das geringste Leid eines Mitgliedes und eine die größte Freude eines Mitgliedes repräsentiert. Die Gruppe entscheidet sich anschließend für eine der drei vorgeschlagenen Routen.

Würde die Applikation selbst eine geeignete Strategie wählen, bräuchte man weitere vom Benutzer angegebene Daten, womit die Oberfläche komplexer werden würde und die Benutzer schon zu Beginn viel Zeit mit Dateneingaben verbringen müssten, was den "Holiday Recommender" mühsamer und unattraktiver lassen werden würde. Zu komplexe Systeme leiden unter der Benutzerfreundlichkeit, daher ist der Grundsatz "keep it short and simple" unbedingt zu beachten [Andrews, 2012a] [Andrews, 2006].

Die meisten Strategien benötigen bewertete Daten. Damit die Benutzer einen Punkt bewerten können, werden ihnen oft kurze Beschreibungstexte angeboten. Stettinger, Felfernig, Leitner, Reiterer und Jeran [2015] haben herausgefunden, dass bei diesen Beschreibungstexten die Reihenfolge, in denen die Plus- und Minuspunkte angezeigt werden, die Bewertung der Benutzer beeinflusst. Die Punkte am Beginn sowie am Ende des Textes fallen bei der Meinungsbildung viel deutlicher ins Gewicht als die Punkte, die in der Mitte des Textes stehen. Stettinger, Felfernig, Leitner, Reiterer und Jeran [2015] zeigen in einem Experiment, dass diese Beeinflussung abgemildert werden kann, indem von den Benutzern eine mehrdimensionale Bewertung (zB Bewertung von Erreichbarkeit, Ambiente, Kosten usw.) verlangt wird anstatt einer einzelnen Gesamtbewertungsangabe.

### 2.3 Identifikation von Gruppen

Bei Empfehlungssystemen für Gruppen geht man im Allgemeinen bereits davon aus, dass die einzelnen Gruppen bereits existieren, das heißt, dass sich die Menschen selbst bereits zusammengeschlossen und eine Gruppe gebildet haben. Manchmal müssen die einzelnen Gruppen aber erst gebildet werden.

Boratto, Carta und Satta [2010] präsentieren in ihrer Arbeit eine Methode für Gruppen-Empfehlungssysteme mit automatisierter Gruppenidentifikation. Grundsätzlich wird hier Clustering zur Gruppenidentifizierung verwendet, indem von Benutzern bereits bewertete Punkte herangezogen werden. Da oft

das Problem herrscht, dass nicht ausreichend viele Punkte evaluiert wurden, um ähnliche Personen mit gutem Ergebnis in Gruppen zusammenfassen zu können, wurde die Methode in dieser Arbeit weiter verfeinert. Dazu werden die fehlenden Daten mit einem Empfehlungssystem vorausgesagt und diese dann gemeinsam mit den vom Benutzer angegebenen Daten als Input für den Cluster-Algorithmus verwendet.

Um die fehlenden Daten gut vorhersagen zu können, wurde ein klassischer, benutzerbasierender Nearest-Neighbor Algorithmus, der auf collaborative Filtering basiert, eingesetzt. Das heißt, dass dieser Algorithmus eine Bewertung für einen nicht bewerteten Punkt berechnet, indem die Bewertungen von allen ähnlichen Benutzern (Nearest Neighbors) zu diesem Punkt betrachtet werden. Dadurch, dass nun durch das Voraussagen der fehlenden Daten für jeden Punkt für jeden Benutzer eine Bewertung vorliegt, können Gruppen viel erfolgreicher identifiziert werden.

## 2.4 Empfehlungsapplikationen auf Facebook

Auch in sozialen Netzwerken sind Empfehlungssysteme längst im Vormarsch. Auf Facebook gibt es bereits einige davon für verschiedene Gebiete, wie zum Beispiel die Applikationen “AppreciateApps”, “Free Games Recommended Applorer”, “Taste It! Try It!”, “Let’s Hangout!”, “Suggest Me a Movie” und “HappyMovie”.

### 2.4.1 Empfehlungsapplikationen für Einzelpersonen

“AppreciateApps” [Appreciate®, 2011] ist eine Applikation, die andere Facebook-Applikationen für eine bestimmte Person vorschlägt. Diese Empfehlungen basieren darauf, welche Applikationen die Person im Moment bevorzugt und verwendet, sowie auch darauf, welche Applikationen die Facebook-Freunde und Experten der Applikationen, an denen die Person interessiert ist, mögen und empfehlen.

Auch findet man auf Facebook den “Free Games Recommended Applorer” [Stream Media Pte Ltd, 2012], der ebenfalls personalisierte Empfehlungen von Facebook-Spielen bietet. Diese Anwendung wählt ihre Vorschläge anhand dessen, welche Spiele die Person spielt und wie häufig sie dies tut. Das Spielverhalten und die Spielgewohnheiten der Personen werden dabei analysiert, woraufhin weitere Spiele, die der Person gefallen könnten, empfohlen werden.

Eine Facebook-Applikation, die personalisiert Restaurants empfiehlt, bietet “Taste It! Try It!” [Łazaruk u. a., 2012]. Der Benutzer bewertet zuerst einige von ihm besuchte Restaurants. Anhand des Benutzerverhaltens und der Information, in welchen Restaurants der Benutzer hauptsächlich verkehrt (Regionale Küche, Internationale Küche, usw.), erstellt die Applikation ein persönliches Profil. Basierend auf diesem Profil empfiehlt die Applikation dem Benutzer für ihn passende Restaurants in seiner Nähe.

Für Facebook entwickelt wurde auch eine Applikation, die eine geeignete Gruppe vorschlägt, der der Benutzer beitreten könnte [Baatarjav, Phithakkitnukoon und Dantu, 2008]. Diese Applikation verwendet hierarchisches Clustering sowie Entscheidungsbaum-Techniken um die am besten passende Gruppe auszuwählen. Zuerst werden die Gruppeninformationen von Gruppen, die im Entscheidungsprozess berücksichtigt werden, analysiert. Dabei werden die persönlichen Daten der jeweiligen Gruppenmitglieder analysiert. Hierarchisches Clustering wird danach verwendet, um Ähnlichkeiten zwischen den Personen bzw. den Gruppen herauszufinden. Zum Schluss wird mit Hilfe eines Entscheidungsbaumes, in dem die einzelnen Gruppen verankert sind, entschieden, welche Gruppe am besten geeignet ist.

“Let’s Hangout!” [Chaudhari, Deshmukh und Mhaskar, 2013] ist eine gruppenbasierende Recommender-Applikation, die einer Gruppe von Freunden Filme und Konzerte vorschlägt, die sie aufgrund gemeinsamer Interessen zusammen besuchen können. Die Applikation filtert benutzerbezogene Daten aus dem Facebook-Profil des Anwenders und untersucht, ob es Filme oder Konzerte gibt, die den Anwender interessieren könnten. Wurde ein Event gefunden, werden Facebook-Freunde des Anwenders, die in der näheren Umgebung sind und die aufgrund ihrer Informationen ebenfalls an diesem Event interessiert sein könnten, gesucht. Das Event wird dann der gesamten gefundenen Gruppe vorgeschlagen.

## 2. Themenverwandte Arbeiten

“Suggest Me a Movie” ist eine Facebook-Applikation, die Filme empfiehlt [Cakiroglu und Birturk, 2010]. Hierbei werden die Empfehlungen basierend auf den Informationen des sozialen Umfeldes der Personen, die der zu beratenden Person am ähnlichsten sind, gemacht. Um einer Person Filme vorzuschlagen, werden zuerst beliebige Personen, die einen ähnlichen Geschmack haben, ausgesucht. Die Filmempfehlungen werden dann basierend auf den Daten und Geschmäckern der Freundeskreise der ausgesuchten Person errechnet. Hierbei spielt also das soziale Umfeld der einzelnen Personen eine große Rolle.

### 2.4.2 HappyMovie - Eine Empfehlungsapplikation für Gruppen

Eine weitere Applikation auf Facebook, die Filme empfiehlt, ist “HappyMovie” [Quijano-Sánchez, Recio-García und Díaz-Agudo, 2011]. Zum Unterschied zu “Suggest Me a Movie” ist sie kein Empfehlungssystem für Einzelpersonen, sondern eines für Gruppen, die gemeinsam einen Film im Kino anschauen wollen, was dem grundlegenden Charakter des “Holiday Recommenders”, wo es darum geht, eine Entscheidung für eine Gruppe von Personen zu finden, die gemeinsam etwas unternehmen wollen, entspricht. Bei “HappyMovie” werden in der Entscheidungsfindung auch die persönliche Bindung, die die einzelnen Gruppenmitglieder zueinander haben, sowie deren Persönlichkeiten mit berücksichtigt.

In der Applikation “HappyMovie” ist unter anderem auch ein Persönlichkeitstest integriert, den die Gruppenmitglieder machen können. Dieser dient dazu, den Typ der Person zu bestimmen, da jeder Persönlichkeitstyp in Konfliktsituationen unterschiedlich reagiert und somit auch unterschiedliche Entscheidungen im Lösungsfindungsprozess getroffen werden. Um eine für den Benutzer passende Lösung im Entscheidungsprozess erreichen zu können, ist es hilfreich, die Persönlichkeit des Benutzers zu kennen.

Der Vorteil, den Persönlichkeitstyp zu kennen ist, dass die Entscheidung, die ein Benutzer im realen Leben aufgrund seines Charakters vermutlich treffen würde, mit in den Entscheidungsprozess einfließen kann. Jedoch genügt es meiner Meinung nach nicht, nur die Charaktertypen der Mitglieder zu identifizieren, da auch andere soziale Faktoren, zum Beispiel, ob jemand ein Experte auf dem Gebiet ist oder gerade bei Filmen wichtig, ob jemand noch ein Kind ist, relevant für den Entscheidungsfindungsprozess sind.

Eine neue Studie von Youyou, Kosinski und Stillwell [2015] zeigt, dass es möglich ist, die Persönlichkeit einer Person exakt mit einem Computer zu bestimmen. Dazu haben sie ein Programm entwickelt, das die “Gefällt mir”-Angaben eines Facebook-Benutzers von dessen Facebook-Profil ausliest und verarbeitet, um die Persönlichkeit dieses Benutzers zu bestimmen. Bereits zehn Angaben reichen aus, um die Persönlichkeit besser einschätzen zu können, als es Arbeitskollegen üblicherweise können. 70 Angaben übertrumpfen gute Freunde und Mitbewohner, 150 übertrumpfen Familienangehörige und 300 sogar den Lebenspartner.

Bevor die Applikation “HappyMovie” einen geeigneten Vorschlag machen kann, welche Filme nun am besten passen, muss jedes Gruppenmitglied einige Filme bewerten. Die zu beurteilenden Filme sind ausgewählte Filme, die nicht mehr im Kino laufen. Anhand dessen, wie ein Benutzer die Filme bewertet, kann eine Abschätzung getroffen werden, wie er die Filme, die für einen gemeinsamen Kinobesuch zur Auswahl stehen, bewerten würde. Für die Beurteilung eines zur Auswahl stehenden Filmes wird der Durchschnitt der drei ähnlichsten Filme, die vom Benutzer bewertet wurden, genommen.

Der Vorteil, den Benutzer ein Kontingent an anderen Filmen bewerten zu lassen liegt darin, dass dies der Benutzer nur einmal machen muss. Diese Bewertungen können dann für jeden Kinobesuch verwendet werden, unabhängig davon, welche Filme gerade im Kino laufen. Der Nachteil ist, dass der Benutzer am Anfang eine große Menge an Filmen zu bewerten hat, die für den bevorstehenden ersten Kinobesuch auf den erste Blick nicht direkt relevant erscheinen und dieser Test viel Zeit in Anspruch nimmt. Vor allem, da es auch noch den Persönlichkeitstest gibt. Offen ist auch, wie der Benutzer reagieren soll, wenn er einen Charakter oder einen Film nicht kennt, denn dafür ist keine Option vorgesehen. Beim “Holiday Recommender” wird der Benutzer gebeten, seine Interessen an neun verschiedenen Kategorien von Sehenswürdigkeiten anzugeben, damit für ihn geeignete Sehenswürdigkeiten gefunden werden können,

anstatt ihn eine Menge an anderen Sehenswürdigkeiten bewerten zu lassen. Das geht weit aus schneller und ist zudem übersichtlich. Da der Benutzer selbst bestimmen kann, welche der vorgeschlagenen Sehenswürdigkeiten er tatsächlich wählt, reicht diese oberflächlichere Interessensangabe aus.

Zwischen allen Benutzern, die am gemeinsamen Kinobesuch teilnehmen, wird das Vertrauen, das sie jeweils zueinander haben, berechnet. Dieses Vertrauen errechnet sich daraus, wie eng die Teilnehmer untereinander befreundet sind, was sich mittels Facebook leicht auslesen lässt. Personen tendieren allgemein dazu, einem guten Freund eher zu glauben oder seine Meinung zu teilen, als Menschen, zu denen sie keinen Bezug haben.

Für jedes Gruppenmitglied wird jeder Film, der derzeit im Kino läuft, anhand einer geeigneten Entscheidungsstrategie, der persönlichen Bindung zu den anderen sowie der Persönlichkeit des Mitgliedes bewertet. Für diese Bewertung werden die Empfehlungen der Freunde des Mitgliedes, für den die Bewertung gerade aufgestellt wird, aufsummiert. Die einzelnen Summanden werden dabei um einen Faktor, der die Persönlichkeit jenes Mitgliedes widerspiegelt, deren Empfehlung gerade betrachtet wird, additiv erweitert. Dazu wird dann noch der Faktor, der die persönliche Bindung zwischen der Person, für die die Bewertung gerade erstellt wird und der Person, deren Empfehlung gerade betrachtet wird, dazu multipliziert.

Wenn ein Event erstellt wurde, werden sofort die drei für den Anleger passendsten Filme ausgewählt und angezeigt. Jedes Mal, wenn ein neuer Benutzer zu diesem Event dazukommt, werden die drei geeignetsten Filme aktualisiert, bis ein vom Anleger erstelltes Datum, an dem keine weiteren Teilnehmer mehr dazukommen dürfen, erreicht ist. Diese drei dann angezeigten finalen Filme werden danach von allen Teilnehmern bewertet und der am besten bewertete Film ist dann der, auf den die Entscheidung fällt.

Im Rahmen eines Tests mit einerseits realen Personen und andererseits fiktiven Charakteren wurde versucht, eine geeignete Entscheidungsstrategie zu finden. Dabei wurde festgestellt, dass für den Test mit realen Personen mit der Strategie des durchschnittlichen Willens die besten Ergebnisse erzielt wurden und für den Test mit den synthetischen Daten die Strategie des geringsten Leides einer Einzelperson die beste war.

Im Rahmen der Applikation wird also eine fixe Entscheidungsstrategie eingesetzt, anstatt eine der jeweiligen Gruppe am besten entsprechende zu wählen. Im "Holiday Recommender" werden drei Routen berechnet, die der drei erfolgreichsten Entscheidungsstrategien entsprechen. Die Benutzer können dann die Routen bewerten und gemeinsam entscheiden, welche der drei sie wählen, also im Grunde, welche Strategie sie bevorzugen.

### 2.4.3 GroupFun - Eine Empfehlungsapplikation für Gruppen

"GroupFun" ist ebenfalls eine Empfehlungsapplikation, die in das soziale Netzwerk Facebook integriert ist [Popescu und Pu, 2011], [Popescu, 2013], [Popescu, 2011]. Sie empfiehlt eine Musik-Liste für eine Gruppe von Personen. In der Applikation kann der Benutzer seine eigene Playlist zusammenstellen und alle Lieder bewerten. Auch kann er Freunde einladen und somit die Playlists der Freunde anschauen. Die Benutzer können ihren Freunden auch Musik vorschlagen und empfehlen. Das eigentliche Augenmerk der Applikation liegt auf der Partyliste, in der eine vom System errechnete Playlist an Liedern für eine bestimmte Gruppe geliefert wird, die die Gruppenzufriedenheit maximieren soll.

Für das Berechnen der vorgeschlagenen gemeinsamen Playlist, die den Geschmäckern aller Mitglieder, die gemeinsam Musik hören wollen, entsprechen soll, werden die Bewertungen der Lieder von den einzelnen Mitgliedern herangezogen. Die Mitglieder haben natürlich nicht alle Lieder, die im Gesamtsystem vorhanden sind, bewertet. Dies ist auch nicht notwendig, da die Mitglieder sich selbst schon die Musik herausgesucht und bewertet haben, die ihnen gefällt und die für sie in Frage kommt. [Popescu und Pu, 2011]

Von diesem Set an Liedern, die grundsätzlich für die einzelnen Mitglieder in Frage kommen, wird nun eine Gemeinschaftsbewertung für jedes einzelne Lied berechnet. Dazu werden alle Bewertungen al-

## 2. Themenverwandte Arbeiten

ler Lieder eines Benutzers zuerst auf die Summe 1 normiert. Dann werden für jedes Lied die normierten Bewertungen aller Mitglieder aufsummiert. Eine naheliegende Lösung wäre nun, die am besten bewerteten  $k$  Stück für die gemeinsame Liste auszuwählen. In dieser Applikation wird dies aus dem Grund nicht gemacht, da die Mitglieder dann ihre eigenen Bewertungen manipulieren würden. Das heißt, ein Benutzer, dem ein bestimmtes Lied besonders gut gefällt, das aber den anderen Mitgliedern nicht gefällt, würde dieses Lied nicht so hoch bewerten, da es dann aufgrund der schlechten Bewertung der anderen sowieso nicht unter die besten  $k$  Stück geraten würde. Stattdessen würde er lieber ein zweitliebstes Lied höher bewerten, wenn es auch den anderen gefällt, damit es dieses in die Auswahl schafft und somit das Ergebnis verfälschen. [Popescu, 2013]

Würde ein Benutzer zufallsbedingt ausgewählt werden, der sich das nächste gespielte Lied aussuchen dürfte, unabhängig davon, ob dieses den anderen gefällt oder nicht, würde er sich für sein Lieblingslied entscheiden. Aus diesem Grund hat Popescu [2013] in der Applikation eine mit Wahrscheinlichkeitsverfahren errechnete gewichtete Summe gebildet, die der zufallsbedingten Auswahl eines Mitgliedes, das sich das nächste Lied aussuchen darf, entspricht. Genau gesagt heißt das, dass die Wahrscheinlichkeit, dass Lied  $X$  in die gemeinsame Liste aufgenommen wird, berechnet wird, indem die Gemeinschaftsbewertung von Lied  $X$  durch die Summe aller Gemeinschaftsbewertungen aller Lieder dividiert wird. Das entspricht der Division durch die Anzahl der Mitglieder.

Der Vorteil der Methode mit Wahrscheinlichkeiten ist, dass auch ein Lied in die gemeinsame Liste aufgenommen werden könnte, das einer Person besonders gut gefällt, ohne dass es den anderen gefällt und diese Person dadurch glücklich gemacht wird und sie weniger dazu verleitet, die eigenen Bewertungen zu manipulieren. Ganz außer Kraft gesetzt ist das Problem mit der Manipulation allerdings nicht, da die Wahrscheinlichkeit für ein Lied durch die Normierung zur Bildung der Gemeinschaftsbewertung immer noch niedriger ist, wenn die Person viele Lieder gut bewertet, als würde die Person nur ein Lied wirklich gut bewerten. Je mehr Lieder von einer Person gut bewertet werden, desto geringer fällt dies in der Gemeinschaftsbewertung eines Liedes ins Gewicht. Man könnte allerdings auch so argumentieren, dass die Stimme von jemandem, dem viel gefällt, nicht so viel zählt, da dieser sowieso mit sehr vielen Liedern einverstanden ist und daher jemand, der sich nur für wenige ausgewählte Lieder begeistern kann, mehr Einfluss hat, um auch diesen zufrieden zu stellen.

Ein wirklicher Nachteil, der sich durch die Normierung zur Bildung der Gemeinschaftsbewertung ergibt, ist, dass die Stimmen einer Person, die im Gegensatz zu den anderen Mitgliedern deutlich mehr Lieder bewertet, weniger ins Gewicht fallen. Bewertet ein Mitglied zum Beispiel doppelt so viele Lieder wie ein anderes, und beide haben nur drei Lieblingslieder mit Höchstbewertung, dann fällt die normierte Bewertung für diese drei Lieder bei der Person, die doppelt so viele Lieder bewertet hat, deutlich niedriger aus als bei der Person, die weniger Lieder bewertet hat, was unfair ist.

Ein weiterer Nachteil bei dieser Methode ist, dass, zwar nur mit geringer Wahrscheinlichkeit, ein Lied in die gemeinsame Liste aufgenommen werden kann, das niemandem wirklich gefällt.

Beim "Holiday Recommender" werden die Bewertungen zur Bildung einer Gemeinschaftsbewertung nicht normiert, damit jede Stimme gleich zählt. Unabhängig davon, wie viel einem gefällt, kann damit eine Lösung gebildet werden, bei der die Stimme einer allgemein zufriedenen Person genauso viel zählt und diese daher genauso viel Mitbestimmung erhält, wie die Stimme einer schwer zufriedenen Person, da dies durchaus fair ist. Um aber eine Person, der nur wenige Sehenswürdigkeiten gefallen und diese dafür sehr gut, ungeachtet dessen, ob diese auch den anderen Mitgliedern gefallen oder nicht, zufrieden zu stellen, wird auch eine Route erstellt, die die größte Freude einer Person bezüglich einer Sehenswürdigkeit repräsentiert. Ebenso wird eine dritte Route erstellt, die Sehenswürdigkeiten vermeidet, die einer Person überhaupt nicht gefallen. Das hat den Vorteil, dass die Gruppe selbst entscheiden kann, ob sie einem Mitglied besonders entgegenkommen oder ob sie lieber die Route, die dem durchschnittlichen Willen entspricht, wählen wollen. Ebenso verlieren die hoch bewerteten Sehenswürdigkeiten eines Gruppenmitgliedes beim "Holiday Recommender" nicht an Gewicht, wenn dieses deutlich mehr Sehenswürdigkeiten bewertet als die anderen. Dadurch wäre eine Manipulation der eigenen Bewertungen nicht zielführend. Da man auch nicht weiß, welches nach welcher Strategie errechnete

Ergebnis am Ende ausgewählt wird, ist eine Manipulation der Daten ebenfalls nicht zielführend, da man nicht weiß, in welche Richtung man die Daten manipulieren soll. Da beim “Holiday Recommender” nicht mit Wahrscheinlichkeiten gerechnet wird, kann es auch nicht passieren, dass eine Sehenswürdigkeit in die Route aufgenommen wird, die niemanden wirklich begeistert, außer, es ist genug Zeit vorhanden, um sie trotzdem anzuschauen.

## 2.5 Urlaubsplaner auf Empfehlungsbasis im Web

Während der Autorin zum Zeitpunkt der Verfassung dieser Arbeit noch kein Urlaubsplaner auf Empfehlungsbasis bekannt ist, der in das soziale Netzwerk Facebook integriert ist, gibt es im Web jedoch schon einige Planungssysteme, mit Hilfe derer man sich Sehenswürdigkeiten vorschlagen lassen kann und/oder man sich eine geeignete Route berechnen lassen kann. Eine Auswahl aus diesen Systemen wird in diesem Kapitel vorgestellt.

### 2.5.1 e-Tourism

Sebastia u. a. [2009a] und Sebastia u. a. [2009b] haben “e-Tourism”, ein Empfehlungssystem für Einzelpersonen und auch für Gruppen entwickelt, das eine Liste von Touristenaktivitäten in Valencia (Spanien) vorschlägt. Diese Liste stellt einen Zeitplan dar, der die Distanzen zwischen den Aktivitäten berücksichtigt, sowie die Öffnungszeiten, jedoch keinen Routenvorschlag macht.

Grundsätzlich gibt es ein Kategorie-Modell, das in oberster Schicht Kategorien, wie zum Beispiel Sport enthält, denen wiederum genauere Unterkategorien untergeordnet sind, wie zum Beispiel Wandern oder Schwimmen [Sebastia u. a., 2009a]. In diesen Unterkategorien sind dann die einzelnen Aktivitäten enthalten mit einer Angabe, zu wie viel Prozent sie den Interessen des Benutzers entsprechen. Die Aktivitäten können dabei auch mehreren Kategorien untergeordnet sein.

Zu den einzelnen Benutzern werden persönliche Angaben gespeichert sowie deren Interessen [Sebastia u. a., 2009a]. Es werden demografische Informationen gesammelt (eine junge Mutter hat andere Interessen als ein Pensionist), inhaltsbasierende Informationen, die auf frühere Bewertungen von Aktivitäten dieses Benutzers zurückführen und generelle Interessen, die der Benutzer in seinem Profil spezifiziert hat. Zu jeder dieser drei Interessenskategorien wird ein Kategorie-Modell erstellt, das die jeweiligen Interessen an den Aktivitäten, die anhand der jeweiligen Interessenskategorie errechnet werden, speichert.

Um aus den einzelnen Profilen ein Gruppenprofil zu erstellen, werden drei verschiedene Verfahren angewendet: Aggregation, Schnittmengenberechnung und inkrementelle Schnittmengenberechnung [Sebastia u. a., 2009a]. In der Aggregationsmethode werden zuerst für jede Interessenskategorie (demografisch, inhaltsbasierend, generelle Interessen) alle jeweiligen persönlichen Präferenzen der Mitglieder zu einer Menge vereint, wobei für die Prozentangaben für das Interesse an jeder Kategorie (zB Sport) der Durchschnitt gebildet wird. Für jede der drei vereinten Mengen werden dann passende Aktivitäten gesucht. Diese drei Ergebnislisten werden zum Schluss vereint und dem Benutzer in einer gereihten Liste präsentiert.

Für die Methode der Schnittmengenberechnung werden für jede Interessenskategorie jeweils die Gemeinsamkeiten, die alle Gruppenmitglieder teilen, zusammengefasst. Es wird also die Schnittmenge der Interessen gebildet [Sebastia u. a., 2009a]. Das weitere Verfahren verläuft gleich wie bei der Aggregationsmethode.

Die inkrementelle Schnittmengenberechnung bildet einen gewichteten Durchschnitt über die Interessen der Gruppenmitglieder, wobei mit in Betracht gezogen wird, bei wie vielen Gruppenmitgliedern diese in den Präferenzen vorkommen [Sebastia u. a., 2009a]. Zuerst wird wie bei der Schnittmengenberechnung die Schnittmenge der Interessen gebildet, die alle Mitglieder teilen. Wie bei der Funktionsweise der Schnittmengenberechnung wird eine gereichte Liste zurückgegeben. In der nächsten Iteration wird die Schnittmenge der Interessen gebildet, die alle Mitglieder außer einem teilen und wieder gleich verfahren.

## 2. Themenverwandte Arbeiten

Diese Ergebnisse werden dann an die Ergebnisliste des ersten Durchganges angehängt. So wird weiter verfahren, bis auch die Interessen, die nur ein Gruppenmitglied betreffen, abgearbeitet sind.

Die Aggregationsmethode empfiehlt die für zumindest ein Mitglied meist priorisierten Attraktionen, was nicht bedeutet, dass diese Attraktionen für die gesamte Gruppe passend sind [Sebastia u. a., 2009a]. Bei der Schnittmengenberechnung herrscht das Problem, dass nur jene Attraktionen im Ergebnis berücksichtigt werden, die alle interessieren. Im schlimmsten Fall sind das gar keine, wenn es keine Kategorie gibt, die alle Mitglieder mögen. Je größer die Gruppen und je unterschiedlicher ihre Mitglieder sind, desto weniger Aktivitäten werden vorgeschlagen werden können. Die inkrementelle Schnittmengenberechnung vereinigt die Vorteile der Aggregationsmethode und die der Schnittmengenberechnung, da sie zuerst die Attraktivitäten reiht, die alle Mitglieder interessieren (wie bei der Schnittmengenberechnung), aber dann auch je nach gemeinsamer Interessen weitere Aktivitäten liefert. Auch das Experiment mit Testpersonen von Sebastia u. a. [2009a] zeigt, dass die inkrementelle Schnittmengenberechnung die besten Ergebnisse liefert.

Nachdem das Ergebnis dem Benutzer bzw. der Gruppe präsentiert wurde, wird von ihnen ein Feedback erwünscht [Sebastia u. a., 2009b]. Die Benutzer wählen dabei Aktivitäten aus der Liste aus, die sie wirklich besichtigen. Wenn sie sich das nächste Mal einloggen, werden sie gebeten, die Aktivitäten der letzten Empfehlung zu bewerten. Diese gewonnenen Daten werden dazu genutzt, um das Profil der inhaltsbasierenden Informationen zu verbessern. Auch wird mitprotokolliert, wie oft eine Aktivität vom Benutzer bzw. der Gruppe akzeptiert wurde. Das hat den Vorteil, dass mehr akzeptierte Aktivitäten anderen bei der Empfehlung vorgezogen werden können.

Dieses Konzept ist nur für eine eingeschränkte Menge von Aktivitäten oder Sehenswürdigkeiten anwendbar, da zu jeder Aktivität das Interesse des Benutzers berechnet und gespeichert werden muss. Das ist beim "Holiday Recommender" nicht möglich, da hierbei Sehenswürdigkeiten aus aller Welt vorgeschlagen werden können und dies viel Speicher benötigen würde sowie auch viel Zeit, bis zu den Sehenswürdigkeiten aussagekräftige Daten vorliegen würden. Da sich "e-Tourism" nur auf die Stadt Valencia in Spanien beschränkt, ist dies gut umsetzbar.

Das Paper von Sebastia u. a. [2009a] endet bei dem Vorschlag von Aktivitäten, die nach Relevanz in einer Liste sortiert sind. Sebastia u. a. [2009b] präsentieren einen weiterführenden Ansatz, wie aus einer Liste, die für eine Einzelperson erstellt wurde, eine passende Besichtigungsreihenfolge für einen Tag gemacht werden kann. Dabei wird eine passende Menge an Sehenswürdigkeiten ausgewählt, die sich an einem Tag in der verfügbaren Zeit ausgeht. Dazu werden Öffnungszeiten und auch eine Mittagspause mitberücksichtigt, sowie die Wegzeiten zwischen den einzelnen Aktivitäten. Um das Problem zu lösen, werden zwei verschiedene Methoden vorgestellt: Lösung mittels "Constraint Satisfaction Problem", indem man alle Bedingungen, die erfüllt sein müssen, in Nebenbedingungen ausdrückt und die Lösung mittels "Partial Satisfaction Planning Problem", wo einzelne Aktionen ausgedrückt werden wie "besichtigen von X", "gehen von X nach Y" usw. Es wird zwar beschrieben, wie diese beiden Methoden modelliert werden, jedoch fehlt die Beschreibung, wie das Problem damit dann genau gelöst wird. Daher bleibt offen, ob damit zeit- und wegoptimierte Pläne errechnet werden oder einfach die erste Lösung, die die Grundbedingungen erfüllt, als Ergebnis dient.

Bei "e-Tourism" werden grundlegend individuelle Präferenzen verschmolzen, um daraus ein Gruppenprofil zu erstellen und dann Sehenswürdigkeiten für die Gruppe zu empfehlen. Im Gegensatz dazu werden beim "Holiday Recommender" individuell Sehenswürdigkeiten empfohlen und deren Bewertungen dann verschmolzen, um daraus eine Empfehlung für die Gruppe zu machen, wodurch das Endergebnis genauer auf die wirklichen Interessen der Gruppe an den einzelnen Sehenswürdigkeiten abgestimmt ist.

"e-Tourism" schlägt, bezogen auf ein Benutzer- oder Gruppenprofil, eine nach Relevanz gereihte Liste von Sehenswürdigkeiten für eine Einzelperson oder Gruppe vor. Beim "Holiday Recommender" werden jedem Reisemitglied individuell Sehenswürdigkeiten aufgrund dessen Interessen vorgeschlagen, sodass das Mitglied selbst die Möglichkeit hat, interessante Sehenswürdigkeiten auszuwählen und den anderen vorzuschlagen. Dadurch wird im Gegensatz zu "e-Tourism" sichergestellt, dass der endgültige

Reiseplan keine Sehenswürdigkeiten enthält, die von keinem Mitglied als sehenswert empfunden wird. Im Gegensatz zu “e-Tourism” können beim “Holiday Recommender” auch eigene Sehenswürdigkeiten von den Mitgliedern eingebracht werden, wodurch sichergestellt ist, dass keine Sehenswürdigkeit von einer Person vermisst wird. Auch ist für die Gruppe beim “Holiday Recommender” nachzuvollziehen, warum eine Sehenswürdigkeit auf der Reiseliste ist, während es bei “e-Tourism”, wo die Sehenswürdigkeiten aufgrund des Gruppenprofils ausgewählt werden, schwierig für die Gruppe ist, die Entscheidung nachzuvollziehen.

Des Weiteren wird bei einer Gruppenreise beim “Holiday Recommender” ein genauer Reiseplan inklusiver grafischer Darstellung der Strecke mit Hilfe einer Karte präsentiert, der auch mehrtägig sein kann. Bei “e-Tourism” kann ein passender Reiseplan ausschließlich für eine Einzelperson und ohne Karte erstellt werden, sowie nur für einen einzelnen Tag, wobei der Benutzer selbst angibt, welche der ihm vorgeschlagenen Aktivitäten er gerne im Plan haben möchte (der Algorithmus sucht dann einige davon aus, die sich zeitlich ausgehen). Im Gegensatz dazu nimmt der “Holiday Recommender” voll automatisch die passendsten Sehenswürdigkeiten in den Reiseplan mit auf. Da bei “e-Tourism” alle Sehenswürdigkeiten kategorisiert sind und zu jeder Sehenswürdigkeit ein Prozentsatz errechnet wird, wie sehr eine Person an dieser interessiert ist, ist dieses System nicht global anwendbar und daher auf eine Stadt beschränkt, während die Reiserouten beim “Holiday Recommender” weltweit erstellt werden können.

### 2.5.2 Your Tour

“Your Tour” ist ein Empfehlungssystem, das eine mehrtägige, personalisierte Tour für Einzelpersonen plant. Laut Almeida [2008] wurde dieses System von Godart [2006] entwickelt. Ebenso verweisen Soufriaou und Vansteenwegen [2010] auf Godart [2001]. Hierbei werden passende Sehenswürdigkeiten sowie auch passende Hotels bei mehrtägigen Touren auf Basis der persönlichen Interessen vorgeschlagen. Auch ein Routenplan wird dabei erstellt. Das System ist auf bestimmte Regionen bestimmter Länder beschränkt und kann unter der Adresse <http://www.yourtour.com> [Natural Digit S.A., Belgien, 2015] eingesehen und verwendet werden.

Auf der Startseite wählt der Benutzer zuerst das Land und die Region aus, in der der Urlaub stattfinden soll, sowie den Startort und das Start- und Enddatum. Basierend auf diesen Daten kann sich der Benutzer eine Route vorschlagen lassen, die zunächst auf unpersonalisierten Standarddaten beruht. Der Benutzer kann aber auch gleich, bzw. auch noch nachdem er eine Route vorgeschlagen bekommen hat, seine Interessen spezifizieren. Dabei gibt er sein Interesse an bestimmten Kategorien von Sehenswürdigkeiten bekannt sowie Anforderungen an das Hotel, Budget, einen Besichtigungszeitraum und weitere Einstellungen. Basierend auf diesen Daten wird dann für jeden Tag eine personalisierte Tour zusammengestellt und präsentiert.

Das System ist dabei sehr übersichtlich und benutzerfreundlich umgesetzt. Nachdem eine Tour erstellt wurde, gibt es einen Karteireiter, der eine Übersicht über die Gesamttour bietet sowie auch für jeden Einzeltag einen eigenen Karteireiter, in dem genauere Details zu den Sehenswürdigkeiten, dem Hotel sowie den Preisen ersichtlich sind. Unter einem weiteren Karteireiter kann der Benutzer jederzeit seine Interessen und Anforderungen ändern. Auch der “Holiday Recommender” ist von der Grundstruktur ähnlich aufgebaut, da es auch hier einzelne Karteireiter zur besseren Übersicht gibt.

Obwohl man bei “Your Tour” die Anzahl der Mitreisenden auf der Startseite angeben kann, können die Interessen nur für eine Person spezifiziert werden. Somit ist die Personalisierung der Reise nur auf eine Einzelperson abgestimmt. Der “Holiday Recommender” verbessert dies, indem jedes Reisemitglied seine eigenen, individuellen Interessen angeben kann. Ebenfalls ist es beim “Holiday Recommender” im Gegensatz zu “Your Tour” möglich, dass jedes Gruppenmitglied eigene Sehenswürdigkeiten einbringen kann und jede Sehenswürdigkeit, die auf die Vorschlagsliste aufgenommen wird, von jedem Mitglied bewertet werden kann, wodurch die berechneten Reiserouten viel besser auf die Gruppeninteressen abgestimmt werden. Des Weiteren können alle Gruppenmitglieder mit dem System interagieren und werden vom System als Individuen angesehen. Während die Auswahl des Urlaubsziels bei “Your Tour” nur auf bestimmte Gebiete beschränkt ist, können beim “Holiday Recommender” weltweit beliebige Städte-

## 2. Themenverwandte Arbeiten

oder auch Streckenreisen gewählt werden. Streckenreisen (der Start der Reise ist ein anderer Ort als das Ende der Reise) sind bei “Your Tour” nicht möglich.

### 2.5.3 City Trip Planner

Der “City Trip Planner” [Vansteenwegen u. a., 2011] ist dem System von “Your Tour” sehr ähnlich und unterstützt ebenfalls Touristen dabei, mehrtägige Städtereisen basierend auf deren persönlichen Interessen zu planen, indem passende Sehenswürdigkeiten ausgewählt werden und auch ein Routenplan erstellt wird. Ebenso ist das System auf wenige Orte beschränkt, in diesem Fall auf fünf belgische Städte, deren Sehenswürdigkeiten mitsamt all ihrer Daten in einer eigenen Datenbank gespeichert werden. Im Empfehlungssystem werden dabei auch Öffnungszeiten und Mittagspausen berücksichtigt, wobei jede Sehenswürdigkeit an jedem Tag andere Öffnungszeiten haben kann. Jede Sehenswürdigkeit wird mit einem Typ und einer Kategorie versehen.

Wenn der Benutzer die Seite betritt und eine Stadt auswählt, wird ihm zuerst eine Route vorgeschlagen, die auf vorausgewählten Standarddaten basieren. Diese Daten kann der Benutzer dann personalisieren, indem er zuerst die Reiseeckdaten sowie seine persönliche Interessen an verschiedenen Kategorien bekannt gibt. Auch kann der Benutzer sein Profil verfeinern, indem er frei wählbare Schlüsselwörter angibt. Pro Reisetag muss er auch seine Startposition und seine Startzeit bekannt geben. Basierend auf diesen Daten wird dann für jede Sehenswürdigkeit im System eine Interessenskennzahl vorhergesagt und daraufhin eine passende Route berechnet. Der Benutzer hat dabei die Möglichkeit, maximal fünf Sehenswürdigkeiten auszusuchen, die er unbedingt in der Route inkludiert haben möchte.

Um die Route zu berechnen, bei der auch Mittagspausen, die keinen fixen Ort oder Zeitpunkt haben, einkalkuliert werden, wurde der Grundalgorithmus von Vansteenwegen u. a. [2009], der in Kapitel 2.6.2 vorgestellt wird, und auf dem das Routenplanungssystem des “Holiday Recommenders” basiert, erweitert und abgeändert. Die Lösung basiert auf einer iterativen Suche, die sowohl durch Greedy- als auch durch Random-Parameter beeinflusst wird. Per random wird für jede Iteration ein Greedy-Wert zwischen 0 und 1 berechnet. Für jede mögliche Sehenswürdigkeit, die eingefügt werden könnte, wird ein heuristischer Wert berechnet. Dann wird ein Grenzwert durch Multiplizieren der Differenz zwischen dem maximalen und minimalen heuristischen Wert mit dem Greedy-Wert errechnet. Aus den Sehenswürdigkeiten, deren Werte über diesem Grenzwert liegen, wird eine Sehenswürdigkeit per random ausgewählt und in die Reiseliste an die passendste Stelle eingefügt. Pro Iterationsschritt werden so viele Sehenswürdigkeiten an die jeweils beste Stelle eingefügt, bis keine mehr eingefügt werden kann. Pro Iteration wird also eine Lösungsmöglichkeit berechnet. Der Algorithmus liefert am Ende die beste davon. Um auch tägliche Mittagspausen einzufügen, wird eine virtuelle Sehenswürdigkeit ohne Ortsangabe, also ohne Wegzeiten, eingefügt.

Am Ende wird dem Benutzer die beste gefundene Route präsentiert, wobei ihm auch die nicht eingefügten Sehenswürdigkeiten extra angezeigt werden. Der Benutzer kann dann Sehenswürdigkeiten dazu- oder abwählen und sich eine neue Route berechnen lassen. Der “City Trip Planner” wurde bis zum heutigen Tag in der Benutzeroberfläche und bezüglich der integrierten Städte weiterentwickelt und kann unter der Adresse <http://www.citytripplanner.com> [CityTripPlanner, 2015] eingesehen und verwendet werden.

Bei diesem System werden schnell und einfach personalisierte Routen berechnet. Es zeichnet sich dadurch aus, dass bei der Routenberechnung auch Öffnungszeiten berücksichtigt werden und sogar Mittagspausen mit eingeplant werden können.

Da hier alle Sehenswürdigkeiten händisch ausgesucht, kategorisiert und mit allen notwendigen Daten versehen werden, ist das System in dieser Form nicht global anwendbar. Des Weiteren müssen die Daten in der Datenbank immer aktuell gehalten werden, was einen erheblichen Aufwand bedeutet. Für die Routenberechnung über mehrere Tage muss der Startort für jeden Tag im Vorhinein bekannt sein. Auch ist keine Rund- oder Streckenreise über mehrere Städte möglich, da das System auf eine Route innerhalb einer Stadt beschränkt ist. Eine Empfehlung, die für eine Gruppe von Reisenden angepasst wird, ist nicht möglich. All diese Punkte werden im “Holiday Recommender” verbessert.

### 2.5.4 INTRIGUE

Auch “INTRIGUE” (INteractive TouRist Information GUIdE) [Ardissono u. a., 2001] ist ein System im Web, das personalisierte Informationen über Touristenattraktionen bietet. Es hilft Gruppen dabei, eine Tour zu organisieren, wobei auf die Charakteristiken der Gruppe geachtet wird. Was “INTRIGUE” besonders auszeichnet ist, dass es auf Gruppenmitglieder wie Kinder oder Behinderte besonders Rücksicht nimmt [Ardissono u. a., 2002]. Das System ist für Turin und Umgebung (Italien) ausgelegt, wobei alle Informationen sowie Feature-Typen über die Attraktionen in einer Datenbank gespeichert werden [Ardissono u. a., 2001].

Die Gruppe wird als Menge von einzelnen in sich homogenen Subgruppen modelliert, anstatt jedes Mitglied als Einzelperson zu betrachten [Ardissono u. a., 2003]. Jede Subgruppe ist beschrieben durch ihre besondere Charakteristik, ihre Vorlieben und ihre Wichtigkeit bezüglich ihres Einflusses. Um Attraktionen zu empfehlen, werden sie zuerst separat für die Subgruppen gereiht und dann für die Gruppe kombiniert. Diese einzelnen, sortierten Listen werden dem Benutzer präsentiert: eine für ihn und je eine für jede Subgruppe, sowie auch eine Liste für die gesamte Gruppe. In der Liste für die gesamte Gruppe wird in kurzen Sätzen zu jeder Attraktion mit einer Begründung erklärt, für welche Subgruppe sie besonders gut passt und für welche sie nicht geeignet ist. Mit einem Klick auf die Attraktion werden alle Informationen inklusive einem Bild übersichtlich angezeigt. Unter einem eigenen Menüpunkt “zeige Agenda” wird ein genauer Reiseplan angezeigt, der jene Attraktionen beinhaltet, die der Benutzer ausgewählt hat. Dieser Reiseplan zeigt genau an, von wann bis wann die Attraktion besucht wird sowie auch Details zum Transfer zwischen den Attraktionen. Leider wird nicht erklärt, wie dieser Reiseplan zustande kommt.

Die Neuerung dieses Systems ist, dass Rücksicht auf Personen mit besonderen Bedürfnissen genommen wird. Des Weiteren gibt es eine Erklärung zu jeder Attraktion, für wen und aus welchem Grund sie geeignet ist oder nicht, was für Reisegruppen, die Personen mit besonderen Bedürfnissen beinhalten, äußerst praktisch ist, da man diese Informationen sonst nur schwer und mit viel Aufwand erhalten würde. So hat man alle diese Informationen übersichtlich dargestellt. Das Problem, das sich daraus ergibt ist, dass man diese Informationen zu allen Attraktivitäten sammeln muss, um sie dem Benutzer in dieser Art präsentieren zu können, was vermutlich ein Mitgrund ist, warum das System nur auf ein kleines Gebiet beschränkt ist.

Eine einzige Person der Gruppe interagiert mit dem System, weswegen sich die anderen Mitglieder auf diese Person verlassen müssen, dass die Angaben zu ihren Charakteristiken und Interessen richtig sind und die Person ihre Bedürfnisse und Wünsche in der Planung wirklich mit berücksichtigt und das Ergebnis nicht manipuliert. Wenn die Person vertrauenswürdig ist und die Aufgabe für ältere, behinderte oder zu junge Personen übernimmt, ist dies sinnvoll, wenn diese Personen nicht selbst mit dem System umgehen können. Ist die Gruppe aber aus Personen, die nicht gehandicapt sind, zusammengesetzt, wird Unzufriedenheit herrschen, wenn nur eine Person mit dem System interagiert und sie vom System nicht als Einzelpersonen behandelt werden. Daher wurde es beim “Holiday Recommender” vermieden, Mitglieder in Subgruppen zusammenzufassen. Auch wird jeder Person ermöglicht, selbst Sehenswürdigkeiten einzubringen und alle eingebrachten Sehenswürdigkeiten zu bewerten und somit aktiv mitgestalten und entscheiden zu können. Zusammenfassend kann man sagen, dass “INTRIGUE” für Gruppen, die hauptsächlich aus Personen mit besonderen Bedürfnissen besteht, geeignet ist, während der “Holiday Recommender” auf Gruppen ausgelegt ist, wo möglichst alle Mitglieder aktiv in die Mitgestaltung der Reise eingebunden werden.

### 2.5.5 Travel Decision Forum

Das “Travel Decision Forum” [Jameson, 2004], [Jameson und Smyth, 2007] ist ein Empfehlungssystem, das einer Gruppe dabei hilft, ein Hotel für den gemeinsamen Urlaub zu finden, das den Wünschen und Anforderungen aller Gruppenmitglieder entspricht. Dazu macht jedes Mitglied explizite Angaben zu seinen Interessen bezüglich des Hotels, wie zum Beispiel zu Hotel- und Raumausstattung, Sportmöglich-

## 2. Themenverwandte Arbeiten

keiten, Freizeitbeschäftigungen oder Wellnessangeboten. In jeder der genannten Kategorien sind einzelne Punkte zu bewerten, die angeben, wie wichtig ihnen diese sind. Dabei kann man auch die Bewertungen der anderen Mitglieder einblenden.

Jameson [2004] erklärt in seiner Arbeit, dass jedes Gruppenmitglied Interesse daran hat, die Beurteilungen der anderen zu kennen. Gründe dafür sind zum Beispiel:

- **Anstrengung zu sparen**  
Eine Kopierfunktion ermöglicht das Kopieren der Bewertungen eines anderen Mitgliedes.
- **Lernen von anderen Mitgliedern**  
Ein anderes Mitglied hat eventuell Wissen oder Erfahrung, das/die man selbst nicht hat.
- **Standpunkt und Verhalten der anderen mit zu berücksichtigen**  
Wenn man für einen Punkt zwei oder mehr Mitglieder braucht, wie zB zum Tennis spielen, sollten das die anderen wissen, damit sie ebenfalls ihr Interesse an diesem Punkt bekanntgeben.
- **Anpassung zu fördern, um das Erreichen von Einigung zu erleichtern**  
Wenn man seine Bewertungen jemand anderem anpasst, minimiert man damit Konflikte.

Zusätzlich zur Möglichkeit, die Bewertungen der anderen einzublenden und zu kopieren gibt es im "Travel Decision Forum" auch die Möglichkeit, kurze Kommentare zu den Bewertungen der anderen einzublenden. Kommentare zu den eigenen Bewertungen zu schreiben, die von den anderen gelesen werden können, hat zwei gute Gründe [Jameson, 2004], [Jameson und Smyth, 2007]:

- um Mitglieder davon zu überzeugen, ähnlich zu bewerten, indem man ihnen Informationen gibt, die sie vielleicht nicht haben.
- um seine Bewertung zu erklären und zu rechtfertigen.

Auch beim "Holiday Recommender" wurde die Option, die Bewertungen der anderen einzublenden, ermöglicht. Eine Umfrage von Jameson [2004] hat ergeben, dass es die meisten Mitglieder bevorzugen, die Bewertungen der anderen sehen zu können. Auch kann man beim "Holiday Recommender" den anderen Kommentare zu den eigenen Bewertungen hinterlassen. Standardmäßig sind diese Bewertungen und Kommentare ausgeblendet, falls ein Benutzer nicht beeinflusst werden will (wie auch beim "Travel Decision Forum"). Es ist sinnvoll, es den anderen mitzuteilen, wenn zum Beispiel eine Sehenswürdigkeit gerade renoviert wird und daher geschlossen ist oder die anderen darüber zu informieren, dass man einen Ausflug zu einer Pferdefarm aufgrund einer Allergie nicht mitmachen kann.

Ein Nachteil, die Bewertungen der anderen sehen zu können, ist, dass man die Bewertungen dadurch manipulieren kann [Jameson, 2004] bzw. sich stark beeinflussen lässt [Stettinger, Felfernig, Leitner und Reiterer, 2015]. Wenn so eine Manipulation vorliegt, operiert der Algorithmus mit falschen Daten, da er davon ausgeht, dass die ihm vorliegenden Daten auf den echten Interessen jedes Mitgliedes basieren [Jameson und Smyth, 2007]. Um Beeinflussungen abmildern zu können, könnte man die Bewertungen der anderen erst zu einem späteren Zeitpunkt zugänglich machen [Stettinger, Felfernig, Leitner und Reiterer, 2015]. Um eine Manipulation von Angaben ausschließen zu können, kann zur Berechnung der Ergebnisse neben der Strategie des durchschnittlichen Willens, die manipulierbar ist, zwischen zwei nicht-manipulativen Entscheidungsstrategien gewählt werden [Jameson, 2004]:

- anstatt des Durchschnittes mit dem Median rechnen. Bei drei Gruppenmitgliedern würde hierbei der zweithöchste Wert zur Berechnung herangezogen werden.
- mit dem Ansatz eines automatisiert designtem Mechanismus nach Conitzer und Sandholm [2002] und Conitzer und Sandholm [2003] zu rechnen. Hierbei wird die Strategie durch einen Algorithmus generiert, der optimal in Bezug auf das erwartete Ergebnis einer gegebenen Objective-Function ist.

Diese beiden Ansätze sind zwar nicht manipulierbar, jedoch haben beide Nachteile. Wenn man mit dem Median statt des Durchschnittes rechnet, erzielt man damit oft suboptimale Ergebnisse, da zum Beispiel nicht mit in Betracht gezogen wird, wenn eine Person einen Punkt sehr schlecht bewertet hat [Jameson, 2004]. Der Ansatz eines automatisierten Mechanismus ist oft asymmetrisch und nicht-deterministisch. Dadurch ist er für den Benutzer schwer nachzuvollziehen und anzunehmen. Es kann sogar passieren, dass damit für eine Gruppengesamtbewertung eines Punktes ein Wert errechnet wird, der höher oder niedriger ist, als ihn überhaupt irgendein Mitglied bewertet hat.

Beim "Holiday Recommender" wurde auf eine nicht manipulierbare Strategie verzichtet, da diese, wie beschrieben, deutliche Nachteile haben. Die Manipulierbarkeit ist beim "Holiday Recommender" aber stark eingeschränkt dadurch, dass drei Routen mittels drei verschiedener Strategien berechnet werden und die Benutzer zum Zeitpunkt der Bewertungsphase nicht wissen, auf welche Route - und damit, auf welche Entscheidungsstrategie - die letztendliche Wahl dann fallen wird, wodurch ein bewusstes Manipulieren der Bewertungen nicht sinnvoll wäre.

Es ist wichtig, dass die Gruppenmitglieder das errechnete Ergebnis nachvollziehen können. Dazu kann eine Analyse des Ergebnisses präsentiert werden, die etwa für die gesamte Gruppe, eine Subgruppe oder auch für die einzelnen Mitglieder gemacht werden kann [Jameson, 2004]. Im "Travel Decision Forum" werden zur Ergebniserklärung zwei Methoden angewendet:

- **Farbliche Hinterlegung zu allen Punkten**

Direkt in der Sektion, wo die Präferenzen der einzelnen zu bewertenden Punkte angegeben wurden, wird der Ergebniswert, der für die Gruppe errechnet wurde, gelb hinterlegt. Dadurch kann man sehen, wie weit die eigene Bewertung und auch die der anderen Mitglieder davon entfernt sind.

- **Verbale Argumente mit Hilfe von animierten Charakteren**

Da ein Mitglied unzufrieden sein könnte, wenn es bei einem Punkt weit entfernt von der vorgeschlagenen Lösung ist, wurden als zweite Methode animierte Charaktere entwickelt, die die einzelnen Gruppenmitglieder verkörpern. Ihre Ansichten und Argumente werden auf Grund der angegebenen Informationen der Präferenzen generiert. Ein Mitglied sieht durch diese Charaktere die verbalen Standpunkte der anderen, um diese besser verstehen zu können.

Die Charaktere haben den Vorteil, dass verbale Ausdrucksweisen für den Menschen meist besser interpretierbar sind als farbliche Markierungen. Ein Nachteil ist aber, dass diese auch ablenkend wirken und sehr zeitintensiv sind [Jameson, 2004]. Beide Ergebniserklärungen eignen sich gut in der Unterstützung dabei, den Mitglieder zu einer finalen Lösung zu verhelfen [Jameson und Smyth, 2007].

Auch beim "Holiday Recommender" wurde auf eine Erklärung des Zustandekommens der jeweiligen Route nicht ganz verzichtet. Oberhalb jeder Route befindet sich ein kurzer Text, der die Entscheidungsstrategie, mit der die Lösung errechnet wurde, erklärt. Es gibt jedoch keine direkte Erläuterung, wer bei welcher Sehenswürdigkeit profitiert und wer weniger. Dies kann in der errechneten Route mit einem Klick auf den Namen der in ihr enthaltenen Sehenswürdigkeit eingesehen werden, indem man sich dort die einzelnen Bewertungen der Mitglieder anschaut.

Zur Findung der finalen Lösung, also eines geeigneten Hotels, werden beim "Travel Decision Forum" ebenfalls die Charaktere, die die einzelnen Gruppenmitglieder verkörpern, verwendet [Jameson, 2004]. Dazu kann jedes Mitglied Angaben machen, auf welche anderen Mitglieder sie Rücksicht nehmen wollen und auf welche nicht. Auch können sie einstellen, wie tolerant sie gegenüber alternativen Lösungen sind. Zusätzlich gibt es noch die Möglichkeit, einzustellen, welches Mitglied gegenüber welchem anderen bevorzugt werden soll. Die stellvertretenden animierten Charaktere argumentieren dann je nach diesen Einstellungen. Für jeden zu bewertenden Punkt soll mit Hilfe der Charaktere ein Konsens gefunden werden. Sobald sich alle Mitglieder in einem Punkt einig sind, wird dieser vom System als Endlösung für diesen Punkt vermerkt. Wenn keine Einigung herrscht, haben die Mitglieder die Wahl, ihre Einstellungen zu ändern oder aber selbst einen Vorschlag einzubringen.

Diese Vorgehensweise, bei der zu jedem Punkt ein Konsens gefunden werden soll, ist für die einzelnen Mitglieder sehr zeitintensiv und mühsam. Indem man angibt, auf welche Mitglieder man gerne

## 2. Themenverwandte Arbeiten

Rücksicht nehmen will, kann man gut miteinbeziehen, wenn ein Mitglied besondere Bedürfnisse hat oder eventuell ein Kind ist. Gleichzeitig wirft das aber auch Probleme auf, wenn man angeben kann, auf wen man nicht Rücksicht nehmen will und wen man gegenüber wem bevorzugen möchte. Dies könnte zu erheblichen Spannungen in der Gruppe sorgen und damit den ganzen Urlaub vermiesen, weswegen dieser Lösungsansatz beim “Holiday Recommender” vermieden wurde. Da der Aufwand mit dem Einbringen der Sehenswürdigkeiten und deren Bewertungen schon hoch ist, wäre es nicht klug, für die Findung der Lösung noch einen zusätzlichen zeitaufwendigen Ansatz zu wählen. Lange Diskussionen sollen daher vermieden werden.

Um die endgültige Entscheidung, auf welche sich die Mitglieder beim “Holiday Recommender” einigen sollen zu unterstützen, können die vorgeschlagenen Routen von jedem Mitglied bewertet werden. Auch kann man über Kommentare zu den Routen ausdrücken, warum einem die Route gefällt oder nicht, sowie auf einen Vor- oder Nachteil der Route für ein anderes Mitglied hinweisen. Dies bietet eine schnelle, faire Alternative, die auch Spannungen, die durch direkte Reihungen, wen man gegenüber wem bevorzugt, entstehen, verhindert.

## 2.6 Routenberechnung

Um eine geeignete Route vorschlagen zu können, muss das Orientierungs-Problem gelöst werden. Das Orientierungs-Problem ist ein mathematisches Problem, das aus mehreren Örtlichkeiten (Sehenswürdigkeiten) besteht, deren Koordinaten bekannt sind und die mit einer Bewertung versehen sind [Souffriau und Vansteenwegen, 2010]. Wie schon erwähnt gibt es dabei zwei Arten von Orientierungsproblemen: eine, in der alle Orte in minimaler Zeit besucht werden müssen und eine, in der nicht alle Orte besucht werden müssen, jeder Ort jedoch eine Gewichtung besitzt und in einer gegebenen Zeit für die Gesamtroute die Auswahl der Orte so erfolgt, dass die Summe der Gewichtungen maximiert wird [Tsiligirides, 1984]. Golden, Levy und Vohra [1987] haben bewiesen, dass dieses Problem NP-Hart ist, weswegen eine Berechnung einer optimalen Lösung in polynomieller Zeit nicht möglich ist. Daher muss eine Näherungslösung angestrebt werden.

Die Berechnung einer geeigneten Näherungslösung für eine Route ist eine große Herausforderung, da sie von mehreren Faktoren abhängt. Für den “Holiday Recommender” muss dabei entschieden werden, welche Sehenswürdigkeiten in die Route aufgenommen werden, sodass die Gruppe zufrieden ist. Auch spielen die Entfernungen der Sehenswürdigkeiten eine Rolle sowie die verfügbare Zeit. Ebenso entscheidend ist, an welchen Tagen die Sehenswürdigkeiten geöffnet oder geschlossen haben. Diese zusätzlichen Anforderungen erschweren die Findung einer Lösung noch zusätzlich. In der Praxis ist die dazu am häufigsten verwendete Heuristik die “iterative lokale Suche”, bei der man iterativ Lösungen errechnet, wobei jede Lösung auf der vorherigen Lösung in einem reduzierten Bereich basiert [Lourenço, Martin und Stützle, 2003]. Hier werden einige Ansätze zur automatischen Berechnung einer Route vorgestellt.

### 2.6.1 Personalisierte Berechnung und Empfehlung einer Route

Rogers und Langley [1998] haben einen personalisierten Routenplaner entworfen, der eine für den Benutzer passende Route zwischen einem gegebenen Start- und einem gegebenen Endpunkt berechnet. Bei der Erstanwendung gibt der Benutzer seine Präferenzen bezüglich schnellster Zeit, kürzester Route, Abbiege-Anzahl und Anzahl der Teilabschnitte bekannt. Aus diesen vier benutzerbezogenen Werten ergeben sich die vier Gewichte der Kostenfunktion. Diese Kostenfunktion dient als User-Modell um Routen zu generieren, die zum Benutzer passen. Nach dem Lösungsvorschlag sowie während des Fahrens hat der Benutzer die Möglichkeit, Feedback zu geben, Änderung vorzunehmen und unter Berücksichtigung der neuen Daten eine neue Route berechnen zu lassen. Dadurch erhält das System mehr persönliche Daten, um die Kostenfunktion und damit das User-Modell zu verbessern. Die Ergebnisse zeigen, dass es möglich ist, eine den Wünschen des Benutzers entsprechende Kostenfunktion durch ständiges Benutzerfeedback zu erlernen.

Auch Niaraki und Kim [2009] haben einen personalisierten Routenplaner entworfen, der eine für den Benutzer passende Route zwischen einem gegebenen Start- und einem gegebenen Endpunkt berechnet. Zunächst wird ein User-Modell mittels demographischer Daten, Präferenzen des Benutzers sowie persönlicher und kontextueller Informationen erstellt. Ebenso wird ein Kontext-Modell erstellt, das Fahrzeugart, Verkehr, Wetter, Sicherheit, Infrastruktur und Touristenattraktionen berücksichtigt. Anhand dieser beiden Modelle werden Gewichte für die einzelnen Straßensegmente eines GIS-Straßennetzwerkes berechnet.

Diese beiden Verfahren sind auf die Präferenzen einer einzelnen Person optimiert. Man kann sie in dieser Form nicht gut auf eine Gruppe anwenden. Des Weiteren kann man keine Zwischenziele angeben und das Planungssystem ist nicht in der Lage, aus einer Anzahl an gegebenen Zwischenpunkten eine optimale Route zu generieren. Jedoch liefern sie eine gute Grundidee, den Routenplaner des "Holiday Recommenders" dahingehend zu optimieren, dass die Benutzer angeben können, ob sie gerne schnell reisen möchten, um schnell zur nächsten Sehenswürdigkeit zu kommen oder ob der Weg das Ziel sein soll und sie eine möglichst landschaftlich schöne Strecke ausgewählt bekommen wollen.

### 2.6.2 Methoden zur Routenberechnung

Sun und Lee [2004] haben ein Agenten-basierendes, personalisiertes Routen-Berechnungssystem für mobile Geräte entwickelt, das auf bestimmte GIS-Daten zugreift. Zuerst wird ein vektorbasierendes User-Modell auf Basis von persönlichen Interessen und geographischen Daten erstellt. Anhand dieses User-Modells werden dann personalisierte Touristenrouten berechnet. Jede Route besteht aus miteinander verbundenen Straßensegmenten, die in einem iterativen Prozess, bei dem die attraktivsten Straßensegmente in jedem Schritt basierend auf den Interessen des Benutzers ausgesucht werden, erstellt wird. Dazu werden zuerst attraktive Segmente mit Hilfe von GIS-Funktionen mittels Clustering ermittelt sowie Sehenswürdigkeiten in der Umgebung, die in einer Liste als potenzielle Kandidaten gespeichert werden. Dann wird ein Graph konstruiert, der Verbindungen zwischen allen Kandidaten enthält. Dazu wird eine von der GIS angebotene Funktion verwendet, die die kürzesten Verbindungen zwischen zwei Punkten findet. Diese Funktion errechnet ebenfalls die Kosten jedes Pfades (Entfernung, Wegzeit, usw.) sowie die Attraktivität des Pfades. Um auszusuchen, welche Kandidaten in die Route unter der Berücksichtigung der Wegkosten aufgenommen werden, wird der Näherungsalgorithmus "Tabu search" von Silver [2004] verwendet. Dieser startet mit einer möglichen, gültigen Route und ermittelt dann weitere Routen durch Erzeugung einer Nachbarschaft von Lösungen, wozu der erstellte Graph benutzt wird. Die beste davon wird ausgewählt und iterativ wird dann wieder eine Nachbarschaft von Lösungen erstellt, bis keine bessere Lösung mehr gefunden wird.

Dieses System ist für Einzelpersonen ausgelegt. Der "Holiday Recommender" kann auch mit Gruppen umgehen und erlaubt es im Gegensatz zu diesem Routen-Berechnungssystem, Routen für mehrere Tage zu generieren, sowie Öffnungszeiten und andere Nebenbedingungen mit in Betracht zu ziehen. Der Ansatz, schnell eine gültige Lösung zu finden und diese dann iterativ zu verbessern entspricht auch der Grundidee des "Holiday Recommenders".

Auch Castillo u. a. [2008] haben ein Agenten-basierendes personalisiertes Routen-Berechnungssystem für mobile Geräte, genannt "SAMAP", entwickelt. Es beschränkt sich dabei auf den Besuch einer bestimmten Stadt einer Einzelperson. Auch hier wird zuerst ein individuelles User-Modell erstellt. Danach werden Aktivitäten für die ausgewählte Stadt ausgesucht, die dem Benutzer gefallen könnten. Als Grundlage, um das Routenproblem zu lösen, das auch Öffnungszeiten, Wegzeiten bzw. Transportmöglichkeiten zwischen den Attraktionen sowie Transportkosten berücksichtigt, wird die Idee eines hybriden Systems von Fox und Long [2001], das aus mehreren Modulen besteht, angewendet. Das erste Modul wandelt die Liste der in Frage kommenden Aktivitäten in Prädikatenlogik um. Ein weiteres wählt eine passende Submenge an Attraktionen aus. Das Planungsmodul generiert mit Hilfe des Transportmoduls, das die Entfernungen und Wegzeiten kennt, eine Route. Für das Generieren der Route wird eine Baumsuche eingesetzt. Leider wurde "SAMAP" weder evaluiert noch sind die erzielten Ergebnisse bekannt.

## 2. Themenverwandte Arbeiten

Soo und Liang [2001] haben eine Reise-Applikation für mehrtägige Reisen entworfen, die aus einem Benutzerinterface- und einem Reise-Agenten besteht. Mit ersterem kann man persönliche Interessen spezifizieren wie Start- und Zielort, Start- und Endtag, maximales Budget sowie Hotelangaben. Des Weiteren kann man Sehenswürdigkeiten einbringen inklusiver Zeitangabe, wie lange man diese besichtigen möchte. Der Reise-Agent berechnet daraufhin eine Route, indem er drei Phasen durchläuft. Zuerst wird ein Gerüst gebaut, das leere Tagespläne enthält, in die jeden Tag ein Hotel eingefügt wird. Danach werden die bisherigen Kosten kalkuliert. In der zweiten Phase werden die Sehenswürdigkeiten in das Gerüst gefüllt. Dabei wird iterativ jene Sehenswürdigkeit als darauffolgende ausgewählt, die der vorherigen am nächsten ist. Ein gültiger Reiseplan muss alle Benutzeranforderungen erfüllen. Konnte eine Anforderung nicht erfüllt werden, weil zu viele Anforderungen oder unerfüllbare Anforderungen gemacht wurden, wird der Benutzer informiert und gebeten, seine Anforderungen entsprechend zu ändern, damit ein gültiger Plan entworfen werden kann. Ist der Benutzer am Ende mit dem Vorschlag nicht einverstanden, kann er diesen modifizieren.

Dieses System kann eine personalisierte Route über mehrere Tage inklusive Berücksichtigung von Hotels entwerfen, ist allerdings nur für Einzelpersonen und berücksichtigt keine Öffnungszeiten. Können nicht alle Sehenswürdigkeiten in den Plan aufgenommen werden, muss der Benutzer welche löschen, damit ein gültiger Plan erstellt werden kann. Selbst wenn das System selbst Sehenswürdigkeiten weglässt und man nicht den Benutzer bemühen würde, würden dadurch, dass immer die nächstgelegene Sehenswürdigkeit als nächste gewählt wird, Sehenswürdigkeiten aus dem Plan fallen, die der Benutzer gerne besucht hätte, da eine Gewichtung der Sehenswürdigkeiten nicht berücksichtigt wird. Es ist unklar, ob der Benutzer aus einer Menge von vorgeschlagenen Sehenswürdigkeiten auswählt oder selbst Sehenswürdigkeiten einbringen muss. Beim "Holiday Recommender" können die Gruppenmitglieder sowohl eigene Sehenswürdigkeiten einbringen als auch welche personalisiert vorgeschlagen bekommen und daraus welche einbringen. Das System wählt aus den so eingebrachten Sehenswürdigkeiten selbst passende für die Route aus, wobei sowohl die örtlichen Distanzen und Öffnungszeiten als auch die Gruppenbewertungen ausschlaggebend sind.

Hashimoto, Yagiura und Ibaraki [2008] haben einen Routenalgorithmus entworfen, der angenähert optimale Routen für eine bestimmte Menge an Fahrzeugen liefert, die die Aufgabe haben, eine Anzahl von Kunden zu besuchen. Jeder Kunde muss dabei exakt ein Mal besucht werden, das heißt, nur ein Fahrzeug hat einen bestimmten Kunden auf der Route. Dabei wird das Problem gelöst, welches Fahrzeug welche Kunden besucht sowie auch das Problem, dass die Kunden nur gewisse Zeitfenster zur Verfügung haben, an denen ein Besuch erlaubt ist. Die Routen sollen dabei natürlich zeit- und wegoptimiert sein. Es ist erlaubt, bei einem Kunden zu warten, bis sein Zeitfenster beginnt. Um die einzelnen Routen zu ermitteln, wird ein iterativer lokaler Suchalgorithmus verwendet. Bei einem iterativen lokalen Suchalgorithmus wird von einer gültigen Lösung ausgegangen, die dann iterativ verbessert wird, indem lokale Minima gesucht werden. In jeder Iteration wird dabei eine Nachbarschaftslösung ermittelt, bei dessen Evaluierung ein optimaler Zeitplan für jede Route berechnet wird. Dies wird mittels dynamischer Programmierung berechnet, die Teil des lokalen Suchalgorithmus ist. Der Ansatz des dynamischen Programmierens wurde dabei verbessert, indem Informationen aus der letzten Rekursion verwendet werden. Auch wird eine Filtermethode verwendet, die den Suchraum in der Nachbarschaft einschränkt, um keine Lösungen zu erhalten, die keine Verbesserungen bringen.

Dieses System ist dem des "Holiday Recommenders" sehr ähnlich, da für dessen Routensuche auch ein iterativer lokaler Suchalgorithmus verwendet wird, der für die Lösung dieses Problems sehr effizient ist. Die Unterschiede liegen aber im Detail der iterativen lokalen Suche, um die Orte einer Route optimal zu reihen: Hashimoto, Yagiura und Ibaraki [2008] suchen eine bessere Lösung in der Nachbarschaft, indem sie einige Orte einer Route mit einigen Orten einer anderen Route austauschen und schauen, ob dadurch eine Verbesserung stattfindet. Da der "Holiday Recommender" nur eine Route zu optimieren hat, kann kein Austausch mit anderen Routen stattfinden, weswegen beim "Holiday Recommender" Orte aus der Route entfernt, die Route dann zusammengeschoben wird und die entfernten Orte wieder eingefügt werden, wodurch eine neue Reihung der Orte entsteht. Die Routensuche von Hashimoto, Yagiura und Ibaraki [2008] kann wie der "Holiday Recommender" mit Zeitfenstern (Öffnungszeiten) umgehen, je-

doch erstrecken sich ihre Routen immer nur über einen einzelnen Tag und beginnen und enden immer am selben, vorher fixierten Ort, was beim “Holiday Recommender” komplexer ist.

Ein ähnliches System, genannt “P-Tour”, haben Maruyama u. a. [2004b] und [Maruyama u. a., 2004a] entwickelt, das eine Tour für einen Touristen berechnet. Shiraishi u. a. [2005] haben dieses weiterentwickelt indem der Suchalgorithmus, um Touren zu berechnen, verbessert wurde. Kinoshita u. a. [2006] haben dieses System dann dahingehend erweitert, dass eine Sightseeing-Tour über mehrere Tage gehen kann, wobei jeder Tag an einem anderen Ort beginnen und enden kann. Zur Berechnung der Route kommt hier ein genetischer Algorithmus zum Einsatz, der ebenfalls eine iterative, lokale Suche beinhaltet. Der Benutzer gibt zu Beginn die Anzahl der Tage bekannt, den Start- und Endort, die Start- und Endzeit sowie seine Präferenzen zu den Ausflugszielen. Zuerst wird die Region, in der die Tour stattfinden soll, in Subregionen zerteilt. Da die passenden Größen der Subregionen schwer automatisch zu wählen sind, kann der Benutzer die einzelnen Regionen einstellen. Pro Tag können nur Sehenswürdigkeiten aus maximal zwei Regionen besucht werden, um den Suchraum zu verkleinern. Jede Sehenswürdigkeit hat Öffnungszeiten sowie eine bestimmte Besuchszeit, die eingehalten werden muss. Die Distanzen zwischen den Sehenswürdigkeiten sind gewichtet, sodass weite Strecken bestraft werden. Auch stellt der Benutzer die ungefähren Regionen ein, in denen er an den einzelnen Tagen übernachten möchte. Der genetische Algorithmus zur Routengenerierung setzt sich aus der Generierung der Ausgangspopulation, Auswahl, Kreuzung, Mutation und lokaler Suche zusammen. Die letzten vier Schritte werden dann iterativ so lange ausgeführt, bis die Suchzeit abläuft [Maruyama u. a., 2004a].

In der ursprünglichen Version von Maruyama u. a. [2004b] wird erklärt, dass “P-Tour” den Benutzer auch mit Hilfe eines portablen Gerätes, das über GPS verfügt, navigieren kann und dass der Zeitplan automatisch neu berechnet wird, wenn das Gerät merkt, dass der Benutzer den gegebenen Zeitplan nicht einhalten konnte. Des Weiteren kann ein neuer Zeitplan berechnet werden, wenn der Benutzer Orte dazugeben oder löschen möchte oder den Aufenthalt an einem Ort verkürzen möchte [Maruyama u. a., 2004a].

Nagata u. a. [2006] haben “P-Tour” dahingehend erweitert, dass Routen für eine Gruppe von Personen berechnet werden. Diese können sich aufspalten, sodass Teilgruppen die Sehenswürdigkeiten sehen, an denen sie interessiert sind. Auch für diese Umsetzung wird ein genetischer Algorithmus angewendet. Das Ziel ist es, die Zufriedenheit aller Gruppenmitglieder zu maximieren, wobei sich die Zufriedenheit eines Mitgliedes erhöht, wenn Sehenswürdigkeiten gemeinsam besichtigt werden können und wenn bevorzugte Sehenswürdigkeiten in der Route sind. Es ist angedacht, den Algorithmus weiter zu verbessern, indem eine lokale Suchmethode eingeführt wird.

In der Version der Navigation, die nur für Einzelpersonen ist, muss der Benutzer vor der Routenberechnung Regionen wählen, in der er täglich Besichtigungen machen will sowie auch Regionen, in denen er übernachten will. Dadurch ist die Berechnung einer nahezu optimalen Route zwar beschleunigt, jedoch auch eingeschränkt, da vielleicht in einer Region sehr viele bevorzugte Sehenswürdigkeiten vorhanden sind und in einer anderen nur wenig, wodurch es passieren kann, dass schlechtere Sehenswürdigkeiten guten vorgezogen werden, da die erlaubte Region für diesen Tag keine besseren Sehenswürdigkeiten zur Verfügung hat. Ohne Einschränkung der Regionen könnte man sich vielleicht in einer Region länger aufhalten und eventuell eine andere Region schneller abhandeln und auch zwei Mal in einer Region übernachten und eine andere dafür nur tagsüber abhandeln. Dadurch, dass pro Tag Sehenswürdigkeiten aus zwei Regionen erlaubt sind und der Benutzer die Größen der Regionen selbst wählen kann, wird dieses Problem minimiert, jedoch setzt es voraus, dass der Benutzer genau weiß, wie viele Sehenswürdigkeiten ihn an welchen Orten interessieren und er dann auch die Übernachtungsregionen passend wählen kann. Die Erweiterung der Applikation bezüglich Gruppentauglichkeit erlaubt es Subgruppen, andere Sehenswürdigkeiten zu besichtigen, wenn Uneinigkeit in der Gruppe herrscht, was die Zufriedenheit der Personen erhöht. Bei großen Gruppen ist es durchaus sinnvoll, sich zwischendurch aufzuspalten. Kleinere Gruppen, die sich sehr nahe stehen, werden vermutlich eher zusammen bleiben wollen. Spaltet sich die Gruppe auf, hat man das Problem, dass Mitglieder eventuell bei einer Sehenswürdigkeit warten müssen, bis die anderen wieder zu ihnen stoßen.

## 2. Themenverwandte Arbeiten

Als Grundlage für den Routenalgorithmus des “Holiday Recommenders” dient die Arbeit von Vans-teenwegen u. a. [2009], in der ein Näherungsalgorithmus, um das Orientierungsproblem mit Zeitfenstern in sinnvoller Zeit zu lösen, präsentiert wird. Dazu wird ein iterativer, lokaler Algorithmus verwendet, bei dem durch Löschen, umsortieren und erneutes Einfügen versucht wird, lokalen Minima zu entkommen. Es sind dabei bewertete Orte gegeben, sowie eine Aufenthaltszeit und ein erlaubtes Zeitfenster für deren Besichtigung. Es können dabei Touren über mehrere Tage optimiert werden, wobei die einzelnen Tage als einzelne Tour gesehen wird, deren Start- und Endpunkt immer der selbe sein muss. Für den “Holiday Recommender” wurde dieser Algorithmus dahingehend verbessert, dass auch Rund- und Streckenreisen möglich sind, wo jeder Tag an einem anderen Ort beginnen und enden darf. Des Weiteren wird es auch ermöglicht, dass ein ganzer Tag nur ein Reisetag sein darf, ohne etwas zu besichtigen, bzw. auch eine Teilanreise an einem Tag und die restliche Anreise am nächsten Tag erfolgen darf. Genauere Details zur Funktionsweise werden in Kapitel 4.3 dargelegt.

## Kapitel 3

# Beschreibung der Applikation aus Benutzersicht

In diesem Kapitel wird die entwickelte Facebook-Applikation “Holiday Recommender” aus Benutzersicht beschrieben. Dabei wird ein genauer Einblick mittels Text und Bild gewährt, wie die Benutzer die Applikation bedienen können.

### 3.1 Grundlegende Funktionsweise und Überblick

Um auf den “Holiday Recommender” Zugriff haben zu können, muss man einen aktiven Account bei Facebook besitzen. Dazu kann man sich unter der Url <https://www.facebook.com/> registrieren. Registrierte Benutzer finden im Navigationsmenü den Punkt “Apps”, wo man sich dann spezielle Reise-Apps anzeigen lassen kann. Hier könnte dann auch der “Holiday Recommender” zu finden sein. Mit einem Klick auf die Applikation würde man direkt auf die Startseite des “Holiday Recommenders” kommen.

Grundsätzlich ist die Applikation in zwei Hauptseiten aufgeteilt: die Startseite und die Reiseseite, wobei letztere mit Hilfe eines Menüs in einzelne Unterseiten aufgegliedert ist. Auf der Startseite sieht man die eigenen bisherigen Reisen sowie eine Sektion, um neue Reisen anzulegen. Wählt man eine bereits angelegte Reise aus oder legt man eine neue Reise an, so gelangt man auf die Reiseseite, wo alle Daten zu dieser speziellen Reise gesammelt sind.

Da die Applikation für Reisen in der Gruppe ausgelegt ist, ist die Applikation in vier verschiedene Phasen unterteilt. In jeder Phase haben alle Mitreisenden die Möglichkeit, die in dieser Phase erlaubten Tätigkeiten auszuführen. Die vier verschiedenen Phasen sind:

- Einbringung von Sehenswürdigkeiten
- Bewertung von Sehenswürdigkeiten
- Berechnung und Bewertung der Routen
- Anzeigen der gewählten Route

Wäre die Applikation für alleine Reisende ausgelegt, wäre eine explizite Aufteilung in einzelne Phasen nicht notwendig. Bei Reisen in der Gruppe muss jedoch sichergestellt werden, dass alle Mitreisenden innerhalb einer gewissen Zeit die Möglichkeit haben, Vorschläge für Sehenswürdigkeiten einzubringen oder zu bewerten, bevor die empfohlenen Routenmöglichkeiten berechnet werden.

In die Applikation sind zwei Empfehlungssysteme eingebettet. Das erste erlaubt es den Benutzern, für sie persönlich interessante Sehenswürdigkeiten vorschlagen zu lassen. Genauere Informationen dazu

### 3. Beschreibung der Applikation aus Benutzersicht

findet man in Kapitel 3.2.4 und in Kapitel 3.2.5. Das zweite System ermittelt drei empfohlene Routen, die basierend auf den Benutzerbewertungen der Sehenswürdigkeiten, deren Entfernungen zueinander, sowie der verfügbaren Zeit berechnet werden. Genauere Informationen dazu findet man in Kapitel 3.2.6.

## 3.2 Detaillierte Beschreibung der Einzelkomponenten

In diesem Kapitel wird eine detaillierte Benutzeranleitung der Applikation “Holiday Recommender” wiedergegeben. In den folgenden Unterkapiteln 3.2.1 bis 3.2.7 werden die einzelnen Seiten bzw. Unterseiten in den einzelnen Phasen erläutert. Im letzten Unterkapitel 3.2.8 wird die Funktionsweise der Applikation anhand eines durchgehenden Beispiels näher gebracht.

### 3.2.1 Startseite

Startet man den “Holiday Recommender”, gelangt man als erstes auf die Startseite der Applikation. Diese bietet im Wesentlichen eine Übersicht über alle Reisen (Touren), die man bisher erstellt hat, bzw. bei denen man mitgereist ist oder mitreisen wird. Eine Veranschaulichung der Startseite ist in Abbildung 3.1 gegeben. In der oberen rechten Ecke der Startseite sieht man zwei Fahnen, eine deutsche und eine englische. Mit einem Klick auf eine der Fahnen kann man die Sprache der gesamten Applikation entsprechend ändern. Diese Änderung wird gespeichert und bleibt damit auch für zukünftige Anwendungen der Applikation erhalten. Standardmäßig ist die Sprache beim ersten Besuch auf die des Facebook-Profiles des Benutzers abgestimmt. Da bisher nur Deutsch und Englisch zur Verfügung stehen, wird bei allen Benutzern, die Deutsch als Standardsprache ihres Profils angegeben haben, Deutsch für den “Holiday Recommender” voreingestellt, für alle anderen Englisch.

Unterhalb der Sprachauswahl befindet sich die bereits erwähnte Liste aller bisherigen Touren. Beim ersten Besuch des “Holiday Recommenders” ist diese Liste noch leer. Neben jeder einzelnen Tour steht der Name der Person, die diese Tour angelegt hat. Mit einem Klick auf den Namen gelangt man zum Facebook-Profil dieser Person (in der aktuellen Version noch nicht implementiert). Mit einem Klick auf den Namen einer bestehenden Tour gelangt man auf die Verwaltungs-Hauptseite dieser Tour, die im folgenden Kapitel erläutert wird.

Zuletzt findet man auf der Startseite einen Bereich mit der Möglichkeit, neue Touren anzulegen. Dazu klickt man auf den dafür vorgesehenen Link, worauf sich ein Fenster öffnet mit der Aufforderung, einen Namen für die neue Tour einzugeben. Hat man einen Namen eingegeben und diesen mit OK bestätigt, gelangt man auf die Verwaltungs-Hauptseite der neuen Tour.



**Abbildung 3.1:** Die Startseite der Applikation “Holiday Recommender” in einem Beispielszenario.

### 3.2.2 Verwaltungs-Hauptseite

Wie in Kapitel 3.2.1 bereits erwähnt, gelangt man von der Startseite aus mit einem Klick auf eine bestehende Tour oder mit dem Anlegen einer neuen Tour auf die Verwaltungs-Hauptseite dieser jeweiligen Tour. Diese Hauptseite gliedert sich in zwei Bereiche: einen statischen Kopfteil und einen Teil mit Registerreitern (Tab Sheets). Abbildung 3.2 zeigt eine Abriss dieser Verwaltungs-Hauptseite.

Der Kopf ist in drei Spalten unterteilt. In der linken oberen Spalte ist ein Link platziert, mit dem man zur Startseite zurückkommt, um eine andere Tour zu laden oder anzulegen. In der mittleren Spalte befindet sich ein Dropdown-Menü, mit dessen Hilfe man sich alle Reisemitglieder dieser Tour anzeigen lassen kann (in der aktuellen Version noch nicht implementiert). Mit einem Klick auf das jeweilige Reisemitglied gelangt man auf das Facebook-Profil dieses Benutzers. In der rechten oberen Spalte kann man einen Facebook-Freund (das sind Personen, mit denen man auf Facebook befreundet ist) zur jeweiligen Tour einladen (in der aktuellen Version noch nicht implementiert).

Mit Hilfe des Menüs kann man die Details der Tour verwalten. Es gibt fünf Tab Sheets, die fünf verschiedene Teilbereiche der Tour abdecken:

- **Reisedaten**  
In diesem Bereich kann man die organisatorischen Details wie Reiseort, Zeitpunkt usw. verwalten.
- **Meine Interessen**  
Hier kann man seine persönlichen Interessen an Kategorien von Sehenswürdigkeiten angeben, um sich in weiterer Folge personalisierte Sehenswürdigkeiten anzeigen lassen zu können.

### 3. Beschreibung der Applikation aus Benutzersicht

- **Sehenswürdigkeiten**

In diesem Bereich kann man alle Sehenswürdigkeiten der Tour organisieren.

- **Reiseroute**

Hier werden nach der Bewertung der Sehenswürdigkeiten drei empfohlene Routen angezeigt.

- **Notizen**

In diesem Bereich kann jedes Reisemitglied Notizen zur Reise hinzufügen.

Mit einem Klick auf ein Tab Sheet kann man den jeweiligen Bereich anzeigen. Die Bereiche der einzelnen Tab Sheets werden in den folgenden Kapiteln 3.2.3 bis 3.2.7 genauer erläutert.



**Abbildung 3.2:** Abriss der Verwaltungs-Hauptseite einer Tour.

#### 3.2.3 Reisedaten

Unter dem Menüpunkt “Reisedaten” kann man die organisatorischen Details der Tour planen und verwalten. Eine grafische Darstellung des Inhaltes dieses Menüpunktes findet man in Abbildung 3.3. Zuerst kann man den Startort sowie den Endort der Urlaubsrouten eintragen. Die später berechneten Routen nehmen den Startort der Urlaubsrouten als Startpunkt für die Besichtigungen und den Endort der Urlaubsrouten als Endpunkt der Besichtigungen. Die Ortseingabe ist jeweils mit einer Google-Funktion versehen, die die gerade getippten Ortsnamen automatisch in einem Dropdown-Menü vervollständigt. Das garantiert, dass die eingegebenen Orte später auch mit Koordinaten für die Routenberechnung versehen werden können. Damit man sieht, ob der eingegebene Ort derjenige ist, den man auch wirklich gemeint hat, wird neben den Eingabefeldern eine Google-Karte angezeigt, in denen der ausgesuchte Ort sofort grafisch mit einem Marker verzeichnet wird.

Unterhalb der Ortsangaben folgen die Angaben des Ankommtages am Startort sowie des Abreisetages vom Endort. Diese Tage werden von der späteren Routenberechnung als Start- und Endtag für die Besichtigung der Sehenswürdigkeiten herangezogen. Für die Datumseingabe steht ein Eingabefeld mit Kalenderfunktion zur Verfügung. Bei einem Klick in dieses Eingabefeld öffnet sich automatisch der Kalender, mit dessen Hilfe man mühelos das gewünschte Datum auswählen kann. Ebenso hat man die Möglichkeit, das Datum ins Eingabefeld zu tippen, wenn es schneller gehen soll. Damit man nicht versehentlich den Abreisetag vom Endort vor dem Ankommtag am Startort wählt, sind im Kalender des Abreisetages die Tage vor dem Ankommtsdatum ausgegraut und somit nicht auswählbar. Ebenso kann man im Kalender des Ankommtages keinen Tag nach dem Abreisetag auswählen. Abbildung 3.4 zeigt ein Beispiel, wo die Reisegruppe am 15. Mai 2015 am Urlaubsziel ankommt. Daher kann beim Abreisetag kein Tag vor dem 15. Mai 2015 ausgewählt werden.

Reisedaten
Meine Interessen
Sehenswürdigkeiten
Reiseroute
Notizen

### Plane eure Tour

Startort der Urlaubsroute:

Endort der Urlaubsroute:



---

Ankunftstag am Startort:

Abreisetag vom Endort:

---

Startzeit für Besichtigungen:

Endzeit für Besichtigungen:

---

Bis wann dürfen Vorschläge für Sehenswürdigkeiten eingebracht werden?

[? Hilfe](#)

Wann ist die Bewertungsphase der Sehenswürdigkeiten abgeschlossen?

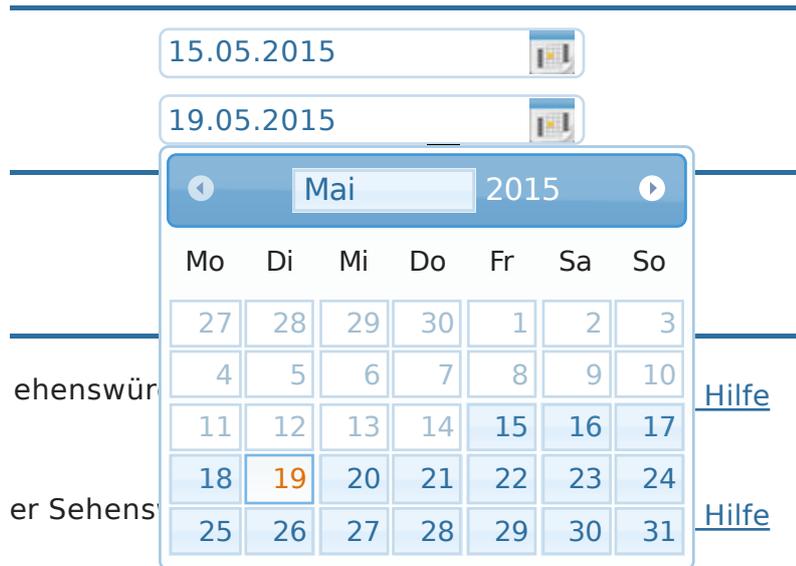
[? Hilfe](#)

**Abbildung 3.3:** Unter dem Menüpunkt “Reisedaten” kann man alle organisatorischen Details der Tour planen und verwalten.

Nach den Ankunfts- und Abreisedaten kann man die Start- und Endzeit der täglichen Besichtigungen angeben. Hierfür sind zwei Dropdown-Menüs bereitgestellt, mit Hilfe derer man die gewünschten Zeiten auswählen kann.

Als letztes müssen noch die Daten für die Zeitspanne der ersten beiden Phasen festgelegt werden. Zuerst kann mittels Kalenderfunktion mit kombiniertem Eingabefeld das Datum ausgewählt werden, bis wann alle Reisemitglieder Vorschläge für Sehenswürdigkeiten einbringen dürfen. Nach Ablauf dieser Frist können keine neuen Sehenswürdigkeiten mehr eingebracht werden und die Bewertungsphase beginnt. Zum besseren Verständnis der Bedeutung dieses Datums steht daneben auf der rechten Seite noch eine Hilfefunktion zur Verfügung. Mit einem Klick auf den Hilfe-Link öffnet sich ein Fenster, in dem eine kurze, genaue Erläuterung zu finden ist. Zuletzt steht eine weitere Kalenderfunktion mit kombiniertem Eingabefeld zur Verfügung, mit dem man das Datum für das Ende der Bewertungsphase auswählen kann. Bis zum angegebenen Datum können die davor eingebrachten Sehenswürdigkeiten von allen Mitreisenden bewertet werden. Nach Ablauf dieser Frist kann keine Bewertung mehr abgegeben oder geändert werden und die möglichen Routen werden automatisch vom System, abhängig von den Bewertungen, berechnet und vorgeschlagen. Auch hier ist zum besseren Verständnis wieder ein Hilfe-Link rechts daneben platziert. Wie auch schon bei der Auswahl des Ankunfts- und Abreisetages gilt auch

### 3. Beschreibung der Applikation aus Benutzersicht



**Abbildung 3.4:** Beispiel der Kalenderfunktion. Der Abreisetag vom Urlaubsziel darf nicht vor dem Ankunftstag am Urlaubsziel, also vor dem 15. Mai 2015, liegen.

für die Auswahl der Phasendaten, dass die Bewertungsphase nicht vor der Vorschlagsphase enden darf und die Vorschlagsphase nicht nach dem Ende der Bewertungsphase beginnen darf.

Am Ende der Reisedaten-Kategorie ist noch ein Button zu finden, um alle eingegebenen Daten zu speichern. Bevor die Daten in der Datenbank gespeichert werden, werden alle Angaben sowohl auf syntaktische als auch auf semantische Richtigkeit hin überprüft. Das heißt, es wird geprüft, ob Google die Orte bekannt sind, ob alle Datumsangaben gültige Daten sind und alle Uhrzeiten gültige Zeiten sind. Zusätzlich wird noch geprüft, ob die Vorschlagsphase vor der Bewertungsphase endet und diese beiden Daten wiederum vor dem Ankunftstag am Startort enden. Auch muss der Ankunftstag am Startort vor dem Abreisetag vom Endort liegen und alle Daten müssen in der Zukunft liegen. Wurde eine der Vorschriften verletzt, bekommt man automatisch eine entsprechende Fehlermeldung am Bildschirm präsentiert und wird zur Ausbesserung der jeweiligen falschen Daten aufgefordert.

Öffnet man den "Holiday Recommender", wenn die Vorschlagsphase um Sehenswürdigkeiten einzubringen schon beendet ist, dann ist das Datumseingabefeld für diese erste Phase ausgegraut und damit nicht mehr abänderbar. Ist auch das Ende der Bewertungsphase für Sehenswürdigkeiten verstrichen, sind alle Daten ausgegraut, sodass sie zwar noch gut sichtbar, jedoch nicht mehr editierbar sind. Ebenso wird der "Speichern"-Button nicht mehr angezeigt. Da nach Ablauf der Bewertungsphase bereits die Routen berechnet werden und diese nicht änderbar sind, hätte auch eine Änderung der Orte und Zeiten keine Auswirkung mehr, weshalb sie ausgegraut sind, um Irritationen zu vermeiden.

#### 3.2.4 Persönliche Interessen

Unter dem Menüpunkt "Meine Interessen" kann man seine persönlichen Interessen zu vorgegebenen Interessenskategorien angeben. Diese Kategorien beziehen sich auf die Interessen an verschiedenen Sehenswürdigkeiten. Anhand der Personalisierung der Interessen kann man sich dann unter dem Menüpunkt "Sehenswürdigkeiten" für einen selbst interessante Sehenswürdigkeiten in der Nähe des Urlaubsortes vorschlagen lassen. Die Sehenswürdigkeiten sind in folgende neuen Kategorien unterteilt:

- **Museen**

Diese Interessenskategorie enthält unter anderem folgende Unterkategorien: Freilichtmuseum, Historisches Museum, Kunstgalerie, Kunstmuseum, Museum, Naturkundemuseum, Planetarium, Skulpturenmuseum, Technikmuseum

- **Historische Stätten**

Diese Interessenskategorie enthält unter anderem folgende Unterkategorien: Amphitheater, Ausgrabungsstätte, Burg, Historisches Denkmal/Monument, Historischer Marktplatz, Historisches Museum, Historische Mühle, Historischer Ort, Historisches Schlachtfeld, Palast, Schloss und Garten, Statue

- **Kirchen und Klöster**

Diese Interessenskategorie enthält unter anderem folgende Unterkategorien: Andachtsort, Dom, Evangelische Kirche, Hindu Tempel, Katholische Kirche, Kirche, Kloster, Moschee, Synagoge

- **Flora und Fauna**

Diese Interessenskategorie enthält unter anderem folgende Unterkategorien: Aquarium, Botanischer Garten, Garten, Nationalpark, Naturpark, Park, Schaugarten, Tierpark, Zoo

- **Einkaufen**

Diese Interessenskategorie enthält unter anderem folgende Unterkategorien: Einkaufszentrum, Kaufhaus, Markthalle, Modegeschäft, Schuhgeschäft, Wochenmarkt

- **Sport**

Diese Interessenskategorie enthält unter anderem folgende Unterkategorien: Badensee, Badmintoncenter, Baseballplatz, Bowling, Eisbahn, Eissporthalle, Erlebnisbad, Fitnesscenter, Football-Platz, Freibad, Fußballplatz, Golfplatz, Hallenbad, Kegelbahn, Kletterhalle, Minigolfplatz, Outdoorkletteranlage, Rennbahn, Skateanlage, Sporthalle, Sportpark, Sportplatz, Squashanlage, Stadion, Tennishalle, Tennisplatz, Turnhalle

- **Spaß und Unterhaltung**

Diese Interessenskategorie enthält unter anderem folgende Unterkategorien: Erlebnisbad, Erlebnispark, Freibad, Freizeitpark, Go-Kart-Bahn, Hallenbad, Hochseilgarten, Indoorspielplatz, Kegelbahn, Kletterhalle, Minigolfplatz, Outdoorkletteranlage, Paintballanlage, Sommerrodelbahn, Spielhalle

- **Abendunterhaltung**

Diese Interessenskategorie enthält unter anderem folgende Unterkategorien: Autokino, Bar, Biergarten, Billardcafé, Bowlingbahn, Bowlingclub, Brauereigaststätte, Café, Cocktailbar, Coffeeshop, Diskothek, Freilichtbühne, Irish Pub, Jazz-Club, Karaoke-Bar, Kino, Kneipe, Konzertbühne, Nachtclub, Oper, Pub, Spielkasino, Tanzlokal, Theater

- **Touristenattraktionen**

Diese Interessenskategorie enthält unter anderem folgende Unterkategorien: Attraktion, Aussichtspunkt, Brücke, Fernsehturm, Rathaus, Sehenswürdigkeit, Springbrunnen, Touristenattraktion, Turm

Jede Kategorie ist grafisch in drei Spalten unterteilt. Links sieht man den Namen der Kategorie sowie einen Link, um die darin enthaltenen Unterkategorien anzuzeigen bzw. auszublenden. Die Unterkategorien beinhalten Stichwörter, nach denen Sehenswürdigkeiten auf Google für die jeweilige Kategorie gesucht werden. In der Mitte ist ein langer Balken mit einem Schieber, mit dem man den Wert des Interesses angeben kann. Rechts vom Schieber sieht man den Wert des Balkens in Prozent. Zum besseren Verständnis des Balkens ist in der linken oberen Ecke der Balken "wenig Interesse" notiert, sowie "großes Interesse" in der rechten oberen Ecke. Standardmäßig sind alle Balken am Anfang auf 50% gesetzt. Am Ende der Seite ist ein Button zu finden, um alle angegebenen Interessen in der Datenbank zu speichern. Ein Beispiel des Menüpunktes "Meine Interessen" ist in Abbildung 3.5 zu sehen.

Der gesamte Inhalt des Interessenmenüpunktes wird natürlich für jede Person individuell angezeigt, im Gegensatz zu allen anderen Inhalten, die für die ganze Reisegruppe gleich ist. Die Interessen können nur in der Vorschlagsphase, also der ersten Phase, eingestellt werden. Danach können keine Sehenswürdigkeiten mehr vorgeschlagen oder eingebracht werden, womit auch die Veränderung der Interessen

### 3. Beschreibung der Applikation aus Benutzersicht

irrelevant wäre. Daher sind in den weiteren Phasen die Interessensbalken ausgegraut. Ebenso verschwindet auch der “Speichern”-Button.

Kategorie	Interesse	Prozent
Museen	85%	85%
Historische Stätten	70%	70%
Kirchen & Klöster	85%	85%
Flora & Fauna	50%	50%
Einkaufen	30%	30%
Sport	43%	43%
Spaß & Unterhaltung	61%	61%
Abendunterhaltung	50%	50%
Touristenattraktionen	84%	84%

**Abbildung 3.5:** Unter dem Menüpunkt “Meine Interessen” kann man die persönlichen Interessen an bestimmten Sehenswürdigkeiten-Kategorien, um sich in weiterer Folge personalisiert ausgewählte Sehenswürdigkeiten vorschlagen zu lassen, einstellen.

#### 3.2.5 Sehenswürdigkeiten

Unter dem Menüpunkt “Sehenswürdigkeiten” kann die Reisegruppe alle Sehenswürdigkeiten, die die einzelnen Mitglieder eventuell besichtigen wollen, verwalten. Der Inhalt ist in drei Teilbereiche aufgeteilt: die Reiseliste, ein Bereich um sich Sehenswürdigkeiten vorschlagen zu lassen und ein Bereich um direkt eigene Sehenswürdigkeiten einzubringen. Ein Beispiel dieses Menüpunktes in der Vorschlagsphase ist in Abbildung 3.6 zu sehen.

Da die Liste der Sehenswürdigkeiten unter Umständen lange werden kann, sodass man nicht mehr den gesamten Inhalt dieses Menüpunktes auf dem Bildschirm sieht, sind als erstes auf der Seite drei Links zu den genannten jeweiligen drei Bereichen dieser Seite platziert.

Reisedaten
Meine Interessen
Sehenswürdigkeiten
Reiseroute
Notizen

## Übersicht der Sehenswürdigkeiten

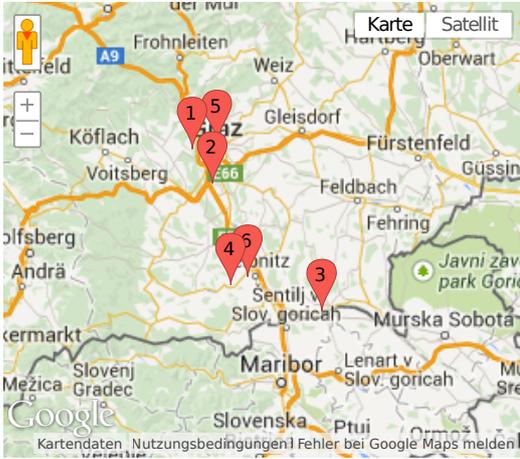
[Reisefliste](#)  
[Sehenswürdigkeiten vorschlagen lassen](#)  
[Sehenswürdigkeiten hinzufügen](#)

Reisefliste

Hier siehst du alle Sehenswürdigkeiten, die von dir oder deinen Freunden vorgeschlagen wurden

Aktuelle Liste:

- 1 [Lama-Garten, Hardter Straße, Thal, Österreich](#) ✘
- 2 [Schwarzlsee, Graz-Umgebung, Österreich](#) ✘
- 3 [Mühlenhof, Mureck, Österreich](#) ✘
- 4 [Ölmühle Hartlieb, Heimschuh, Österreich](#) ✘
- 5 [Landeszeughaus](#) ✘
- 6 [Hauptplatz, Leibnitz, Österreich](#) ✘



---

Sehenswürdigkeiten vorschlagen lassen

Klicke auf den Button rechts, um dir Sehenswürdigkeiten, die nach deinen spezifischen Interessen ausgewählt werden, vorschlagen zu lassen

Sehenswürdigkeiten vorschlagen

---

Sehenswürdigkeit hinzufügen

Füge deine eigene Sehenswürdigkeit zur Reisefliste hinzu

Gib einen Standort ein

Sehenswürdigkeit hinzufügen

**Abbildung 3.6:** Unter dem Menüpunkt “Sehenswürdigkeiten” kann die Reisegruppe alle Sehenswürdigkeiten, die auf der Reise eventuell besucht werden sollen, verwalten. In der Vorschlagsphase (Phase 1) können Sehenswürdigkeiten zur Reisefliste hinzugefügt oder gelöscht werden.

### 3. Beschreibung der Applikation aus Benutzersicht

#### 3.2.5.1 Reiseliste

Der erste Bereich beinhaltet die aktuelle Reiseliste, wo alle Sehenswürdigkeiten, die von einem selbst oder von anderen Reisemitgliedern eingebracht wurden, aufgelistet sind. In der Vorschlagsphase (der ersten Phase) ist neben jeder Sehenswürdigkeit, die man selbst eingebracht hat, ein “x” zu sehen, um diese Sehenswürdigkeit wieder aus der Liste zu entfernen. Jeder Mitreisende kann nur selbst eingebrachte Sehenswürdigkeiten löschen, um Fairness zu wahren. Sonst könnte ein Mitglied, das eine von einem anderen eingebrachte Sehenswürdigkeit nicht besuchen will, einfach löschen. Es soll der Gerechtigkeit wegen aber jeder Mitreisende ein Recht darauf haben, seine Vorschläge einzubringen. In den folgenden Phasen ist das “x” nicht mehr zu sehen, da es nur in der Vorschlagsphase erlaubt ist, Sehenswürdigkeiten zu löschen.

In der zweiten Phase, der Bewertungsphase, wird der Text “Bitte bewerte die Sehenswürdigkeiten, indem du auf die Bewertungs-Buttons in der Liste klickst!” in orange eingeblendet, damit die Benutzer gleich wissen, was jetzt zu tun ist. Unter jeder Sehenswürdigkeit ist ein Button eingeblendet, mit Hilfe dessen man direkt zur Bewertung dieser Sehenswürdigkeit kommt. Neben dem Button ist die bisherige eigene Bewertung grafisch in Form von fünf Sternen illustriert. Anhand dieser Sterne kann man schnell sehen, welche Sehenswürdigkeiten man bereits bewertet hat und wie. Mit ihnen sollte auch eine Schnellbewertung möglich sein, was in der derzeitigen Version aber noch nicht implementiert ist. Ein Beispiel, wie der Inhalt des “Sehenswürdigkeiten”-Menüpunktes in der Bewertungsphase aussieht, ist in Abbildung 3.7 zu sehen.

Nach Ablauf der Bewertungsphase bleibt die Reiseliste im Wesentlichen gleich wie zur Bewertungsphase, nur dass die Buttons, um zu den Bewertungen zu gelangen, nicht mehr angezeigt werden, da man seine Bewertungen nun nicht mehr ändern kann. Nach der Bewertungsphase werden bereits die Routen berechnet. Nach dieser Berechnung hätte eine Änderung einer Bewertung keine Auswirkung mehr auf die vorgeschlagenen Routen, womit die Änderung einer Bewertung nicht mehr erlaubt ist. Die eigenen Bewertungen selbst bleiben sichtbar, da sie hier übersichtlich dargestellt sind und somit hilfreich für die Bewertungen der Routen sind.

Rechts neben dieser Reiseliste ist in allen Phasen eine Google-Karte eingebettet, auf der alle Sehenswürdigkeiten aus der Reiseliste mit einem Marker grafisch eingetragen sind. Jeder Marker ist fortlaufend mit einer Nummer versehen. Fährt man mit der Maus über einen dieser Marker, sieht man den Namen der Sehenswürdigkeit, die sich an diesem Ort befindet. Um die Zuordnung zusätzlich besser erfassbar zu machen, ist ein Marker mit entsprechender Nummer auch vor dem Namen jeder Sehenswürdigkeit in der Reiseliste zu sehen. Die Karte wird nach dem Hinzufügen oder Löschen einer Sehenswürdigkeit automatisch skaliert, sodass immer alle Sehenswürdigkeiten darauf zu sehen sind. Man kann die Karte jedoch auch selbst mit den “-” bzw. “+”-Tasten in der Grafik verkleinern oder vergrößern. Ebenso hat man die Möglichkeit, auf Satellitenbilder umzuschalten. Zieht man das Männchen, das im linken oberen Eck der Karte angezeigt ist, über einen Bereich der Karte, so kann man sich auch Fotos dieses Ortes anzeigen lassen, die andere Google-Benutzer Google zur Verfügung gestellt haben. Klickt man auf den Namen einer Sehenswürdigkeit aus der Reiseliste oder auf einen Marker aus der Karte, öffnet sich ein Fenster mit genaueren Details zu dieser Sehenswürdigkeit. Ein Beispiel, wie so ein Detailfenster einer Sehenswürdigkeit aussieht, ist in Abbildung 3.8 gegeben.

Ist das Fenster zu den Details einer Sehenswürdigkeit geöffnet, sieht man als erstes links oben den Namen sowie die genaue Adresse inklusive Telefonnummer, Webseite und der Google-Seite für diese Sehenswürdigkeit. Mit einem Klick auf die Webseite öffnet sich ein eigener Tab im Browser, der die jeweilige Webseite dann anzeigt. Mit einem Klick auf die Google-Seite der Sehenswürdigkeit bekommt man einen neuen Browser-Tab, wo man weitere Details, die Google gespeichert hat, einsehen kann. Das sind zum Beispiel auch Erfahrungsberichte, Bewertungen und Fotos von anderen Google-Nutzern. Unterhalb des Google-Links sieht man im Detail-Fenster ein Foto von der Sehenswürdigkeit, wenn Google eines dazu hat. Wenn kein Foto verfügbar ist, wird ein Symbolbild mit dem Vermerk, dass es kein Foto gibt, dort gezeigt. Unterhalb des Fotos wird die passende Kategorie der Sehenswürdigkeit angezeigt, so sie sich zu einer Kategorie zuordnen lässt.

Reisedaten
Meine Interessen
Sehenswürdigkeiten
Reiseroute
Notizen

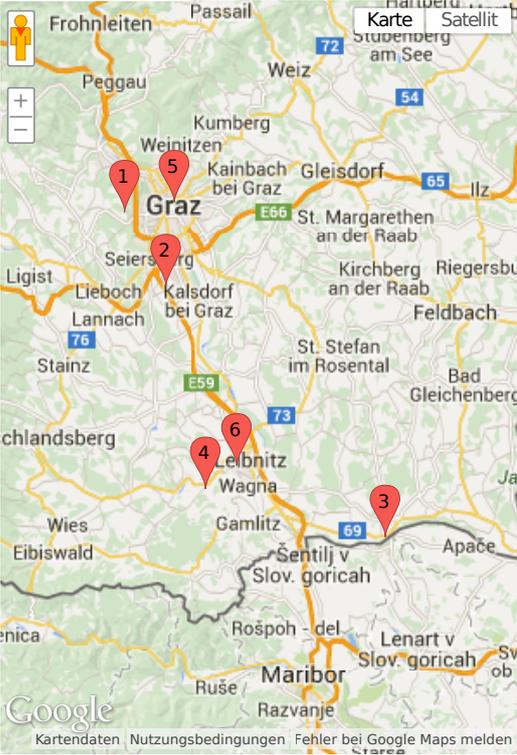
### Reiseliste

Hier siehst du alle Sehenswürdigkeiten, die von dir oder deinen Freunden vorgeschlagen wurden

Bitte bewerte die Sehenswürdigkeiten, indem du auf die Bewertungs-Buttons in der Liste klickst!

Aktuelle Liste:

- 1 [Lama-Garten, Hardter Straße, Thal, Österreich](#)  
Bewertung ★★☆☆☆
- 2 [Schwarzlsee, Graz-Umgebung, Österreich](#)  
Bewertung ★★☆☆☆
- 3 [Mühlenhof, Mureck, Österreich](#)  
Bewertung ★★★★★
- 4 [Ölmühle Hartlieb, Heimschuh, Österreich](#)  
Bewertung ★★★★★☆
- 5 [Landeszeughaus](#)  
Bewertung ☆☆☆☆☆
- 6 [Hauptplatz, Leibnitz, Österreich](#)  
Bewertung ☆☆☆☆☆



**Abbildung 3.7:** Unter dem Menüpunkt “Sehenswürdigkeiten” kann die Reisegruppe in der Bewertungsphase (Phase 2) alle eingebrachten Sehenswürdigkeiten bewerten.

Rechts neben diesem Kurzinformativblock ist eine Google-Karte eingebunden, die den genauen Ort der Sehenswürdigkeit markiert. Auch diese Karte lässt sich beliebig verkleinern oder vergrößern. Ebenfalls kann man sich die Karte als Satellitenbild anzeigen lassen sowie mit dem Männchen Umgebungsfotos anschauen. Unterhalb der Karte ist ein Button platziert, mit dem man die Sehenswürdigkeit auf seinem Facebook-Profil posten kann, damit auch Freunde, die nicht an der Reise teilnehmen, die Sehenswürdigkeit sehen können und somit auf den “Holiday Recommender” aufmerksam gemacht werden. Die Funktion, Sehenswürdigkeiten auf dem Profil posten zu können ist in der aktuellen Version noch nicht implementiert.

Darunter ist in der Vorschlagsphase (also so lange es den Mitgliedern erlaubt ist, Vorschläge einzubringen), ein zweiter Button platziert, mit dem man, je nach dem, diese Sehenswürdigkeit zur Reiseliste hinzufügen oder wieder löschen kann. Dieser Button wird nur angezeigt, wenn die Sehenswürdigkeit noch nicht auf der Reiseliste ist oder wenn man die Sehenswürdigkeit selbst hinzugefügt hatte. Ist die Sehenswürdigkeit bereits auf der Reiseliste und man hat sie nicht selbst hinzugefügt, darf man sie auch

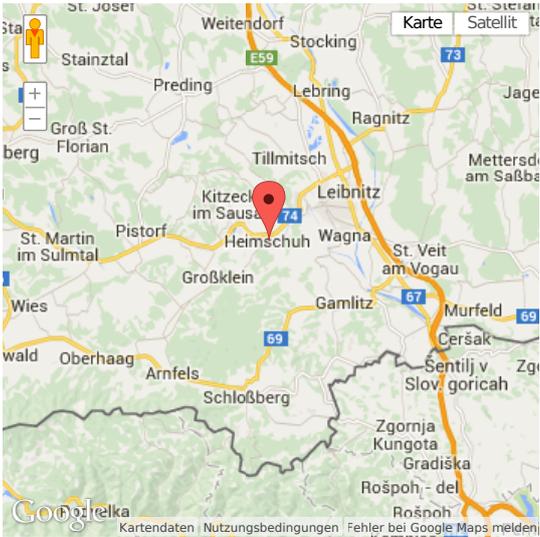
### 3. Beschreibung der Applikation aus Benutzersicht

#### Details

## Ölmühle Hartlieb

Heimschuh 107, 8451 Heimschuh,  
Österreich  
Telefon: +43 3452 82551

[hartlieb.at](http://hartlieb.at)  
[Sehenswürdigkeit bei Google](#)



**Kategorie:** Museen

Füge weitere Informationen zu dieser Sehenswürdigkeit hinzu:

[Sehenswürdigkeit auf deiner Pinnwand posten](#)

[Speichern](#)

---

Sehenswürdigkeit an folgenden Tagen für einen Besuch vorschlagen:

Mo  Di  Mi  Do  Fr  Sa  So

[Alle Tage wählen](#) [Keinen Tag wählen](#)

Gewünschte Aufenthaltsdauer:

[Speichern](#)

---

### Bewertung

Meine Bewertung:      Bitte bewerten

Kommentar:

[Speichern](#)

---

### Bewertungen der anderen

[Bewertungen anzeigen](#)

**Abbildung 3.8:** Zu jeder Sehenswürdigkeit, die man vorgeschlagen bekommt oder die in der Reiseliste ist, kann man sich Details anzeigen lassen und Informationen hinzufügen.

### 3.2. Detaillierte Beschreibung der Einzelkomponenten

nicht löschen, dies darf nur die Person, die die Sehenswürdigkeit eingebracht hat. Wird der Button angezeigt, wechselt er beim Hinzufügen der Sehenswürdigkeit in einen Löschen-Button und umgekehrt.

Ist die Sehenswürdigkeit noch nicht auf der Reiseliste, wird im Details-Fenster darunter nur noch ein Vermerk angezeigt, dass man in der Bewertungsphase (Phase 2) dann hier die Sehenswürdigkeit bewerten können wird, inklusive dem Datum, ab welchem diese Bewertung zur Verfügung stehen wird. Dieser Vermerk ist auch am Ende der Seite sichtbar, wenn die Sehenswürdigkeit bereits auf der Reiseliste ist. Ist die Sehenswürdigkeit bereits auf der Reiseliste, werden davor aber noch weitere Blöcke sichtbar. So wird unterhalb des Kurzinformationsblocks ein Textfeld angezeigt, wo jedes Mitglied weitere Informationen zu dieser Sehenswürdigkeit hinzufügen kann. Hat jemand schon einen Text hineingeschrieben, so ist dieser natürlich für alle anderen Reisemitglieder sicht- und editierbar. Um die Informationen zu speichern, ist ein "Speichern"-Button unterhalb des Textfeldes platziert.

Auch ein weiterer Bereich ist im Falle dessen, dass die Sehenswürdigkeit bereits auf der Reiseliste ist, sichtbar: ein Bereich, um Besichtigungsdetails einzustellen. So kann man als erstes jeden Wochentag mittels Checkboxen anhaken, an dem man diese Sehenswürdigkeit gerne besuchen möchte. Standardmäßig sind alle Wochentage ausgewählt. Soll eine Sehenswürdigkeit an einem bestimmten Tag nicht besucht werden (zum Beispiel, da sie geschlossen hat), und ist dieser Wochentag abgewählt, wird sie in der später berechneten Reiseroute auch nicht an diesem Tag vorgeschlagen. Um die Aus- und Abwahl der Tage zu beschleunigen, werden dafür zwei Buttons darunter angeboten: einer, um alle Tage auszuwählen und einer, um alle Tage abzuwählen. Darunter wiederum kann man noch die gewünschte Aufenthaltsdauer aus einer Dropdown-Liste auswählen. Auswählen kann man Zeiten zwischen 15 Minuten und dem ganzen Tag. Wählt man den ganzen Tag aus, so wird die Tageslänge bei der Routenberechnung auf den in den Reisedaten angegebenen Besichtigungszeitrahmen gesetzt. Denn hat man als Besichtigungszeit unter den Reisedaten angegeben, dass man nur von 9 Uhr bis 11 Uhr Sehenswürdigkeiten besichtigen möchte, dann wird der "ganze Tag" bei der Routenkalkulation mit 2 Stunden gerechnet. Am Ende des Bereiches für Besichtigungszeiten ist noch ein "Speichern"-Button angebracht, um alle angegebenen Zeiten speichern zu können.

In der Bewertungsphase (Phase 2) werden aus dem zuletzt angezeigten Vermerk mit dem Bewertungshinweis zwei Bereiche, um zum einen die Sehenswürdigkeit selbst bewerten zu können und zum anderen die Bewertungen der anderen Mitglieder einsehen zu können. Im Bereich der eigenen Bewertung sind fünf Sterne in einer Linie angeordnet, mit Hilfe derer die Bewertung abgegeben werden kann. Hat man die Sehenswürdigkeit noch nicht bewertet, werden die fünf Sterne in zartem Grau angezeigt mit einer Bitte um Bewertung daneben. Fährt man mit der Maus über die Sterne, so werden diese einzeln, je nach Mausposition, gelb eingefärbt. Neben den Sternen steht der jeweilige kurze Text, was die Anzahl an Sternen zu bedeuten hat. Ein Stern bedeutet "Nicht sehenswert", fünf Sterne bedeuten "Sehr sehenswert". Mit einem Klick bestätigt man die Bewertung, die man aber jederzeit während der Bewertungsphase wieder ändern kann. Unterhalb der Sterne hat man die Möglichkeit, einen kurzen Kommentar zur Bewertung in ein Feld einzugeben. Am Ende dieses Bereiches steht noch ein "Speichern"-Button zur Verfügung, um Bewertung und Kommentar zu speichern.

Im Bereich "Bewertung der anderen" kann man sich die Bewertungen der anderen Reisemitglieder anzeigen lassen. Wie Jameson [2004] in einigen Testversuchen zu einem Empfehlungssystem für Gruppenentscheidungen gezeigt hat, gibt es immer Mitglieder, die die Bewertungen der anderen nicht sehen wollen, da sie sich bei ihrer eigenen Bewertung nicht beeinflussen lassen möchten. Wie im System von Jameson [2004] werden daher auch hier standardmäßig die Bewertungen der anderen Mitglieder nicht direkt, sondern optional angezeigt. Erst mit einem Klick auf den "Bewertungen anzeigen"-Link werden die Bewertungen sowie die Kommentare der anderen sichtbar. Sind sie sichtbar, kann man sie natürlich auch wieder einklappen.

Abgesehen vom Hinzukommen der beiden Bewertungsbereiche ändert sich das Details-Fenster in der Bewertungsphase im Gegensatz zur Vorschlagsphase nicht, außer dass der Button, um die Sehenswürdigkeit zu löschen, verschwindet. In der Bewertungsphase dürfen keine Sehenswürdigkeiten mehr

### 3. Beschreibung der Applikation aus Benutzersicht

eingebraucht oder gelöscht werden, sondern nur noch vorhandene (also vorher eingebrachte) von allen Mitgliedern bewertet werden.

Ist die Bewertungsphase auch verstrichen und die Routen werden kalkuliert und angezeigt, ist der Kurzinformationsblock mit Google-Karte sichtbar sowie der Button, um die Sehenswürdigkeit auf der Pinwand zu posten, nicht aber der Button, um die Sehenswürdigkeit zu löschen. Weiters ist der Textblock um eigene Informationen hinzuzufügen sichtbar und editierbar. Danach ist der Block mit den Besichtigungsdetails sichtbar, jedoch können die Zeiten nicht mehr geändert werden, weswegen auch der dazugehörige "Speichern"-Button nicht mehr zur Verfügung steht. Da die Routen bereits kalkuliert werden oder wurden, hätte die Änderung der Zeiten keine Auswirkung mehr. Danach werden noch die beiden Bewertungsblöcke angezeigt. Während man die eigene Bewertung sowie den eigenen Kommentar nicht mehr editieren oder speichern kann, bleibt der Block mit den Bewertungen der anderen unverändert.

#### 3.2.5.2 Sehenswürdigkeiten vorschlagen lassen

Unterhalb der Reiseliste ist unter dem Menüpunkt "Sehenswürdigkeiten" ein Bereich platziert, in dem man sich individuelle Sehenswürdigkeiten vorschlagen lassen kann. Klickt man auf den dafür vorgesehenen Button rechts, werden Sehenswürdigkeiten, die nach den spezifischen Interessen, die der Benutzer unter dem Menüpunkt "Meine Interessen" angegeben hat, gesucht. Es öffnet sich ein Fenster, in dem die Ergebnisse präsentiert werden. Das Fenster hat zwei Seiten, auf jeder Seite werden fünf Sehenswürdigkeiten angezeigt. Es wird immer nur eine Seite angezeigt, am Ende jeder Seite kann man zwischen Seite 1 und Seite 2 hin und her wechseln. Ein Beispiel eines Fensters, in dem vorgeschlagene Sehenswürdigkeiten präsentiert werden, ist in Abbildung 3.9 gegeben.

Zu jeder Sehenswürdigkeit wird deren Name, eine kurze Adressangabe sowie die dazugehörige Kategorie angezeigt. Des Weiteren gibt es einen Link, um sich genauere Details zu dieser Sehenswürdigkeit anzeigen zu lassen. Klickt man auf diesen Link, öffnet sich ein weiteres Fenster, das bereits in Kapitel 3.2.5.1 beschrieben wurde. Rechts neben dem Kurzbeschreibungsbereich findet man ein Foto der Sehenswürdigkeit, das von Google zur Verfügung gestellt wird. Ist von Google kein Foto verfügbar, wird ein Symbolfoto mit dem Vermerk, dass kein Foto vorhanden ist, angezeigt. Als letztes sind zu jeder Sehenswürdigkeit zwei Buttons bereitgestellt: einer, um diese Sehenswürdigkeit zur Reiseliste hinzuzufügen und ein weiterer, um die Sehenswürdigkeit zu verwerfen, sodass sie nicht mehr angezeigt wird. Klickt man einen der Buttons an oder fügt man von der Details-Seite aus die Sehenswürdigkeit zur Reiseliste hinzu, wird die Sehenswürdigkeit von der Vorschlagsliste genommen, und eine weitere Sehenswürdigkeit rückt an der letzten Stelle nach.

Die vorgeschlagenen Sehenswürdigkeiten orientieren sich, wie schon erwähnt, an den Interessen der Benutzer. Hat ein Benutzer zum Beispiel ein großes Interesse an Museen angegeben, aber gar kein Interesse an Abendunterhaltung, so bekommt er viele Museen vorgeschlagen, aber keine Vorschläge, die in die Kategorie Abendunterhaltung fallen. Die Museen, die dem Benutzer in diesem Fall vorgeschlagen werden, sind Museen, die die höchsten Bewertungen auf Google bekommen haben. Haben die weltweiten Google-Benutzer also das Landeszeughaus als bestes Museum von Graz bewertet, so wird es dem Benutzer als erstes vorgeschlagen, wenn sein Reiseziel Graz oder in der Nähe von Graz ist.

Möchte der Benutzer zu einem bestimmten Zeitpunkt Sehenswürdigkeiten einer einzigen, bestimmten Kategorie sehen, kann er rechts oben des Vorschlags-Fensters die entsprechende Kategorie aus der Dropdown-Liste auswählen. Wählt ein Benutzer also die Kategorie "Kirchen und Klöster" aus, werden ihm zehn Vorschläge zu dieser Kategorie angezeigt. Hat der Benutzer genügend Sehenswürdigkeiten ausgewählt, kann er das Vorschlags-Fenster mit dem "x" in der rechten oberen Ecke wieder schließen und kommt auf den Inhalt des "Sehenswürdigkeiten"-Menüpunktes zurück.

Dieser Bereich, um sich Sehenswürdigkeiten vorschlagen zu lassen ist nur in der Vorschlagsphase (Phase 1) zu sehen. In den folgenden Phasen gibt es diesen Bereich nicht, da dann keine Sehenswürdigkeiten mehr aufgenommen oder gelöscht werden dürfen.

Vorschläge für Sehenswürdigkeiten
✕

Folgende Vorschläge werden angezeigt: Alle Kategorien lt. Interessen ▾

---

**S' Landhaus**  
Ritzinger Straße 4, Wolfsberg

[Details anzeigen](#)

Kategorie: Sport

Zur Reiseliste hinzufügen
Nicht mehr anzeigen



---

**Graz Opera**  
Kaiser-Josef-Platz 10, Graz

[Details anzeigen](#)

Kategorie: Abendunterhaltung

Zur Reiseliste hinzufügen
Nicht mehr anzeigen



---

**Zotter Schokoladen Manufaktur GmbH**  
Bergl 56, Riergersburg

[Details anzeigen](#)

Kategorie: Touristenattraktionen

Zur Reiseliste hinzufügen
Nicht mehr anzeigen



---

**Copacabana**  
Copacabana 1, Kalsdorf bei Graz

[Details anzeigen](#)

Kategorie: Spaß & Unterhaltung

Zur Reiseliste hinzufügen
Nicht mehr anzeigen



---

**Kunsthhaus Graz**  
Lendkai 1, Graz

[Details anzeigen](#)

Kategorie: Museen

Zur Reiseliste hinzufügen
Nicht mehr anzeigen



<< 1 2 >>

**Abbildung 3.9:** Hat man seine Interessen spezifiziert, kann man sich individuelle Sehenswürdigkeiten vorschlagen lassen.

### 3. Beschreibung der Applikation aus Benutzersicht

#### 3.2.5.3 Sehenswürdigkeiten hinzufügen

Der letzte Bereich des “Sehenswürdigkeiten”-Menüpunktes ist jener, wo man eigene Sehenswürdigkeiten einbringen kann. Damit ist gemeint, dass man Sehenswürdigkeiten, die einem einfallen, in das bereitgestellte Textfeld tippen kann und so auf diesem Wege auch Sehenswürdigkeiten zur Reiseliste hinzufügen kann, die nicht bei den Vorschlägen dabei sind. Voraussetzung für das Hinzufügen einer eigenen Sehenswürdigkeit ist aber, dass diese Sehenswürdigkeit Google bekannt ist, damit dann für die Routenberechnung die benötigten Koordinaten vorliegen. Dazu ist das Eingabefeld mit einer Google-Funktion versehen, die den gerade getippten Sehenswürdigkeitennamen automatisch in einem Dropdown-Menü vervollständigt. Ist die gewünschte Sehenswürdigkeit ausgewählt, klickt man auf den Button rechts daneben, dann wird die Sehenswürdigkeit in die Reiseliste aufgenommen. Dieser Bereich, um eigene Sehenswürdigkeiten einzubringen ist nur in der Vorschlagsphase (Phase 1) zu sehen. In den folgenden Phasen gibt es diesen Bereich nicht, da dann keine Sehenswürdigkeiten mehr aufgenommen oder gelöscht werden dürfen.

#### 3.2.6 Reiserouten

Unter dem Menüpunkt “Reiseroute” werden nach Ablauf der Bewertungsphase (Phase 2) drei empfohlene Routen angezeigt, von denen sich die Reisegruppe eine Route auswählt. Während der ersten beiden Phasen erhalten die Benutzer unter diesem Menüpunkt eine Information, dass sie, wenn die Sehenswürdigkeiten der Reiseliste bewertet wurden und die Bewertungsphase abgeschlossen ist, hier automatisch drei vom System generierte Routen empfohlen bekommen und diese Routen dann auf ihre Bewertungen der Sehenswürdigkeiten zugeschnitten sind. Ebenso ist zur Information noch das Datum angegeben, wann die Bewertungsphase endet.

Bevor die Routen berechnet werden, wird kontrolliert, ob alle dafür notwendigen Daten von der Gruppe angegeben wurden und diese richtig sind. Das betrifft die unter den Reisedaten angegebenen Ankunfts- und Abreiseorte, die Ankunfts- und Abreisetage sowie Start- und Endzeit der Besichtigungen. Fehlen Daten oder sind sie nicht korrekt, wird man darüber informiert und die Bewertungsphase wird um einen Tag verlängert, um die fehlenden oder falschen Daten korrigieren zu können. Die Bewertungsphase wird so lange um einen Tag hinausgezögert, bis alle Daten vollständig und korrekt sind.

Die drei Routen haben jeweils als Startort den unter den Reisedaten angegebenen Startort der Urlaubsrouten und als Endort den dort angegebenen Endort der Urlaubsrouten. Natürlich können diese beiden Orte auch dieselben sein. Der erste Tag für Besichtigungen ist der Ankunftstag am Startort und der Abreisetag ist der letzte Tag, an dem Besichtigungen stattfinden können. An jedem Urlaubstag werden nun zwischen der angegebenen Startzeit für Besichtigungen bis zur angegebenen Endzeit für Besichtigungen Sehenswürdigkeiten angezeigt. Die genauen Uhrzeiten, wann was besichtigt wird, werden nicht angezeigt, es ist lediglich eine Liste der Sehenswürdigkeiten für jeden Tag zu sehen, welche und in welcher Reihenfolge diese besichtigt werden. Auf eine genaue Uhrzeitangabe wurde bewusst verzichtet, da bisherige Erfahrungswerte zeigen, dass genaue Uhrzeiten auf einer Reise meist nicht eingehalten werden können. Wann welche Sehenswürdigkeiten vorgeschlagen werden, wird anhand von drei Kriterien berechnet:

- **Entfernung**

Die Route wird möglichst so gewählt, dass der Gesamtweg der Route, der über alle Tage zurückgelegt wird, möglichst kurz ist.

- **Öffnungszeiten und Besichtigungsdauer**

Sehenswürdigkeiten werden nur an den Tagen eingeplant, an denen die Benutzer bei den Details der Sehenswürdigkeit erlaubt haben, diese zu besuchen. Ebenso muss die Besichtigung in der ebenfalls bei den Details angegebenen Besichtigungszeit unter der Berücksichtigung der Anfahrtszeit an diesem Tag zeitlich noch möglich sein.

- **Bewertung**

Befinden sich mehr Sehenswürdigkeiten, als sich auf der Reise ausgehen würden, auf der Reiseliste, so werden Sehenswürdigkeiten, die eine bessere Bewertung von der Gruppe bekommen haben, den anderen vorgezogen.

Bei der Auswahl der Sehenswürdigkeiten wird also ein guter Kompromiss zwischen der benötigten Zeit (dazu zählt die Besichtigungsdauer ebenso wie die Wegzeiten) und der Bewertung der Sehenswürdigkeiten getroffen. Ist eine gut bewertete Sehenswürdigkeit etwas weiter weg von den anderen und vom kürzesten Weg, dann wird sie in die Route aufgenommen, wenn als Alternative zwei oder drei Sehenswürdigkeiten stünden, die eine schlechte Bewertung haben. Haben aber die zwei oder drei alternativen Sehenswürdigkeiten ähnlich gute Bewertungen, so werden sie der einen Sehenswürdigkeit, die weiter weg ist, vorgezogen. Als Grundprinzip gilt also: Qualität vor Quantität.

Wie schon erwähnt, werden vom System drei Routen berechnet. Diese können unterschiedliche Sehenswürdigkeiten beinhalten, da die Bewertungen der Sehenswürdigkeiten in den einzelnen Routen unterschiedlich gewichtet werden. Wie Gartrell u. a. [2010] erklären und in Kapitel 2.2 genauer erläutert und diskutiert wurde, gibt es verschiedene Gruppenentscheidungsstrategien, von denen die drei gebräuchlichsten die Strategie des durchschnittlichen Willens der Gruppe, die Strategie des geringsten Leides einer Einzelperson der Gruppe und die Strategie der größten Freude einer Einzelperson der Gruppe sind. Die drei Routen entsprechen jeweils einer der drei Gruppenentscheidungsstrategien. Eine beispielhafte Darstellung der drei Routen unter dem Menüpunkt "Reiseroute" wird in Abbildung 3.10 gegeben.

In der Abbildung sieht man, dass sich unter diesem Menüpunkt ein weiteres Menü mit drei Tab Sheets befindet. Jedes Tab Sheet beinhaltet eine der vorgeschlagenen Routen. Zu jeder Route findet man zuerst eine kurze Beschreibung, unter welchen Gesichtspunkten sie entstanden ist und welche Bedürfnisse der Gruppenmitglieder hier besonders berücksichtigt wurden. Darunter ist auf der linken Seite der Zeitplan der Reise angezeigt, während auf der rechten Seite daneben eine Google-Karte präsentiert wird. Der Zeitplan besteht aus einer Liste mit den Tagesplänen der Reise. Für jeden Tag stehen untereinander die Sehenswürdigkeiten aufgelistet, die an diesem Tag in dieser Reihenfolge besucht werden sollen. Hat die Gruppe in der Vorschlagsphase weniger Sehenswürdigkeiten auf die Reiseliste gestellt, als auf der Reise besucht werden könnten, werden die letzten Tage besichtigungsfrei bleiben. Im Zeitplan würde dann unter dem Datum anstatt der Aufzählung der Sehenswürdigkeiten "Heute keine Besichtigungen" stehen. Ebenso kann es zu einem freien Tag kommen, wenn die Gruppe an einem bestimmten Tag keine Sehenswürdigkeiten erlaubt hat oder die nächste Sehenswürdigkeit so weit weg ist, dass ein Tag ein reiner Reisetag ist. Jede Sehenswürdigkeit, die sich auf den Tagesplänen befindet, ist anklickbar. Mit einem Klick auf den Namen der Sehenswürdigkeit werden, wie auch in der Reiseliste unter dem Menüpunkt "Sehenswürdigkeiten", die Details zu dieser Sehenswürdigkeit in einem separaten Fenster der Applikation angezeigt. Vor jeder Sehenswürdigkeit wird in der Liste ein Marker mit einer Nummer angezeigt. Die Marker sind aufsteigend sortiert, sodass die erste Sehenswürdigkeit, die auf der Reise besichtigt wird, die Nummer 1 hat, und die letzte Sehenswürdigkeit der Reise die höchste Nummer hat. Diese Marker finden sich auch auf der rechten Seite in der Google-Karte wieder, damit man jede Sehenswürdigkeit auch örtlich schnell zuordnen kann. In der Karte gibt es zusätzlich zu den Markern für die einzelnen Sehenswürdigkeiten noch zwei Marker mit einem "S" und einem "E" für den Start- und den Endpunkt der Gesamtroute. In Abbildung 3.10 sind diese beiden Marker durch die Marker "2" und "6" verdeckt, man sieht die Start- und Endpunkte in diesem Fall nur nach Vergrößerung der Karte. Zwischen den einzelnen Markern ist der Weg, wie man von einer Sehenswürdigkeit zur nächsten kommt, mit einer blauen Linie markiert. Dadurch ist die Gesamtroute der Tour leicht zu erkennen. Wie bei allen anderen Karten, die in diese Applikation eingebunden sind, kann man auch hier die Karte beliebig verkleinern und vergrößern, sowie zwischen Satelliten- und Stadtplan-Ansicht wechseln und mit Hilfe des Männchens Fotos eines beliebigen Ortes anzeigen lassen. Google hat für die kostenfreie Nutzung einer solchen Streckenkarte die Einschränkung, dass nicht mehr als acht Zwischenziele angezeigt werden können [Google Developers, 2013a]. Sind also in der Route insgesamt mehr als acht Sehenswürdigkeiten enthalten, kann nicht die volle Strecke in der Karte dargestellt werden. In diesem Falle bekommen die Benutzer eine entsprechende Meldung angezeigt.

### 3. Beschreibung der Applikation aus Benutzersicht

Reisedaten   Meine Interessen   Sehenswürdigkeiten   **Reiseroute**   Notizen

## Empfohlene Reiserouten

Durchschn.-Wille-Routen   Geringstes-Leid-Routen   Größte-Freude-Routen

Bei dieser Route sind die durchschnittlichen Bewertungen der einzelnen Sehenswürdigkeiten berücksichtigt. Dabei wird jede Bewertung gleich gewichtet. Das heißt, der durchschnittliche Wille der Gruppe wird berücksichtigt.

**Freitag, 15. Mai 2015**

- 1 [Lama-Garten, Hardter Straße, Thal, Österreich](#)
- 2 [Landeszeughaus](#)

**Samstag, 16. Mai 2015**

- 3 [Schwarzlsee, Graz-Umgebung, Österreich](#)

**Sonntag, 17. Mai 2015**

- 4 [Mühlentof, Murek, Österreich](#)

**Montag, 18. Mai 2015**

- 5 [Hauptplatz, Leibnitz, Österreich](#)

**Dienstag, 19. Mai 2015**

- 6 [Ölmühle Hartlieb, Heimschuh, Österreich](#)



**Bewertung**

Meine Bewertung: ☆☆☆☆☆ Bitte bewerten

Kommentar:

**Bewertungen der anderen**

[Bewertungen anzeigen](#)

**Abbildung 3.10:** Unter dem Menüpunkt "Reiseroute" werden der Reisegruppe nach der Bewertungsphase (in Phase 3) drei Routen vorgeschlagen, wovon eine als endgültige Route gewählt werden kann.

### 3.2. Detaillierte Beschreibung der Einzelkomponenten

Unterhalb des Zeitplans und der Google-Karte befindet sich eine Sektion, um diese vorgeschlagene Route zu bewerten. Die Bewertung erfolgt gleich wie die Bewertung der Sehenswürdigkeiten. Es gibt also wieder fünf Sterne, wobei ein gelber Stern "Sehr schlechte Route" und fünf gelbe Sterne "Sehr gute Route" bedeuten. Neben den Sternen ist wieder die Bedeutung dynamisch zu finden. Ebenfalls gibt es wieder ein Kommentarfeld, wo man einen Kommentar zur Route hinterlassen kann sowie einen "Speichern"-Button, um Bewertung und Kommentar zu speichern. Darunter ist wie in der Bewertungsphase bei den Sehenswürdigkeiten wieder eine Sektion, wo man sich die Routenbewertungen der anderen Reisemitglieder anzeigen lassen kann. Standardmäßig ist diese wieder verdeckt, falls man bei der eigenen Bewertung nicht beeinflusst werden möchte. Mit einem Klick kann man die Bewertungen beliebig aus- und wieder einklappen. Dabei sieht man auch die Kommentare der anderen, falls sie welche abgegeben haben.

Zuletzt befindet sich am Ende jeder Route ein Button, um diese gerade aktiv angezeigte Route als fixe, endgültige Route zu wählen, wenn sich die Reisemitglieder auf eine Route geeinigt haben. Klickt man auf den Button, wird zur Sicherheit noch einmal nachgefragt, ob man diese Route wirklich als fixe, endgültige Route wählen möchte, damit man nicht aus Versehen eine Route wählt. Hat die Gruppe eine endgültige Route gewählt, wird die letzte Phase aktiv, in der unter dem Menüpunkt "Reiseroute" nur noch die ausgewählte Route angezeigt wird. Eine Darstellung dieses Menüpunktes in dieser letzten Phase ist in Abbildung 3.11 gegeben. Im Wesentlichen ändert sich hier bis auf das Verschwinden des Untermenüs und der nicht gewählten beiden Routen nicht mehr viel. Ganz oben bekommen die Benutzer die Information, welche der drei Routen sie gewählt haben. Darunter bleiben die Darstellung des Zeitplans und der Google-Karte gleich. Die eigene Bewertung ist nicht mehr abänderbar und auch der Kommentar wird angezeigt, ist aber nicht mehr änderbar. Des Weiteren ist damit der "Speichern"-Button überflüssig und wird nicht mehr angezeigt. Die Sektion mit der Routenbewertung der anderen Reisemitglieder bleibt gleich. Der Button, um eine Route zu wählen, ist natürlich auch nicht mehr sichtbar.

### 3. Beschreibung der Applikation aus Benutzersicht

Reisedaten   Meine Interessen   Sehenswürdigkeiten   **Reiseroute**   Notizen

## Eure gewählte Reiseroute

Ihr habt die Durchschn.-Wille-Route gewählt.

**Freitag, 15. Mai 2015**

- 1 [Lama-Garten, Hardter Straße, Thal, Österreich](#)
- 2 [Landeszeughaus](#)

**Samstag, 16. Mai 2015**

- 3 [Schwarzlsee, Graz-Umgebung, Österreich](#)

**Sonntag, 17. Mai 2015**

- 4 [Mühlenhof, Mureck, Österreich](#)

**Montag, 18. Mai 2015**

- 5 [Hauptplatz, Leibnitz, Österreich](#)

**Dienstag, 19. Mai 2015**

- 6 [Ölmühle Hartlieb, Heimschuh, Österreich](#)



**Bewertung**

Meine Bewertung: ★★★★★ Sehr gute Route

Kommentar: Alles dabei, was mir gefällt :)

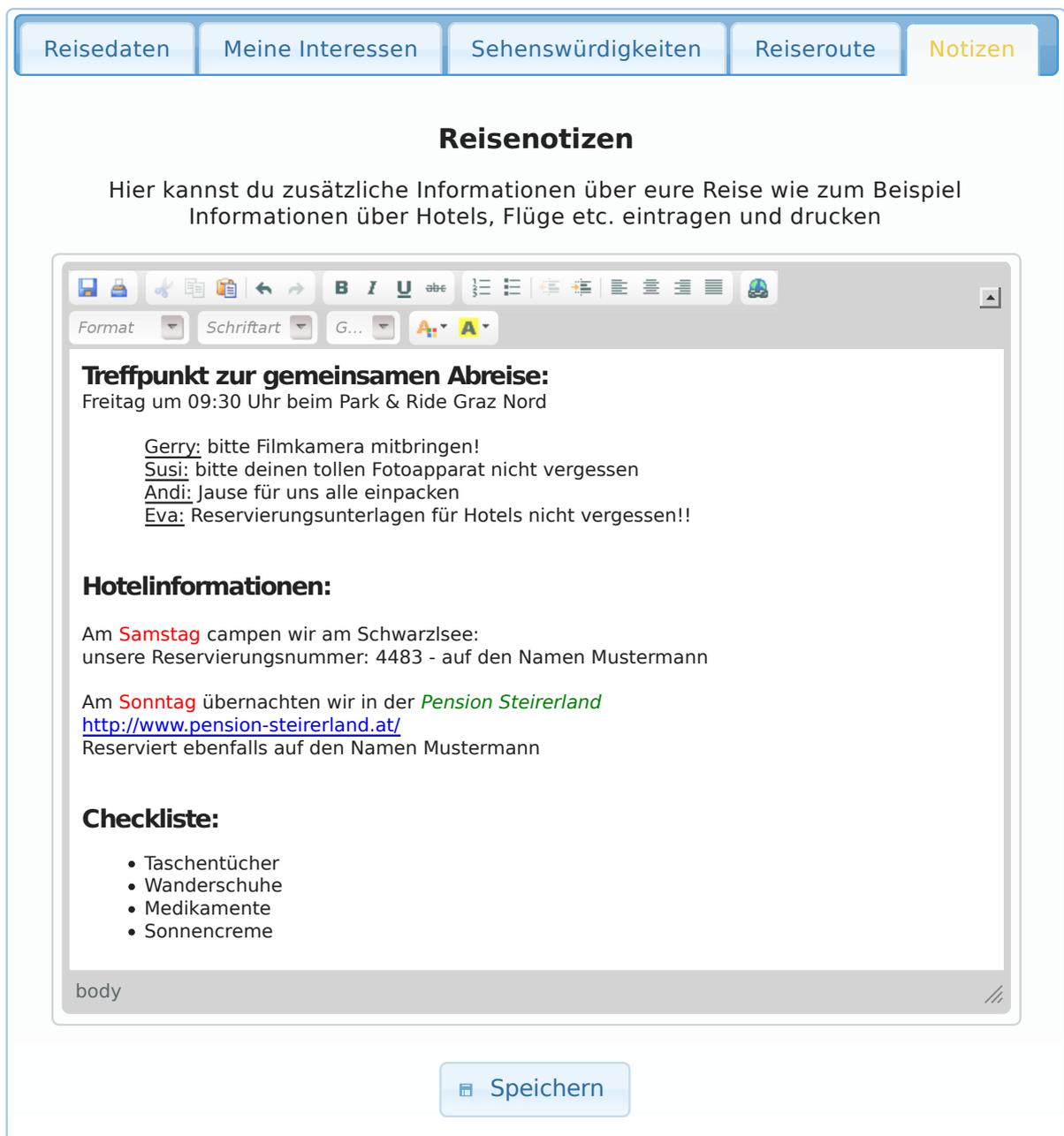
**Bewertungen der anderen**

[Bewertungen anzeigen](#)

**Abbildung 3.11:** Unter dem Menüpunkt "Reiseroute" wird der Reisegruppe nach der Auswahl einer der drei Routen selbige präsentiert.

### 3.2.7 Notizen

Unter dem letzten Menüpunkt “Notizen” kann die Reisegruppe eigene Zusatzinformationen zur Reise speichern und drucken. Gedacht ist, dass man hier Informationen über Flüge, Hotels, Mietautos, Treffpunkte (zum Beispiel wann man sich wo am Flughafen trifft) und andere organisatorische Daten, die die Reise betreffen, einträgt. Damit man diese Informationen auch optisch übersichtlich gestalten kann, steht ein eigener Editor, genannt “CKEditor” [Knabben, 2015] zur Verfügung. Mit Hilfe dieses Editors kann man zum Beispiel Schriftarten, die Schriftgröße, Formate, Textdarstellungen, Farben, Auflistungen, Texteinzüge und anderes abändern und einstellen. Auch gibt es die Funktionen kopieren, einfügen, ausschneiden, rückgängig machen sowie speichern und drucken der Unterlagen. Ein Beispiel, wie der Editor benutzt werden könnte, ist in Abbildung 3.12 zu sehen. Aus Gründen der Konsistenz wurde unterhalb des Editors ein zusätzlicher “Speichern”-Button angebracht.



**Abbildung 3.12:** Unter dem Menüpunkt “Notizen” kann die Reisegruppe Zusatzinformationen zur Reise verwalten, speichern und drucken.

### 3. Beschreibung der Applikation aus Benutzersicht

#### 3.2.8 Szenario

In diesem Unterkapitel wird die Applikation anhand eines vollständigen Beispiels, an dem man die zeitliche Abfolge der einzelnen Tätigkeiten besser verstehen kann, veranschaulicht. Nehmen wir an, es gibt eine Gruppe aus vier Personen, die gemeinsam einen Urlaub in Spanien machen wollen und sich dazu entschieden haben, für die nötige Organisation der Reise den "Holiday Recommender" zu verwenden. Nehmen wir weiter an, die vier Personen heißen Andreas, Sabine, Markus und Anita.

Markus startet von seiner Facebook-Seite aus den "Holiday Recommender" und klickt auf der Startseite auf den Link "Erzeuge eine neue Tour". Er benennt die Tour mit dem Namen "Spanien 2015" und landet automatisch auf der Reiseseite, auf der der Menüpunkt "Reisedaten" aktiv ist. Gleich beginnt er damit, die Reisedaten anzugeben. Als Ankunftsort wählt er "Malaga, Spanien", als Abreiseort "Sevilla, Spanien". Auf der Google-Karte rechts daneben sieht er die beiden Marker der gewählten Orte. Den Ankunfts- sowie Abreisetag lässt er vorerst aus, da er noch nicht weiß, wann die Flüge gehen werden und wann seine Kollegen Urlaub bekommen werden. Als Startzeit für Besichtigungen wählt er "10 Uhr", die Endzeit belässt er bei der voreingetragenen Uhrzeit "17 Uhr". Als Enddatum für die Vorschlagsphase wählt er den 1. März 2015. Als Ende der Bewertungsphase den 31. März 2015. Anschließend speichert er die angegebenen Daten. Rechts oben auf der Seite gibt er in das Feld, das dazu vorgesehen ist, Freunde zur Reise hinzuzufügen, die Namen seiner drei Freunde Sabine, Andreas und Anita ein und wählt sie dabei jeweils aus seiner Facebook-Freundesliste aus. Sabine, Andreas und Anita erhalten nun eine Einladung zur Reise.

Als Sabine das nächste Mal ihre Facebook-Seite startet, sieht sie, dass sie von Markus eine Einladung zur Reise "Spanien 2015" beim "Holiday Recommender" bekommen hat. Sie akzeptiert die Einladung und ist somit ein Mitglied der Reise. Ebenso nehmen auch Anita und Andreas die Einladung an und sind nun ebenfalls Reisemitglieder. Wenn die vier nun den "Holiday Recommender" starten, sehen sie auf der Startseite unter "Deine Touren" nun den Eintrag "Spanien 2015" und daneben den Namen von Markus, der die Reise angelegt hat.

Anita startet nun den "Holiday Recommender" und klickt auf die Reise "Spanien 2015", wodurch sie direkt zur Reiseseite weitergeleitet wird. Sie sieht nun die Reisedaten, die Markus bereits eingegeben hat. Sie klickt auf das Dropdown-Menü der Reisemitglieder und sieht, dass alle Mitreisenden bereits der Reisegruppe beigetreten sind. Anita und auch die anderen drei Reisemitglieder spezifizieren zunächst unter dem Menüpunkt "Meine Interessen" ihre persönlichen Interessen, indem sie den Schieber auf dem Balken der jeweiligen Kategorie hin- und herschieben. Danach klicken sie auf "Speichern", um ihre persönlichen Daten festzuhalten. Markus interessiert sich besonders für historische Stätten und Museen, Anita möchte auf der Reise viel Spaß und Unterhaltung haben, während Sabine Tiere und Natur liebt und daher ihr besonderes Interesse an Flora & Fauna spezifiziert. Andreas ist historisch sehr interessiert und gibt daher sein besonderes Interesse für Museen, historische Stätten und Kirchen & Klöster an.

Sabine klickt nun auf den Menüpunkt "Sehenswürdigkeiten" und sieht, dass sie sich hier Sehenswürdigkeiten vorschlagen lassen kann. Sie ist neugierig und klickt auf den Button, wodurch für sie interessante Orte gesucht werden. Es erscheint ein Fester, in dem ihr für sie passende Sehenswürdigkeiten präsentiert werden. Gleich fällt ihr der Dinosaurier-Park ins Auge und sie klickt auf "Details anzeigen", um genaueres zu diesem Park zu erfahren. Erneut öffnet sich ein Fenster mit weiteren Details zum Park. Sie sieht auf der Google-Karte sofort, dass der Park nicht weit von Malaga, ihrem Ankunfts-ort, entfernt ist. Auch sieht sie den Link zur offiziellen Webseite des Parks, auf den sie gleich klickt. Es öffnet sich ein neuer Tab ihres Browsers, in dem die Webseite nun angezeigt wird. Ihr gefällt der Dinosaurier-Park und so klickt sie im Details-Fenster des "Holiday Recommenders" auf "Zur Reiseliste hinzufügen". Sie erhält sofort die Information, dass die Sehenswürdigkeit in die Reiseliste aufgenommen wurde und gleichzeitig erweitert sich das Details-Fenster, indem ein Textfeld, um weitere Informationen zum Park anzugeben, erscheint, sowie eine Sektion, wo sie die Öffnungszeiten, bzw. die Tage, an denen sie den Park besuchen möchte, angeben kann und auch die gewünschte Aufenthaltsdauer bekannt geben kann. Da sie auf der Webseite gesehen hat, dass der Park täglich geöffnet hat, lässt sie die Vorauswahl, dass das System den Park an jedem Tag in der Route vorschlagen darf, bestehen und ändert die

### 3.2. Detaillierte Beschreibung der Einzelkomponenten

Aufenthaltsdauer auf “3 Stunden”. Nun schließt sie das Details-Fenster, wodurch das Fenster mit den vorgeschlagenen Sehenswürdigkeiten wieder zum Vorschein kommt. Der Dinosaurier-Park ist nun nicht mehr in den Vorschlägen enthalten, da dieser bereits ausgewählt wurde. Es ist durch die Auswahl aber eine neue Sehenswürdigkeit auf der zweiten Seite des Vorschlagsfensters hinzugekommen. Als “Flora & Fauna”-Attraktivität bekommt Sabine noch einen Hochseilgarten in den Bäumen vorgeschlagen sowie einen Besuch im Naturschutzgebiet Torcal de Antequera. Das Naturschutzgebiet möchte sie auf jeden Fall besuchen, daher klickt sie gleich direkt beim Vorschlag auf “Zur Reiseliste hinzufügen”, ohne sich die Details anzuschauen. Am Hochseilgarten ist sie nicht interessiert, daher klickt sie dort auf “Nicht mehr anzeigen”, damit sie diesen Vorschlag in der Zukunft nicht mehr erhält. Wieder rücken für diese beiden Sehenswürdigkeiten zwei weitere nach. Sie wählt noch ein paar weitere Sehenswürdigkeiten aus und schließt nun das Fenster mit den Vorschlägen. Sie ist nun wieder auf der Reiseseite, wo immer noch der Menüpunkt “Sehenswürdigkeiten” aktiv ist. Sie sieht jetzt direkt die aktuelle Reiseliste, in der ihre ausgewählten Sehenswürdigkeiten zu sehen sind. Daneben sieht sie die Google-Karte, in der alle Sehenswürdigkeiten mit Markern eingezeichnet sind. Sie ist vorerst zufrieden und beendet den “Holiday Recommender”.

Als Andreas das nächste Mal den “Holiday Recommender” nutzt und auf die Spanienreise klickt, sieht er unter dem Menüpunkt “Sehenswürdigkeiten” die aktuelle Reiseliste mit all den Sehenswürdigkeiten, die Sabine bereits eingebracht hat. Andreas möchte auch Ausflugsorte für die Reise einbringen und lässt sich ebenfalls Sehenswürdigkeiten vom System vorschlagen. Entsprechend seiner angegebenen Interessen bekommt er viele Museen, historische Stätten und religiöse Bauten vorgeschlagen. Wie Sabine wählt er ein paar davon aus. Dann möchte Andreas aber noch weitere Museen, die sich in der Gegend befinden, anzeigen lassen. Dazu wählt er in dem Fenster mit den Vorschlägen oben rechts aus dem Dropdown-Menü den Punkt “Museen” aus. Somit werden im Vorschlagsfenster ausschließlich nur noch Museen angezeigt, wovon er ein paar zur Reiseliste hinzufügt. Aus Interesse klickt Andreas auf den Menüpunkt “Reiseroute” und erhält dort die Information, dass, wenn die Sehenswürdigkeiten der Reiseliste bewertet wurden und die Bewertungsphase abgeschlossen ist, hier automatisch drei vom System generierte Routen empfohlen werden. Ebenfalls erhält er die Information, dass die Bewertungsphase am 31. März 2015 enden wird.

Markus sieht im “Holiday Recommender” ebenfalls die aktuelle Reiseliste mit den bisher eingebrachten Sehenswürdigkeiten von Sabine und Andreas. Er war schon einmal in Andalusien und bemerkt, dass noch niemand die Alhambra in Granada eingebracht hat. Die möchte er seinen Freunden gerne zeigen und tippt den Namen daher direkt unter dem Menüpunkt “Sehenswürdigkeiten” in der untersten Sektion “Sehenswürdigkeiten hinzufügen” in das Eingabefeld und wählt nun von den Google-Vorschlägen die Alhambra in Granada aus und fügt sie zur Reiseliste hinzu. Nun klickt er direkt in der Reiseliste die Alhambra an und kommt auf die Seite mit den Details zur Alhambra. Er trägt die gewünschte Aufenthaltsdauer ein und merkt in dem Textfeld für zusätzliche Informationen an, dass er schon einmal dort war und sie die anderen seiner Meinung nach unbedingt sehen sollten.

Anita bemerkt, dass ein Museum, das Andreas eingebracht hat, zur Urlaubszeit renoviert wird und daher geschlossen haben wird. Sie merkt dies im Informationstextfeld dieser Sehenswürdigkeit an, woraufhin Andreas diese Sehenswürdigkeit wieder von der Reiseliste löscht. Dies tut er, indem er im Details-Fenster direkt neben dem Informationstextfeld auf “Von der Reiseliste löschen” klickt. Ebenso hätte Andreas das Museum auch direkt aus der Reiseliste entfernen können, indem er auf das “X” neben dem entsprechenden Museum geklickt hätte. Anita hätte das Museum nicht löschen können, da dies nur der Einbringer, in diesem Fall Andreas, tun darf. Damit wird verhindert, dass zum Beispiel Markus nicht einfach ein Schuhgeschäft, in das Sabine gerne gehen möchte, von der Liste löscht, nur weil er an einem Schuheinkauf nicht interessiert ist. Er hat dann später in der Bewertungsphase die Möglichkeit, seinen Unwillen mit einer entsprechend negativen Bewertung zum Ausdruck zu bringen. Als Andreas und Anita einmal gleichzeitig im “Holiday Recommender” online sind und sich beide gleichzeitig Sehenswürdigkeiten vorschlagen lassen, wurde beiden eine bestimmte Touristenattraktion vorgeschlagen. Als Andreas dieses zur Reiseliste hinzufügen möchte, bekommt er die Meldung, dass sich diese Sehenswürdigkeit bereits auf der Reiseliste befindet, da Anita ein paar Sekunden davor ebenfalls auf “Zur Reiseliste hin-

### 3. Beschreibung der Applikation aus Benutzersicht

zufügen“ geklickt hat. Die Sehenswürdigkeit verschwindet im Vorschlagsfenster von Andreas und eine weitere Sehenswürdigkeit als Ersatz scheint auf.

Unter dem Menüpunkt “Notizen” schreiben alle Mitglieder vorerst einmal hinein, wann sie Urlaub bekommen sowie ihren Wunschtermin für die Reise. Bald einigt man sich auf ein Datum und Anita trägt unter dem Menüpunkt “Reisedaten” die bisher noch ausstehenden Tage ein: als Ankunftstag ist der 5. Juni 2015 geplant, als Abreisetag der 14. Juni 2015. Da bis dahin noch etwas Zeit ist, verlängert Anita noch die Endzeiten der Vorschlags- und der Bewertungsphase um jeweils ein Monat. Wie beschrieben bringen so alle Mitglieder weitere Sehenswürdigkeiten ein, bis die Vorschlagsphase am 1. April 2015 endet und gleichzeitig die Bewertungsphase der Sehenswürdigkeiten beginnt.

Als Sabine nun am 1. April den “Holiday Recommender” startet, sieht sie unter dem Menüpunkt “Meine Interessen” ihre angegebenen Interessen in ausgegrautem Farbton und kann diese nicht mehr ändern, da dies jetzt nicht mehr notwendig ist. Unter dem Menüpunkt “Sehenswürdigkeiten” sieht Sabine nur noch die Liste aller Sehenswürdigkeiten, die sie und ihre Freunde in der ersten Phase eingebracht haben inklusive der Google-Karte, in der alle Sehenswürdigkeiten verzeichnet sind. Die Sektionen, um sich Sehenswürdigkeiten vorschlagen zu lassen und direkt einzubringen, sind nicht mehr vorhanden, da nun keine weiteren Sehenswürdigkeiten mehr eingebracht werden können. Direkt in der Reiseliste sieht sie aber eine orange Überschrift, die besagt, dass sie die Sehenswürdigkeiten nun bewerten sollte, indem sie auf die Bewertungs-Buttons in der Liste klicke. In der Reiseliste sieht sie jetzt unter jeder Sehenswürdigkeit einen Button mit der Aufschrift “Bewertung” sowie jeweils fünf leere Sterne daneben. Die fünf Sterne deuten an, dass Sabine diese Sehenswürdigkeit noch nicht bewertet hat. Sie klickt nun auf den Bewertungs-Button des Dinosaurier-Parks und es öffnet sich das Fenster mit dessen Details, wobei in der Mitte die Bewertungssektion zu sehen ist. Sabine fährt mit der Maus über die Sterne, diese färben sich gelb ein und sie sieht daneben die Bedeutung der jeweiligen Anzahl an Sternen. Sie gibt dem Dinosaurier-Park volle fünf Sterne (sehr sehenswert), da sie die Sehenswürdigkeit selbst eingebracht hat und speichert ihre Bewertung. Anschließend schließt sie das Details-Fenster und sieht wieder die Reiseliste vor sich. Nun erscheinen neben dem Bewertungs-Button des Dinosaurier-Parks fünf gelbe Sterne, die ihre eben abgegebene Bewertung darstellen. Sabine bewertet auf diesem Wege noch ein paar weitere Sehenswürdigkeiten und schließt den “Holiday Recommender” wieder.

Auch Andreas findet bei seiner ersten Anmeldung nach Ablauf der Vorschlagsphase unter dem Menüpunkt “Sehenswürdigkeiten” die Reiseliste mit den Bewertungs-Buttons und der jeweils fünf leeren Sterne daneben. Auch er bewertet, gleich wie Sabine, ein paar Sehenswürdigkeiten und sieht in der Reiseliste seine eigenen Bewertungen anhand der kleinen gelben Sterne unter jeder Sehenswürdigkeit. Anita ist bei ihren Bewertungen unentschlossen und möchte gerne wissen, wie ihre Freunde über die einzelnen Sehenswürdigkeiten denken. Dazu klickt sie auf der Details-Seite der jeweiligen Sehenswürdigkeit in der Sektion “Bewertungen der anderen”, die direkt unter der Sektion der eigenen Bewertung ist, auf den Link “Bewertungen anzeigen”, worauf sie dort die Bewertungen ihrer Freunde angezeigt bekommt. Beim Dinosaurier-Park sieht sie, dass ihn Sabine mit fünf Sternen bewertet hat und Andreas mit drei Sternen. Markus scheint in der Liste noch nicht auf, woraus Anita richtig schließt, dass er diese Sehenswürdigkeit noch nicht bewertet hat. Markus möchte sich bei seinen Bewertungen nicht von den anderen beeinflussen lassen, weswegen er sich deren Bewertungen nicht anzeigen lässt. Weil er sich über das eingebrachte Schuhgeschäft von Sabine ärgert, bewertet er es mit nur einem Stern (nicht sehenswert).

Markus hat in der Zwischenzeit auch schon den Flug für alle gebucht und gibt die genauen Flugdaten unter dem Menüpunkt “Notizen” ein, damit alle Reisemitglieder jederzeit auf diese Informationen zugreifen können. Die Gruppe beschließt, die Besuchszeit mancher Sehenswürdigkeiten unter den Details zu verkürzen, damit sich auf der Route möglichst viele Sehenswürdigkeiten zeitlich ausgehen. Immer wieder melden sich die einzelnen Reisemitglieder während der Bewertungsphase an und bewerten ein paar weitere Sehenswürdigkeiten bzw. ändern gelegentlich auch eine ihrer Bewertungen. Am 30. April 2015 endet nun die Bewertungsphase automatisch und die Routen werden vom System berechnet.

Als sich Anita Anfang Mai einloggt, findet sie unter dem Menüpunkt “Reiseroute” drei vorgeschlagene Routen, die mit Hilfe eines eigenen Menüs angezeigt werden. Für jede Route sieht sie links den

### 3.2. Detaillierte Beschreibung der Einzelkomponenten

Terminplan mit den Sehenswürdigkeiten und rechts eine Google-Karte, in der alle Sehenswürdigkeiten inklusive der gesamten Wegstrecke eingezeichnet sind. Unterhalb jedes Routenvorschlages sieht sie eine Sektion, um die Route zu bewerten und einen Kommentar dazu abzugeben. Die Bewertung ist sowohl optisch als auch von der Bedienung her gleich wie die Bewertung der Sehenswürdigkeiten. Darunter ist wieder eine Sektion, in der man sich die Bewertung der anderen anzeigen lassen kann. Hier sieht Anita beim Ausklappen, dass die anderen Gruppenmitglieder noch keine Route bewertet haben. Als letztes auf jeder Seite findet Anita noch einen Button, um eine Route als fixe Route zu wählen. Anita vergleicht die drei vorgeschlagenen Routen, indem sie zwischen den drei Tabs hin- und herschaltet. Unter dem Menüpunkt "Sehenswürdigkeiten" sieht sie in der Reiseliste, wie sie die einzelnen Sehenswürdigkeiten bewertet hatte. Eine Änderung der Bewertungen ist nicht mehr möglich, weswegen auch die Buttons zur Bewertung nicht mehr angezeigt werden. Da sie hier ihre Bewertungen der Sehenswürdigkeiten übersichtlich sieht, fällt es ihr leichter, die einzelnen Routen zu beurteilen. Bei manchen Sehenswürdigkeiten fehlt ihr die genaue Erinnerung. Daher klickt sie in der Reiseliste oder auch direkt im Terminplan der Route selbst oder auf den Marker einer Karte auf die Sehenswürdigkeit, sodass sich das Details-Fenster dieser Sehenswürdigkeit öffnet. Wie immer sieht sie die kurze Beschreibung mit Foto und Google-Karte. Das Informationstextfeld wird ebenfalls wie gewohnt angezeigt und sie kann die Informationen noch editieren. Auch die angegebenen Besuchstage und -zeiten sowie ihre und die Bewertung der anderen werden ihr angezeigt, allerdings sind diese nicht mehr änderbar. Unter dem Menüpunkt "Reisedaten" sieht Anita, dass alle Informationen ausgegraut und nicht mehr änderbar sind.

Als sich Sabine ebenfalls anmeldet und die Routen vergleicht, fällt ihr auf, dass in der "Geringstes-Leid-Route" der Besuch ihres eingebrachten Schuhgeschäftes fehlt. Da Markus dieses mit "nicht sehenswert" bewertet hat, wurde bei dieser Route sein Unmut über das Geschäft besonders berücksichtigt. Im Gegensatz dazu sieht Sabine, dass bei der "Größte-Freude-Route" ihr Schuhgeschäft eingeplant ist, da hier ihre besondere Freude über das Geschäft extra berücksichtigt wurde. Bei der "Durchschnittlicher-Wille-Route" scheint das Schuhgeschäft nicht auf, da außer Markus auch Andreas und Anita das Geschäft nicht besonders attraktiv empfanden und bei dieser Route der durchschnittliche Wille der Gruppe berücksichtigt wurde. Sabine und auch Andreas bewerten die drei Routen.

Zuletzt bewertet Markus die drei Routen und bringt in die Reisenotizen unter dem Menüpunkt "Notizen" auch gleich ein paar Hotelvorschläge ein. Da er anhand der vorgeschlagenen Routen sehen kann, an welchem Tag sie in welchem Ort sind und an welchem Ort der Tag endet, kann er nun auch gezielt an diesen Orten Unterkünfte suchen. Alle Reisemitglieder haben nun ihre Bewertungen zu den einzelnen Routen abgegeben. Nun müssen die vier Freunde eine Route als ihre fixe, endgültige Reiseroute wählen. Da bei der Route, bei der der durchschnittliche Wille der Gruppe berechnet wurde, die größte Übereinstimmung herrscht, wählt die Gruppe diese Route als ihre fixe Reiseroute aus. Markus klickt dazu bei dieser Route auf den Button "Diese Route als fixe Reiseroute wählen", woraufhin sich ein Fenster öffnet, wo noch einmal nachgefragt wird, ob man diese Route wirklich als fixe, endgültige Route wählen möchte, damit man nicht aus Versehen eine Route wählt. Markus bestätigt die Auswahl und ab diesem Zeitpunkt sehen alle Mitglieder nur noch die gewählte Route unter dem Menüpunkt "Reiseroute". Damit auch Anita, Sabine und Andreas wissen, welche Route Markus gewählt hat, steht als erstes auf der Seite der Reiseroute, welche Route gewählt wurde. Darunter werden dann links der Terminplan und rechts die Google-Karte wie gewohnt angezeigt. In der darunter positionierten Bewertungs-Sektion sieht man seine eigene Routenbewertung sowie den Kommentar, diese sind aber nicht mehr änderbar. Die Sektion mit der Bewertung der anderen wird ebenfalls wie gewohnt dargestellt. Der Button, um die Route zu wählen, wird nun nicht mehr angezeigt.

Bevor die Reise losgeht, drucken sich die Reisemitglieder die Informationen, die sie unter dem Menüpunkt "Notizen" gesammelt haben, aus, um auf der Reise immer vorbereitet zu sein. Dazu klicken Anita, Sabine, Markus und Andreas einfach auf das Druckersymbol in der Menüleiste des Notizenfeldes.



## Kapitel 4

# Technische Beschreibung der Applikation

In diesem Kapitel wird die entwickelte Facebook-Applikation “Holiday Recommender” aus technischer Sicht beleuchtet. Dabei werden kurz die grundsätzliche Funktionsweise, sowie ausführlicher die wesentlichen Algorithmen, die den “Holiday Recommender” prägen, präsentiert.

### 4.1 Technische Grundlagen

Der “Holiday Recommender” ist als Applikation in das bestehende System Facebook integriert. Dazu stellt Facebook einen bestimmten Bildbereich zur Verfügung, in dem der “Holiday Recommender” eingebettet ist. Die Applikation selbst läuft auf einem eigenen Server, ihre Dateien sind aber mit Facebook verknüpft. Sie ist im Wesentlichen in PHP und JavaScript implementiert, wobei die Kommunikation zwischen Server und Klienten mittels AJAX-Technologie [Wikipedia, 2014] funktioniert. Somit läuft die Applikation vorwiegend auf dem Rechner des Klienten und der Server tritt nur in Aktion, wenn neue Daten gebraucht oder gespeichert werden, was den Server entlastet. Dies ermöglicht, dass eine größere Anzahl an Benutzern den “Holiday Recommender” gleichzeitig verwenden kann, ohne dabei an Geschwindigkeit einzubüßen. Die gesamte grafische Darstellung der Applikation ist aufgrund der besseren Übersicht und Wartbarkeit in eigenen CSS-Dateien gekapselt. Sämtliche Daten, die nach einer Sitzung noch erhalten bleiben müssen, werden in einer MySQL-Datenbank auf dem Server gespeichert. Spezielle Daten zu den Sehenswürdigkeiten sowie Entfernungen zueinander werden zum einen von Google mit Hilfe der Google API [2013], zum anderen von MapQuest mit Hilfe der MapQuest API [2014] eingeholt. Auch die grafischen Ortsdarstellungen der Sehenswürdigkeiten sowie der Reiserouten werden mit Hilfe von Google mittels Einbettung von Google-Karten realisiert.

### 4.2 Empfehlen von Sehenswürdigkeiten

Das Empfehlen der Sehenswürdigkeiten erfolgt mit einem Recommender System, das einerseits auf jene Daten, die die Benutzer eingegeben haben, zurückgreift und andererseits auch auf externe Google-Daten. Die Benutzer spezifizieren dazu ihre persönlichen Interessen an bestimmten Kategorien von Sehenswürdigkeiten. Werden keine eigenen Interessen spezifiziert, werden alle Kategorien als gleich interessant gewertet. Klickt man auf den Button, um sich Sehenswürdigkeiten vorschlagen zu lassen, werden zunächst die persönlichen Interessen sowie der Start- und Endort der Reise von der Datenbank abgefragt. Dann werden für alle neun Kategorien von Sehenswürdigkeiten Google-Anfragen gesendet. Sind der Startort und der Endort verschieden, werden pro Kategorie zwei “Nearby Search Requests” [Google, 2014], einen für die Umgebungssuche des Startortes und einen für die Umgebungssuche des Endortes,

## 4. Technische Beschreibung der Applikation

abgeschickt. Dabei sucht Google jeweils in einem Umkreis von 50 Kilometern nach den beliebtesten Sehenswürdigkeiten, die zur angegebenen Kategorie passen. Die Beliebtheit wird unter anderem an deren Google-Ranking und an den Bewertungen der Google-Benutzer gemessen.

Aus den erhaltenen Listen an Sehenswürdigkeiten jeder Kategorie werden zunächst diejenigen gelöscht, die sich bereits auf der Reiseliste befinden oder die der Benutzer nicht mehr angezeigt haben will. Da bei verschiedenen Start- und Endorten für jede Kategorie nun zwei Listen existieren, werden diese beiden Listen jeweils miteinander verschmolzen, wobei sichergestellt wird, dass Sehenswürdigkeiten, die zuvor in beiden Listen enthalten waren, in der neuen Liste nur noch ein Mal enthalten sind. Bei einigen Kategorien kann es vorkommen, dass Hotels als Ergebnisse von Google zurückgeliefert werden. So liefert die Unterkategorie "Schloss" gerne Hotels zurück (zum Beispiel ein Schlosshotel) oder die Unterkategorie "Schwimmbad" gerne Hotels, in denen sich ein Hallenbad befindet. Diese Hotels sollen dem Benutzer natürlich nicht als Sehenswürdigkeit präsentiert werden. Daher werden die Listen auch auf Hotels untersucht, die vor der Verschmelzung der Listen ebenfalls aus den Ergebnislisten entfernt werden.

Dadurch, dass die Ergebnisliste jeder Kategorie aussortiert wurde, kann es sein, dass für manche Kategorien nun zu wenig Sehenswürdigkeiten vorhanden sind. In diesem Falle werden weitere Sehenswürdigkeiten dieser Kategorie von Google angefordert, wieder aussortiert und zur Liste hinzugefügt. Dies geschieht so lange, bis ausreichend viele Sehenswürdigkeiten vorhanden sind oder es keine weiteren mehr gibt. Am Ende wird jede Liste sortiert, sodass die beliebteste Sehenswürdigkeit dieser Kategorie als erstes steht und danach die jeweils nächst beliebtesten. Ein stark vereinfachter Pseudo-Code des Algorithmus' ohne Beachtung von Sonderfällen ist in Quellcode 4.1 gegeben.

```
1 Startort = erhalteStartOrt();
2 Endort = erhalteEndOrt();
3 Interessen = erhalteInteressen();
4 für jede Kategorie:
5 {
6     Startort_Liste = erhalteGoogleSehenswürdigkeiten();
7     Endort_Liste = erhalteGoogleSehenswürdigkeiten();
8     Startort_Liste = LöscheSehenswDieSchonAufReiselisteSind();
9     Endort_Liste = LöscheSehenswDieSchonAufReiselisteSind();
10    Startort_Liste = LöscheSehenswDieNichtMehrAngezeigtWerdenSollen();
11    Endort_Liste = LöscheSehenswDieNichtMehrAngezeigtWerdenSollen();
12    Startort_Liste = LöscheHotels();
13    Endort_Liste = LöscheHotels();
14    Kategorie_Liste = VerschmelzeStarortUndEndortListe();
15    Solange zu wenig Sehenswürdigkeiten in der Liste:
16    {
17        neue_Startort_Liste = erhalteWeitereGoogleSehenswürdigkeiten();
18        neue_Endort_Liste = erhalteWeitereGoogleSehenswürdigkeiten();
19        neue_Startort_Liste = LöscheWiederAlleIrrelevantenSehensw();
20        Kategorie_Liste = VerschmelzeAlteUndNeueListen();
21    }
22    Kategorie_Liste = SortiereNachBeliebtheit();
23    Anzahl_benötigter_Sehenswürdigkeiten = BerechneAnzahl();
24 }
25 Vorschlagsfenster = PräsentiereSehenswürdigkeiten();
```

**Src. 4.1:** Stark vereinfachter Pseudo-Code des Algorithmus' um Sehenswürdigkeiten von Google zu erhalten und zu präsentieren ohne Beachtung von Sonderfällen, wenn der Startort nicht derselbe ist wie der Zielort.

Nun muss aus den erhaltenen Sehenswürdigkeiten eine Empfehlung für den Benutzer erstellt werden, die seine Interessen widerspiegelt. Die Anzahl  $n_i$  (siehe Gl. 4.1 und 4.3) an Sehenswürdigkeiten aus jeder der  $K = 9$  Kategorien ist daher abhängig vom Interesse  $I_i$  des Benutzers an Sehenswürdigkeiten aus

dieser Kategorie  $i$ . ( $I_i$  und  $I_j$  stehen hierbei für den Wert in Prozent, den der Benutzer bei den Interessen für die jeweilige Kategorie  $i$  bzw.  $j$  angegeben hat:  $I_i, I_j \in [0,100]$ ). Berechnet wird eine Aufteilung von insgesamt etwa  $N = 20$  Sehenswürdigkeiten. Sind diese vorbereiteten ca.  $N$  Stück durch Auswählen oder nicht mehr anzeigen lassen verbraucht, werden die nächsten ca.  $N$  anhand der erhaltenen Aufteilung vorbereitet:

$$N = \sum_{i=1}^K n_i \quad (4.1)$$

Mit Hilfe von Gleichung 4.2 wird allgemein berechnet, wie viele Sehenswürdigkeiten aus jeder Kategorie ausgewählt werden müssen:

$$N = 20 \cdot \frac{1}{K} \cdot \sum_{i=1}^K \overbrace{\left( I_i \cdot \frac{1}{\frac{1}{K} \cdot \sum_{j=1}^K I_j} \right)}^{=1} = \sum_{i=1}^K 20 \cdot \frac{\frac{I_i}{K}}{\sum_{j=1}^K \frac{I_j}{K}} \quad (4.2)$$

$\underbrace{\qquad\qquad\qquad}_{=A}$   
arithm. Mittel der  $I_i$

Genau genommen wird hier das arithmetische Mittel über die, mittels arithmetischen Mittels der Interessen des Benutzers ( $A := \frac{1}{K} \cdot \sum_{j=1}^K I_j$ ), gemittelten Interessen des Benutzers gebildet ( $\frac{1}{K} \cdot \sum_{i=1}^K \frac{I_i}{A}$ ) ( $= 1$ ), das dann auf den Wert 20 normiert wird. Aus Gleichung 4.2 ergibt sich nun Gleichung 4.3, mit der man für jede einzelne Kategorie eine bestimmte Anzahl an Sehenswürdigkeiten  $n_i$  erhält:

$$n_i = N \cdot \frac{\frac{I_i}{K}}{\sum_{j=1}^K \frac{I_j}{K}} \quad (4.3)$$

Da man nun für  $n_i$  reelle, nicht-ganze Zahlen erhalten könnte, man aber als Anzahl an Sehenswürdigkeiten natürliche Zahlen braucht, wird  $n_i$  am Ende mathematisch auf die nächstliegende natürliche Zahl gerundet. Dadurch kann es sein, dass eventuell auch 19 oder 21 Sehenswürdigkeiten statt exakt 20 vorbereitet werden, was den logischen Verlauf des Algorithmus' aber nicht stört.

Da nun die Anzahl an Sehenswürdigkeiten pro Kategorie bekannt ist und auch die Listen mit den Sehenswürdigkeiten vorhanden sind, kann damit begonnen werden, dem Benutzer die entsprechenden Sehenswürdigkeiten zu präsentieren. Dazu werden zehn Stück der jeweils beliebtesten Sehenswürdigkeiten aus allen Kategorien, aus denen welche präsentiert werden sollen, angezeigt. Es wurden aber ca. 20 Stück vorbereitet, da 20 Sehenswürdigkeiten die Interessen des Benutzers besser repräsentieren als nur zehn. Bei nur zehn Stück würde sich durch die starke Reduktion (die durch das Runden auf Ganzzahlen entsteht) eine zu große Ungenauigkeit ergeben. Bei einem kleinen Wert von  $N$  würden selbst große Interessensabstände auf ein und dieselbe Ganzzahl abgebildet werden. Es ist aber nicht ratsam, dem Benutzer mehr als zehn Sehenswürdigkeiten gleichzeitig anzuzeigen, da man sonst leicht die Übersicht verlieren würde. Es ist aber davon auszugehen, dass der Benutzer gleich ein paar Sehenswürdigkeiten auswählt oder nicht mehr angezeigt bekommen möchte. In diesem Fall wird sofort eine Sehenswürdigkeit aus der nächsten Kategorie, von der noch Sehenswürdigkeiten gebraucht werden, ausgewählt und präsentiert. Somit bekommt man, wenn man zehn Sehenswürdigkeiten auswählt oder wegklickt, insgesamt eine gute Auswahl, die den angegebenen Interessen entspricht.

Vermutlich will der Benutzer aber auch nicht mehr als 20 Sehenswürdigkeiten ohne Unterbrechung bearbeiten, weswegen die Interessen des Benutzers auf 20 Stück abgebildet wurden. Möchte der Benutzer mehr Sehenswürdigkeiten bearbeiten, ist dies aber auch kein Problem, denn wenn alle ca. 20 Sehenswürdigkeiten verbraucht sind, wird einfach eine Auswahl der nächsten 20 Sehenswürdigkeiten anhand der berechneten Aufteilung getroffen. Die Berechnung stellt daher einen guten Kompromiss zwischen einer noch übersichtlich darstellbaren Anzahl an Sehenswürdigkeiten und einer noch guten Repräsentation der Interessen des Benutzers dar.

Die ersten zehn Sehenswürdigkeiten, die dem Benutzer vorgeschlagen werden, sind aus allen Kategorien, aus denen welche gebraucht werden, zusammengestellt. Hat der Benutzer an einer Kategorie

## 4. Technische Beschreibung der Applikation

überhaupt kein Interesse, werden von dieser keine Sehenswürdigkeiten benötigt. So sieht der Benutzer gleich viele verschiedene Sehenswürdigkeiten, was sehr interessant erscheint. Das ist besser, als dem Benutzer gleich aus den ersten Kategorien alle benötigten Sehenswürdigkeiten zu zeigen, solange, bis zehn Sehenswürdigkeiten erreicht sind, denn in diesem Falle würden aus manchen Kategorien (den hinteren), die aber auch als interessant vermerkt wurden, gar keine Sehenswürdigkeiten dabei sein, was den Benutzer verwundern dürfte. Bearbeitet der Benutzer dann ein paar Sehenswürdigkeiten, werden weitere nachgeladen, wobei dann automatisch welche aus den für den Benutzer besonders interessanten Kategorien nachgeladen werden, da aus diesen ja noch nicht alle der insgesamt 20 vorbereiteten Sehenswürdigkeiten präsentiert wurden. Es wird sowohl bei der Auswahl der ersten zehn Sehenswürdigkeiten als auch beim Nachladen der Sehenswürdigkeiten immer nacheinander eine aus der jeweils nächsten Kategorie gewählt, aus der noch eine gebraucht wird. Dazu werden alle neun Kategorien immer systematisch nacheinander abgefragt, ob aus ihnen noch Sehenswürdigkeiten gebraucht wird, bis die 20 Sehenswürdigkeiten abgearbeitet wurden. Danach gilt selbiges für die nächsten 20 Sehenswürdigkeiten.

Für jede einzelne Liste jeder Kategorie wurden, wie bereits beschrieben, schon im Vorfeld doppelte Einträge aussortiert. Somit ist sichergestellt, dass zum Beispiel das Schloss Schönbrunn in der Liste der Kategorie "Historische Stätten" nicht mehrfach vorkommt. Trotzdem kann es aber passieren, dass das Schloss Schönbrunn zum Beispiel auch in der Liste der Kategorie "Touristenattraktionen" vorkommt. Daher muss, bevor die Sehenswürdigkeit dem Benutzer angezeigt wird, überprüft werden, ob diese Sehenswürdigkeit nicht schon vorgeschlagen wurde.

Gehen während der Erstellung der ersten zehn Sehenswürdigkeiten oder bei der Bearbeitung der vorgeschlagenen Sehenswürdigkeiten selbige einer Kategorie aus, obwohl noch Sehenswürdigkeiten von ihr gebraucht werden würden, wird automatisch eine neue Google-Anfrage für diese eine Kategorie gestartet. Hat Google keine weiteren Sehenswürdigkeiten dieser Kategorie mehr im angegebenen Suchraum, wird diese Kategorie ignoriert. Ein stark vereinfachter Pseudo-Code des Algorithmus' um aus den von Google erhaltenen Sehenswürdigkeiten die ersten zehn Stück auszuwählen und diese dem Benutzer zu präsentieren ist in Quellcode 4.2 gegeben.

```
1 Anzahl_noch_anzuzeigender_Sehenswürdigkeiten = 10;
2 Aktuelle_Kategorie = Museum; // Starte mit Kategorie Museum
3 solange noch Sehenswürdigkeiten gebraucht werden:
4 {
5     wenn es keine weiteren Sehenswürdigkeiten mehr gibt:
6     {
7         teileEsDemBenutzerMit();
8     }
9     wenn aus aktueller Kategorie Sehenswürdigkeiten gebraucht werden:
10    {
11        Sehenswürdigkeit = NimmNächsteSehenswAusDerKategorieListe();
12        wenn Sehenswürdigkeit erhalten:
13        {
14            ÜberprüfeObSehenswSchonAufAnzeigeIst();
15            wenn ja, mach solange bis eine Sehensw. noch nicht vorhanden ist:
16            {
17                Sehenswürdigkeit = NimmNächsteSehenswAusDerKategorieListe();
18            }
19            ZeigeSehenswürdigkeitAn();
20            Anzahl_noch_anzuzeigender_Sehenswürdigkeiten --;
21            Aktuelle_Kategorie = nächste_Kategorie;
22        }
23    ansonsten:
24    {
25        wenn nicht schon eine ergebnislose Anfrage gemacht wurde:
26        {
27            StarteNeuenGoogleRequestFürDieseKategorie();
28        }
```

```

29     ansonsten:
30     {
31         //ignoriere diese Kategorie
32         Aktuelle_Kategorie = nächste_Kategorie;
33     }
34 }
35 }
36 ansonsten:
37 {
38     Aktuelle_Kategorie = nächste_Kategorie;
39 }
40 }

```

**Src. 4.2:** Stark vereinfachter Pseudo-Code des Algorithmus' um aus den von Google erhaltenen Sehenswürdigkeiten die ersten zehn Stück auszuwählen und diese dem Benutzer zu präsentieren ohne Beachtung von Fehlerfällen.

### 4.3 Empfehlen von Routen

Nachdem eine Reisegruppe Sehenswürdigkeiten eingebracht und bewertet hat, werden nach Ablauf der Bewertungsphase vom System automatisch drei Routen berechnet. Das Empfehlen dieser drei Routen erfolgt mit Hilfe eines Recommender Systems, das auf die Bewertungen der Sehenswürdigkeiten durch die Reisemitglieder, die erlaubten Tage (Öffnungszeiten), die gewünschte Besichtigungsdauer und die Reisedaten sowie externe Google- und MapQuest-Daten zurückgreift.

Da die Problematik, eine optimale Route zu finden, dem Problem des Handelsreisenden entspricht [Cook, 2011] und dieses NP-hart ist [Prokop, 2014], [Cook, 2011], also vermutlich nicht deterministisch in polynomieller Zeit lösbar ist, die Routen aber in akzeptabler Zeit berechnet werden müssen, kommt für die Berechnung des vorliegenden Optimierungsproblems eine Näherungslösung zum Einsatz. Der für den "Holiday Recommender" entwickelte Algorithmus, der sowohl die Auswahl der Sehenswürdigkeiten trifft als auch die Routen berechnet, ist stark an die Arbeit von Vansteenwegen u. a. [2009] angelehnt. Es handelt sich hierbei um eine Annäherung, die iterativ vorgeht und dabei in jedem Iterationsschritt einen Einfüge- und einen Mischprozess durchführt. Im Einfügeprozess werden die Sehenswürdigkeiten durch lokale Suche in die Route eingefügt und mit Hilfe des anschließenden Mischprozesses wird versucht, einem lokalen Optimum zu entkommen. Der Algorithmus berücksichtigt dabei, dass jede Sehenswürdigkeit nur an den Tagen, an denen die Reisegruppe eine Besichtigung erlaubt hat, eingefügt werden darf, und dass die angegebene Besichtigungszeit eingehalten wird. Darüber hinaus werden die täglichen Start- und Endzeiten für Besichtigungen, die die Gruppe unter dem Menüpunkt "Reisedaten" angegeben hat, sowie die Fahrtzeiten zwischen den Sehenswürdigkeiten berücksichtigt. Unter all diesen Bedingungen wird dann eine kürzestmögliche Strecke berechnet.

Google, Open Street Map und andere im Internet verfügbare Kartenseiten, die Navigationen berechnen können, liefern nur in sehr eingeschränkter Weise Annäherungen an optimale Routen. So gibt es keine Online-Plattform, die eine Navigation unter Berücksichtigung von Öffnungszeiten oder Zeitfenstern zulässt. Ebenso muss man jedem solchen System für jede Routenanfrage eine fixe Anzahl an Ortspunkten zur Routenoptimierung übergeben, wobei die maximal erlaubte Anzahl überall sehr gering ist. Das heißt, dass diese Systeme weder eine optimale Auswahl an Sehenswürdigkeiten treffen können, noch in einem vorgegebenen Zeitfenster arbeiten können. Man könnte also nur selbst iterativ eine Auswahl an Sehenswürdigkeiten vornehmen und dann daraus Routenanfragen erstellen und die erhaltenen Routen danach vergleichen, was aber ebenfalls problematisch wird, da diese Online-Plattformen nur eine beschränkte Anzahl an Anfragen pro Tag zulassen. Auf eine Einbindung von Routenberechnungen durch Online-Plattformen wurde daher verzichtet und stattdessen ein eigener Algorithmus entwickelt.

## 4. Technische Beschreibung der Applikation

In der Näherungslösung von Vansteenwegen u. a. [2009] wird zwar die Einhaltung der Öffnungszeiten der Sehenswürdigkeiten berücksichtigt, jedoch müssen bei diesem Algorithmus für jeden Tag der Start- und der Endpunkt der Tagesteilstrecke bekannt sein. Jeder Tag wird als unabhängige Teilroute mit fixem Anfangs- und Endpunkt gesehen, was gut funktioniert, wenn man Urlaub an einem einzigen Ort macht. In diesem Fall nächtigt man immer im selben Hotel und kann täglich Sehenswürdigkeiten in der Nähe besichtigen, wobei man immer im Hotel startet und abends ins selbe Hotel zurückkehrt. Im Falle des “Holiday Recommenders” kann die Reise aber auch über eine bestimmte Strecke gehen. So kann man zum Beispiel eine Reise in Dublin (Irland) starten und in Shannon (Irland) wieder abreisen. Im Fall einer solchen Streckenreise kennt man also die täglichen Start- und Endorte nicht, man kennt lediglich den Startort am ersten Tag und den Endort am letzten Urlaubstag.

Es kann bei einer Streckenreise auch notwendig sein, dass ein gesamter Tag als reiner Reisetag ohne Besichtigungen dient. Dies ist im Algorithmus von Vansteenwegen u. a. [2009] nicht möglich, da die Tage unabhängig, als einzelne Routen, behandelt werden, sodass eine Anfahrt immer nur gemeinsam mit der Besichtigung einer Sehenswürdigkeit an einem Tag erfolgen kann.

Im neuen Algorithmus, der den Ansprüchen des “Holiday Recommenders” gerecht wird, wurde das Grundprinzip der lokalen, iterativen Suche mittels Einfüge- und Mischprozess von Vansteenwegen u. a. [2009] übernommen, da dieser sehr effizient eine gute Näherungslösung liefert und grundsätzlich mit Zeitfenstern umgehen kann. Er wurde aber insofern erweitert, als nun auch reine Reisetage ohne Besichtigungen möglich sind und die Start- und Endorte der Tagesteilstrecken nicht mehr bekannt sein müssen. Das wurde dadurch erreicht, dass die einzelnen Tage nicht mehr als unabhängige Teilrouten gesehen werden, sondern nur noch eine Gesamtroute modelliert wird, deren Start- und Endort ja bekannt sind. Die Nachtruhezeiten, also die Zeiten, in denen die Reisegruppe keine Besichtigungen wünscht, wird dabei als eigenes Zeitfenster behandelt. Die modellierte Gesamtroute besteht aus den einzelnen Reisetagen, an denen jede Sehenswürdigkeit zu den zugelassenen Zeiten völlig flexibel eingefügt werden kann, wodurch eine Optimierung der Tagesteilstrecken und dadurch eine Optimierung der Gesamtroute erreicht wird.

Müsste man Start- und Endpunkt jeder Tagesteilstrecke vor der Routenberechnung angeben, würde dies in den meisten Fällen eine weniger optimale Gesamtroute ergeben, da die Optimierung der Tagesteilstrecken durch diese Angaben in ihrer Flexibilität stark eingeschränkt wäre. Da sich durch die Einschränkung der Tagesteilstrecken eine andere, nicht so optimale, Gesamtroute ergeben könnte, als sich bei einer optimierten, unbeschränkten Wahl der Tagesteilstrecken ergeben würde, würden möglicherweise interessante Sehenswürdigkeiten durch diese Einschränkung wegfallen und dafür stattdessen eventuell weniger gute in die Route aufgenommen werden. Aufgrund der optimierten, flexiblen, automatischen Wahl von Start- und Endpunkt jeder Tagesteilstrecke durch den Algorithmus des “Holiday Recommenders” wird dies verhindert, was eine deutliche Verbesserung gegenüber dem Algorithmus von Vansteenwegen u. a. [2009] darstellt. Des Weiteren wird dadurch automatisch eine bewusste Manipulation der Route verhindert. Möchte zum Beispiel nur ein Mitglied eine weit abseits gelegene Sehenswürdigkeit besuchen und die ändern nicht, könnte dieses Mitglied das System von Vansteenwegen u. a. [2009] zu seinem Vorteil ausnutzen, indem es als Endpunkt eines Tages den Ort bei dieser Sehenswürdigkeit auswählt. Denn dadurch könnte diese Sehenswürdigkeit gegenüber einer optimierten Gesamtlösung doch noch in die Route aufgenommen werden. Man müsste sich in diesem Fall also eine faire Strategie überlegen, wie Start- und Endpunkt jeder Tagesteilstrecke in der Gruppe fair gewählt werden können, was beim Routenalgorithmus des “Holiday Recommenders” nicht notwendig ist.

### 4.3.1 Entscheidungsstrategien

Für den “Holiday Recommender” werden insgesamt drei Routen berechnet und der Reisegruppe vorge schlagen. Eine Route, die den durchschnittlichen Willen der Gruppe berücksichtigt, eine, die der Strategie des geringsten Leides einer Einzelperson der Gruppe entspricht und eine, die die Strategie der größten Freude einer Einzelperson der Gruppe verkörpert. Für alle drei Strategien wird der selbe Routenalgorithmus angewendet. Lediglich die Bewertung der Sehenswürdigkeiten durch die Gruppenmitglieder, die dem Algorithmus übergeben wird, wird anders gewichtet.

Die Bewertungsfunktion für die Strategie des durchschnittlichen Willens gewichtet jede Bewertung der einzelnen Gruppenmitglieder gleich. Das heißt, dass der Durchschnitt über die einzelnen Bewertungen gebildet wird. Wenn  $n$  die Anzahl der Gruppenmitglieder ist und  $r_{ij}$  die Bewertung von Benutzer  $j$  für die Sehenswürdigkeit  $i$  ist, dann ergibt sich für die Gesamtwertung der Sehenswürdigkeit  $i$  folgende Formel [Masthoff, 2010], [Gartrell u. a., 2010]:

$$G_i = \frac{\sum_{j=1}^n r_{ij}}{n} \quad (4.4)$$

Bei der Bewertungsfunktion für die Strategie des geringsten Leides einer Einzelperson der Gruppe wird die schlechteste (niedrigste) Bewertung einer Sehenswürdigkeit als Gesamtwertung herangezogen [Masthoff, 2010], [Gartrell u. a., 2010]:

$$G_i = \min(r_{ij}) \quad (4.5)$$

Bei der Bewertungsfunktion für die Strategie der größten Freude einer Einzelperson der Gruppe wird die beste (höchste) Bewertung einer Sehenswürdigkeit als Gesamtwertung herangezogen [Masthoff, 2010], [Gartrell u. a., 2010]:

$$G_i = \max(r_{ij}) \quad (4.6)$$

### 4.3.2 Inputdaten und Vorbedingungen

Damit der Routenalgorithmus arbeiten kann, müssen ihm folgende Daten als Input zur Verfügung gestellt werden:

- **Gesamtwertungen der Sehenswürdigkeiten**  
Das sind insgesamt drei Listen, die die jeweiligen Gesamtwertungen aller Sehenswürdigkeiten, nach der entsprechenden Strategie berechnet, enthalten.
- **Start- und Enddatum der Tour**  
Das sind Ankunftstag am Startort und Abreisetag vom Endort, die die Gruppe unter dem Menüpunkt "Reisedaten" angegeben hat.
- **Besichtsungszeitraum**  
Dieser beginnt mit der täglichen Startzeit für Besichtigungen und endet mit der täglichen Endzeit für Besichtigungen, die die Gruppe unter dem Menüpunkt "Reisedaten" angegeben hat.
- **Erlaubte Besuchstage und Besuchszeiten**  
Das ist eine Liste, in der für jede Sehenswürdigkeit notiert ist, an welchen Tagen sie besucht werden darf und wie lange sie besucht werden soll, so wie es die Gruppe in den Details zu jeder Sehenswürdigkeit angegeben hat.
- **Koordinaten**  
Das ist eine Liste, in der alle Koordinaten der Sehenswürdigkeiten wie auch des Start- und des Endpunktes der Reise enthalten sind. Alle Koordinaten werden von Google mithilfe der Google API eingeholt.
- **Distanzmatrix**  
Die Distanzmatrix ist eine Tabelle, die die Fahrtzeiten zwischen allen Sehenswürdigkeiten enthält. Diese Fahrtzeiten beziehen sich auf das Straßennetz mit einem Auto als Fortbewegungsmittel. Die Distanzmatrix wird von MapQuest mittels MapQuest API eingeholt.

## 4. Technische Beschreibung der Applikation

Hat die Gruppe bei einer Sehenswürdigkeit angegeben, dass sie den ganzen Tag besichtigt werden soll, muss diese Angabe vor dem Start des Algorithmus' noch in eine passende Zeitangabe umgewandelt werden. Hat die Reisegruppe bei den Reisedaten zum Beispiel eine Besichtigungszeit von 9 Uhr bis 17 Uhr angegeben, wäre es einfach, anstatt der Besichtigungsdauer "ganzer Tag" einfach acht Stunden zu setzen, also die Zeitspanne der Besichtigungszeit. Jedoch ist davon auszugehen, dass die Gruppe in diesem speziellen Fall auch die Anreisezeit zu dieser Sehenswürdigkeit bei der Besichtigungszeit inkludiert haben möchte. Hat man zum Beispiel einen Vergnügungspark auf die Reiseliste gesetzt, und bei diesem angegeben, dass man dort den ganzen Tag verweilen möchte, ist es wahrscheinlich, dass man bei diesem Tagesausflug etwas Zeit für die Anreise mit eingeplant haben möchte. Anders ist es bei den anderen Sehenswürdigkeiten, wo die Zeitangabe 30 Minuten oder zwei Stunden ist. Hier ist davon auszugehen, dass das wirklich die reine Besichtigungszeit ist. Würde man beim Ganztagsausflug im Beispiel die Zeit auf die vollen acht Stunden setzen, so könnte es außerdem passieren, dass die Anfahrt an Tag X in der Früh ist, man dann den restlichen Tag wartet und erst am nächsten Tag die eigentliche Besichtigung angesetzt wird, weil ja an Tag X dann keine acht Stunden mehr frei sind. Einfach die Anreisezeit mit einzurechnen ist aber auch keine gute Lösung, wenn die Anreise mehrere Stunden dauert. Eine sinnvolle Lösung ist daher, bei einer Ganztagsbesichtigung einen gewissen Prozentsatz des Tages für die Anfahrt einzuräumen. Für den "Holiday Recommender" wird daher die Zeit "ganzer Tag" auf die auf 85 % verkürzte Zeitspanne der Besichtigungszeit gesetzt. Somit sind 15 % des Tages für die Anfahrt freigehalten. Wenn die Startzeit für tägliche Besichtigungen  $s$  ist und die Endzeit  $e$ , ergibt sich für die Dauer  $D$  eines Ganztagsausfluges:

$$D = \frac{e - s}{100} \cdot 85 \quad (4.7)$$

Die Distanzmatrix wurde nicht von Google, sondern von MapQuest eingeholt, da MapQuest diesbezüglich die besseren Konditionen bietet. Bei Google darf die generierte Distanzmatrix maximal 100 Einträge haben [Google Developers, 2014]. Da man alle Distanzen von jedem Punkt zu jedem anderen braucht, also eine  $N \times N$  Matrix, kann man somit maximal zehn Punkte behandeln. Wenn man davon ausgeht, dass der Weg von A nach B gleich lange dauert wie der Weg von B nach A (was wegen Einbahnen und ähnlichem nicht immer genau stimmt, aber für die Applikation ausreichend wäre), könnte man maximal 20 Punkte behandeln. Dazu kommt die Bedingung, dass die Distanzmatrix nur angefordert werden darf, wenn das Ergebnis in einem Bezug zu den angezeigten Informationen einer Google-Karte steht [Google Developers, 2014]. Da die Google-Karte des "Holiday Recommenders" dann die Reiseroute anzeigt, die wahrscheinlich nicht alle Punkte aus der Distanzmatrix enthält und auch keinen direkten Bezug zu den erhaltenen Distanzen anzeigt, sondern nur einen indirekten, befindet man sich hier in einem Graubereich.

MapQuest erlaubt pro Anfrage 25 Punkte anzugeben [MapQuest API Route Matrix, 2014], was in jedem Fall mehr ist als bei Google. Auch gibt es keine Verpflichtung, die erhaltenen Daten in einer MapQuest-Karte darzustellen. Zusätzlich hat MapQuest aber noch eine Beschränkung, dass sämtliche Routen-Anfragen, was die Distanzmatrix inkludiert, 8.000 Meilen (= 12.872 Kilometer) nicht überschreiten darf. Diese Überprüfung addiert die Wegdistanz vom ersten übergebenen Punkt zum zweiten mit der Distanz vom zweiten Punkt zum dritten und so weiter [MapQuest API Locations, 2014]. MapQuest ermöglicht es also, maximal 23 Sehenswürdigkeiten einzubringen. Zwei der 25 erlaubten Punkte werden für den Startort und den Zielort benötigt.

### 4.3.3 Übersicht der Funktionsweise des Algorithmus'

Da nun alle Daten vorhanden sind, die der Algorithmus braucht, kann er gestartet werden. Zur Übersicht, wie der Algorithmus grundsätzlich funktioniert, wird in Quellcode 4.3 die grundlegende Funktionsweise dargelegt. In Zeile 1 wird zuerst ein leerer Zeitplan angelegt, in dem alle Reisetage eingetragen sind. Als nächstes wird die Variable `Anzahl_zu_löschender_Sehensw` auf den Wert 1 initialisiert. Diese Variable sagt aus, wie viele aufeinanderfolgende Sehenswürdigkeiten im Mischprozess wieder aus

dem Zeitplan entfernt werden. Die Variable `Index_Löschstart`, die im Mischprozess angibt, ab welcher Sehenswürdigkeit gelöscht werden soll, wird auf 0 initialisiert. Ebenso wird auch die Variable `Anzahl_Iterationen_ohne_Verbesserung` auf 0 initialisiert. Sie zählt mit, wie viele Iterationsschritte seit der letzten gefundenen besten Route schon vergangen sind, ohne dass die neu gefundene Route besser ist als die bisher beste. Die beste gefundene Route ist am Anfang der leere Zeitplan. Danach beginnen die Iterationsschritte, die so lange laufen, bis 300 Schritte lang keine neue bessere Route mehr gefunden wurde. Die Zahl 300 hat sich in der Praxis als sinnvoll erwiesen, da weniger Durchläufe oft keine idealen Routen liefern und mehr Durchläufe die Berechnungszeit erheblich steigern, aber nur selten bessere Routen liefern.

```

1  initZeitplan();
2  Anzahl_zu_löschender_Sehensw = 1;
3  Index_Löschstart = 0;
4  Anzahl_Iterationen_ohne_Verbesserung = 0;
5  Beste_Route = Zeitplan;
6  solange (Anzahl_Iterationen_ohne_Verbesserung < 300)
7  {
8      Einfügeprozess();
9      wenn Route besser ist als die bisher beste:
10     {
11         Beste_Route = neu_gefundene_Route;
12         Anzahl_zu_löschender_Sehensw = 1;
13         Anzahl_Iterationen_ohne_Verbesserung = 0;
14     }
15     ansonsten:
16     {
17         Anzahl_Iterationen_ohne_Verbesserung ++;
18     }
19     Mischprozess(Index_Löschstart, Anzahl_zu_löschender_Sehensw);
20     Index_Löschstart = Index_Löschstart + Anzahl_zu_löschender_Sehensw;
21     wenn Anzahl Sehensw im Zeitplan < Index_Löschstart:
22     {
23         Index_Löschstart = Zufallszahl zw. 0 und Anzahl Sehensw im Zeitplan;
24     }
25     Anzahl_zu_löschender_Sehensw = Zufallszahl zw. 1 und Anzahl aller
        Sehensw;
26 }

```

**Src. 4.3:** Stark vereinfachter Pseudo-Code des Routenalgorithmus' zur Übersicht, wie er grundsätzlich funktioniert.

In jeder Iteration werden zuerst im Einfügeprozess solange Sehenswürdigkeiten in den Zeitplan eingefügt, bis keine weiteren mehr zeitlich hineinpassen (bis ein lokales Optimum gefunden wurde), womit man eine neue Route gefunden hat, wobei diese neue Route natürlich in einem früheren Iterationsschritt schon einmal gefunden worden sein kann. Die neu erhaltene Route wird dann sofort mit Hilfe einer entsprechenden Maßfunktion überprüft, ob sie besser ist als die bisher beste Route und im gegebenen Falle als neue beste Route gespeichert. Ist sie die bisher beste Route, wird auch die Variable für die Anzahl zu löschender aufeinanderfolgender Sehenswürdigkeiten wieder auf den ursprünglichen Wert 1 zurückgesetzt. Denn es ist davon auszugehen, dass die neue gefundene beste Route recht gut ist, weswegen für die Berechnung der nächsten Route nur eine kleine Änderung zu dieser Route vorgenommen werden soll. Der Zähler, der mitprotokolliert, wie lange schon keine bessere Route mehr gefunden wurde, muss im Falle einer neuen besten Route wieder zurückgesetzt werden, im Falle eines Misserfolges wird er um eins erhöht.

Die Maßfunktion, mit der beurteilt wird, wie gut eine Route ist, muss ein vernünftiges Verhältnis der Fahrtzeiten zum Verhältnis von Qualität der Sehenswürdigkeit zu dessen Menge darstellen. Sei  $Q_i$

#### 4. Technische Beschreibung der Applikation

die Qualität einer Sehenswürdigkeit  $i$ , gemessen anhand ihrer Bewertung nach der jeweiligen Gruppenentscheidungsstrategie und sei  $T_j$  die Fahrzeit zur Sehenswürdigkeit  $j$  in Minuten. Für den "Holiday Recommender" wurde folgende Maßfunktion  $M$  für die Beurteilung einer Route erstellt, die alle  $n$  Stück der in ihr enthaltenen Sehenswürdigkeiten berücksichtigt:

$$M = \frac{(\sum_{i=1}^n (Q_i^{1,7}))^3}{\sum_{j=1}^n T_j} \quad (4.8)$$

Je höher der Wert von  $M$  ist, desto besser ist die Route. Da die Qualitäten der Sehenswürdigkeiten wesentlich mehr Einfluss auf die Qualität der Route haben, als die Anfahrtszeiten, die sie benötigen, werden die aufsummierten Qualitäten potenziert um ihnen somit mehr Gewicht zu geben. Wenn es sich um eine Sehenswürdigkeit handelt, die sehr gut bewertet wurde, wird die Reisegruppe auch eine längere Anfahrtszeit in Kauf nehmen, jedoch muss die Anfahrtszeit dennoch in einem vernünftigen Verhältnis zu anderen Alternativen bleiben. Der Faktor drei hat sich dabei als gut erwiesen, bei einer Quadratur wurden schlechtere Ergebnisse erreicht. Bei einer geringeren Potenzierung entsteht zusätzlich das Problem, dass eine Route, die weniger Sehenswürdigkeiten enthält, oft besser bewertet wird als eine Route, die viele Sehenswürdigkeiten beinhaltet. Hier mit Gleichung 4.9 und Gleichung 4.10 ein Beispiel: In Route  $A$  seien zwölf Sehenswürdigkeiten enthalten, deren aufsummierte Qualitäten 130 ergibt, also quadriert dann 16.900 ergibt und die Fahrzeit 750 Minuten beträgt, dann wäre das Ergebnis  $M$  ungefähr 22,53. Seien nun in Route  $B$  nur sieben gleich gut bewertete Sehenswürdigkeiten enthalten, erhalten wir für ihre aufsummierten Qualitäten 74, also quadriert dann 5.476 und würde die Fahrzeit in Route  $B$  dann entsprechend kürzer sein, also nur 240 Minuten betragen, dann wäre das Ergebnis  $M$  für Route  $B$  ungefähr 22,82:

$$M = \frac{(\sum_{i=1}^n (Q_i^{1,7}))^2}{\sum_{j=1}^n T_j} \quad \text{eingesetzt für:} \quad \begin{cases} \text{Route A: } \frac{130^2}{750} = 22,5\dot{3} \\ \text{Route B: } \frac{74^2}{240} \approx 22,82 \end{cases} \quad (4.9)$$

Dieses Beispiel zeigt, dass man für Route  $B$  einen höheren Wert erhält als für Route  $A$ . Das heißt, dass in diesem Fall die Route mit weniger Sehenswürdigkeiten gewinnt, obwohl die Sehenswürdigkeiten in beiden Routen im Schnitt gleich gut bewertet sind. Nimmt man nun selbiges Beispiel und verwendet als Potenz die Zahl drei ergäbe sich für Route  $A$  ein Ergebnis von 2.929,33 und für Route  $B$  ein Ergebnis von 1.688,43:

$$M = \frac{(\sum_{i=1}^n (Q_i^{1,7}))^3}{\sum_{j=1}^n T_j} \quad \text{eingesetzt für:} \quad \begin{cases} \text{Route A: } \frac{130^3}{750} = 2.929,33 \\ \text{Route B: } \frac{74^3}{240} = 1.688,43 \end{cases} \quad (4.10)$$

In Gleichung 4.10 sieht man, dass Route  $A$  im Gegensatz zu Gleichung 4.9 deutlich besser abschneidet, also Routen, die mehr Sehenswürdigkeiten enthalten als eine Route mit im Schnitt gleich gut bewerteten Sehenswürdigkeiten, die aber weniger Sehenswürdigkeiten abdeckt, besser gewertet werden. Bei einer Potenz von drei müssen dabei schon große Umwege gemacht werden, damit eine Route doch schlechter bewertet wird als eine von den Sehenswürdigkeiten her ähnlich gute Route mit weniger Sehenswürdigkeiten.

Bei der Potenzierung der einzelnen Qualitäten vor deren Aufsummierung geht es darum, ein geeignetes Gewicht zu finden, sodass viele Sehenswürdigkeiten mit schlechter Bewertung in Summe nicht besser gewertet werden als wenige, aber dafür gute Sehenswürdigkeiten. Hier geht es also darum, dass die Qualität der Sehenswürdigkeiten wichtiger ist als deren Quantität. Keine Reisegruppe hätte etwas davon, viele Sehenswürdigkeiten zu sehen, die sie schlecht bewertet haben. Besser wäre es hingegen, nur wenige ausgesuchte Sehenswürdigkeiten zu sehen, die dafür aber gut bewertet wurden. Würde man die Qualitäten gar nicht gewichten, wäre eine Route, die zwölf Sehenswürdigkeiten, die alle mit "wenig

sehenswert" oder "nicht sehenswert" bewertet wurden besser abschneiden als eine, die nur fünf Sehenswürdigkeiten enthält, die dafür aber alle mit "sehenswert" oder "sehr sehenswert" bewertet wurden, wie der Vergleich in Gleichung 4.11 zeigt:

$$\begin{aligned}(\alpha) \quad & 1 + 1 + 1 + 1 + 2 + 2 + 2 + 2 + 2 + 2 + 2 + 2 = 4 \cdot 1 + 8 \cdot 2 = 20 \\(\beta) \quad & 5 + 5 + 4 + 4 + 4 = 2 \cdot 5 + 3 \cdot 4 = 17\end{aligned} \quad (4.11)$$

Aus Gleichung 4.11 folgt also, dass eine Gewichtung der einzelnen Qualitäten unbedingt notwendig ist. Nimmt man als Gewicht eine Quadratur an, so wäre das Problem aus Gleichung 4.11 gelöst, wie Gleichung 4.12 in (a) und (b) zeigt, jedoch würde die Besichtigung einer "sehr sehenswerten" Sehenswürdigkeit mit einer weiteren "nicht sehenswerten" Sehenswürdigkeit einer Besichtigung einer "sehenswerten" Sehenswürdigkeit mit einer weiteren "eventuell sehenswerten" Sehenswürdigkeit vorgezogen werden, wie Gleichung 4.12 in (e) und (f) zeigt:

$$\begin{aligned}(\text{a}) \quad & 4 \cdot 1^2 + 8 \cdot 2^2 = 36 \\(\text{b}) \quad & 2 \cdot 5^2 + 3 \cdot 4^2 = 98 \\(\text{c}) \quad & 3 \cdot 1^2 + 4 \cdot 2^2 = 19 \\(\text{d}) \quad & 4^2 + 4^2 = 32 \\(\text{e}) \quad & 5^2 + 1^2 = 26 \\(\text{f}) \quad & 4^2 + 3^2 = 25\end{aligned} \quad (4.12)$$

Dass die Besichtigung zweier "sehenswerten" Sehenswürdigkeiten gegenüber den anderen am besten abschneidet (Fall (d)), ist sicherlich im Sinne der Reisegruppe. Aber es ist davon auszugehen, dass die Gruppe die Besichtigung von Fall (f) einer Besichtigung von Fall (e) vorziehen wird, was dem Ergebnis einer Quadratur widerspricht. Einige Experimente mit verschiedenen Routen haben ergeben, dass eine Potenzierung mit Faktor 1,7 die besten Ergebnisse liefert. Auch für das Beispiel in Gleichung 4.12 liefert dies gute Ergebnisse, wie Gleichung 4.13 zeigt:

$$\begin{aligned}(\text{a}) \quad & 4 \cdot 1^{1,7} + 8 \cdot 2^{1,7} = 30 \\(\text{b}) \quad & 2 \cdot 5^{1,7} + 3 \cdot 4^{1,7} = 62,5 \\(\text{c}) \quad & 3 \cdot 1^{1,7} + 4 \cdot 2^{1,7} = 16 \\(\text{d}) \quad & 4^{1,7} + 4^{1,7} = 21,1 \\(\text{e}) \quad & 5^{1,7} + 1^{1,7} = 16,4 \\(\text{f}) \quad & 4^{1,7} + 3^{1,7} = 17\end{aligned} \quad (4.13)$$

Auch hier wird Fall (b) dem Fall (a) deutlich vorgezogen. Ebenso wird hier Fall (f) dem Fall (e) vorgezogen im Gegensatz zur Quadratur der Qualitäten. Auch Fall (d) gewinnt gegenüber Fall (e) und Fall (f). Es fällt auf, dass Fall (a) und Fall (b) gegenüber allen anderen Fällen gewinnen würde, jedoch ist es unwahrscheinlich, dass sich im selben Zeitfenster einmal zwölf Sehenswürdigkeiten ausgehen und einmal nur zwei. Es müsste also schon ein sehr deutlicher Unterschied in der Anzahl der Sehenswürdigkeiten auftreten, sodass eine Route mit vielen, eher schlechteren Sehenswürdigkeiten gegenüber einer mit wenigen, aber guten Sehenswürdigkeiten gewinnt. Fall (c) zeigt, dass bei einer Anzahl von sieben eher schlechten Sehenswürdigkeiten diese Route bereits gegenüber Fall (e) und Fall (f) mit nur zwei Sehenswürdigkeiten verliert, was von der Reisegruppe sicherlich gewünscht wird. Da bei den Routenberechnungen eher nur geringe Unterschiede in der Anzahl der Sehenswürdigkeiten auftreten werden, ist es

## 4. Technische Beschreibung der Applikation

wichtiger, sich auf die Unterschiede bei Routen mit ähnlich vielen Sehenswürdigkeiten zu konzentrieren, wobei hier eine Potenzierung der Qualitäten mit dem Faktor 1,7 die besten Ergebnisse liefert.

Nachdem nun der Einfügeprozess abgeschlossen ist und die erhaltene Route bewertet wurde, wird als nächstes der Mischprozess vorgenommen. In diesem Schritt werden von `Index_Löschstart` an `Anzahl_zu_löschender_Sehensw` viele Sehenswürdigkeiten wieder aus dem Zeitplan entfernt, um lokalen Optima zu entkommen, denn im nächsten Iterationsschritt wird wieder versucht, im Einfügeprozess die entfernten Sehenswürdigkeiten an der bestmöglichen Stelle einzufügen. Nachdem einige Sehenswürdigkeiten im Mischprozess entfernt wurden, werden ebenfalls noch im Mischprozess alle Sehenswürdigkeiten, die im Zeitplan hinter der herausgenommenen Menge lagen, nach vorne gereiht, sodass die entstandene Lücke aufgefüllt ist, falls nicht die letzten Sehenswürdigkeiten gelöscht wurden oder sogar alle. Nach dem Mischprozess muss noch ein neuer Index für den Löschbeginn berechnet werden, sowie die Anzahl zu löschender Sehenswürdigkeiten für den nächsten Iterationsschritt. Der neue Index, ab dem begonnen wird zu löschen, errechnet sich grundsätzlich aus dem alten Löschbeginn-Index addiert zur alten Anzahl zu löschender Sehenswürdigkeiten. Sollte diese Summe aber höher sein als die Anzahl an Sehenswürdigkeiten im aktuellen Zeitplan, wird für den nächsten Löschbeginn-Index eine Zufallszahl zwischen 0 und der Anzahl an Sehenswürdigkeiten im aktuellen Zeitplan genommen. Da zunächst im Einfügeprozess des nächsten Iterationsschrittes nur noch Sehenswürdigkeiten dazu kommen können, aber es nicht weniger werden, besteht also keine Gefahr, dass dieser Index für den nächsten Mischprozess zu hoch gewählt wurde. Für die neue Anzahl zu löschender Sehenswürdigkeiten wird immer eine Zufallszahl zwischen 1 und der Anzahl aller Sehenswürdigkeiten gewählt. Ist diese Anzahl höher als die Anzahl an Sehenswürdigkeiten im Zeitplan der nächsten Iteration, werden einfach alle Sehenswürdigkeiten aus dem Zeitplan gelöscht. Dadurch, dass der Löschbeginn und die Anzahl zu löschender Sehenswürdigkeiten immer variieren, ist die Wahrscheinlichkeit groß, dass jede Sehenswürdigkeit, die in der Reise enthalten sei soll, mehrmals gelöscht und wieder eingefügt wird und somit in der besten Lösung an einem wirklich passenden Platz landet.

In der Arbeit von Vansteenwegen u. a. [2009] werden der Index für den Löschbeginn sowie die Anzahl zu löschender Sehenswürdigkeiten anders berechnet. Deren Berechnungen gehen aber von einzelnen, unabhängigen Tagestouren aus, was nicht zum gegebenen Fall des "Holiday Recommenders" passt und weiters landet man schnell in einer sich immer wiederholenden Schleife, wenn man die Werte fix berechnet, weswegen die Wahl von Zufallszahlen eine weit effizientere Art für den Algorithmus ist.

Da der Einfüge- und der Lösprozess große Zeiteinsparungen und Optimierungen im Gegensatz zu einem wahllosen Einfügen und Löschen aufweisen und im Einfügeprozess weiters die Auswahl der Sehenswürdigkeiten enthalten ist, wird im Folgenden genauer auf diese beiden Prozesse eingegangen.

### 4.3.4 Der Einfügeprozess

Im Einfügeprozess werden die Sehenswürdigkeiten nacheinander in den Zeitplan an den besten Platz eingefügt. Dabei wird jene Sehenswürdigkeit, die an ihrer Bewertung und der benötigten Mehrzeit bemessen am besten abschneidet, als nächste Einzufügende genommen. Dabei muss natürlich auch überprüft werden, dass sich alle zeitlich nachstehenden Sehenswürdigkeiten immer noch zeitlich ausgehen. Damit dies in sinnvoller Zeit geschehen kann, hält jede bereits eingefügte Sehenswürdigkeit ein paar Variablen, mit Hilfe deren schnell evaluiert werden kann, ob eine neue Sehenswürdigkeit an einer bestimmten Stelle eingefügt werden kann oder nicht. Jedes Mal alle nachfolgenden Sehenswürdigkeiten zu überprüfen, ob sie sich zeitlich noch ausgehen, würde zu viel Zeit in Anspruch nehmen. In diesem Algorithmus muss bei jeder Positionsüberprüfung immer nur die nachfolgende Sehenswürdigkeit überprüft werden, da diese Variablen hält, die auch alle notwendigen Informationen über alle nachfolgenden Sehenswürdigkeiten indirekt beinhalten. Diese zeitliche Beschleunigung des Algorithmus' ist stark angelehnt an die Arbeit von Vansteenwegen u. a. [2009]. Es wurde lediglich ein Fehler ausgebessert und die Idee an die Gegebenheiten des "Holiday Recommenders" angepasst, sodass sie für Routen, bei denen alle Tage als Gesamtroute gesehen werden und nicht als Einzelrouten, verwendet werden kann.

Im Gegensatz zum Algorithmus von Vansteenwegen u. a. [2009] werden hier auch die Nächte im Zeitplan mit berücksichtigt. Die Nacht stellt einen ungültigen Zeitraum dar, in dem keine Sehenswürdigkeiten besichtigt werden dürfen. Ab 08 Uhr in der Früh und bis 18 Uhr am Abend dürfen aber Anfahrten zur nächsten Sehenswürdigkeit vorgenommen werden, unabhängig vom Besichtigungszeitraum, den die Reisegruppe angegeben hat. Da im “Holiday Recommender” die Orte der Hotels und die Anfahrtszeiten zu ihnen nicht von den Benutzern angegeben werden, da bei einer Rund- oder Streckenreise die jeweiligen Übernachtungsorte erst nach der Routenberechnung bekannt werden würden, werden diese Reisezeiten bei der Routenberechnung zusätzlich mitberücksichtigt. Es wurde dabei angenommen, dass die Reisegruppe ein Hotel wählen wird, das örtlich gesehen zwischen der letzten Sehenswürdigkeit des Tages und der ersten Sehenswürdigkeit des nächsten Tages liegen wird. Es wird also keine zusätzliche Zeit für die Fahrt ins Hotel benötigt, sondern die Übernachtung stellt lediglich einen Punkt, der sowieso auf der Route liegt, dar. Es geht nur darum, einen Teil der Fahrt zur ersten Sehenswürdigkeit des nächsten Tages als Fahrt zu einem Hotel zu sehen, die noch nach der Besichtigung der letzten Sehenswürdigkeit des Tages angetreten wird.

Daher wird im Reiseplan eine Reisezeit nach Besichtigung der letzten Sehenswürdigkeit des Tages bis maximal 18 Uhr ermöglicht. Hat die Reisegruppe ihr Ende für Besichtigungen auch mit 18 Uhr gewählt, ist davon auszugehen, dass sie ein Hotel wählen werden, das sich in der Nähe der letzten Sehenswürdigkeit befindet, womit automatisch keine Teilreisezeit mehr möglich ist. Am nächsten Tag gilt analoges für die Anreisezeit zur ersten Sehenswürdigkeit in der Früh. In der Früh darf eine Anreisezeit bzw. Weiterfahrt bereits um 08 Uhr beginnen, unabhängig davon, welchen Besichtigungszeitraum die Reisegruppe angegeben hat. Der Beginn der Besichtigung einer Sehenswürdigkeit darf natürlich erst ab der Zeit erfolgen, die die Reisegruppe unter den Reisedaten angegeben hat. Will die Gruppe erst ab 11 Uhr besichtigen und die verbleibende Restanreisezeit zur ersten Sehenswürdigkeit des Tages wäre 45 Minuten, so würde die Anreisezeit zur ersten Sehenswürdigkeit schon um 10:15 Uhr beginnen. (Im Algorithmus intern wird die Anreise um 08 Uhr begonnen und dann am Ziel bis 11 Uhr gewartet.) Will die Gruppe bereits ab 08 Uhr besichtigen, darf die Anreise trotzdem erst um 08 Uhr beginnen und die Besichtigung verschiebt sich nach hinten, da davon auszugehen ist, dass die Gruppe nicht schon vor 08 Uhr vom Hotel aufbrechen möchte.

Somit gilt für den Einfügeprozess, dass für die Besichtigungszeiten der von der Gruppe angegebene Zeitraum genau eingehalten werden muss, jedoch die Anfahrtszeiten in einem erweiterten Rahmen von 08 Uhr bis zur angegebenen Startzeit und von der angegebenen Endzeit bis 18 Uhr stattfinden dürfen. Zwischen 18 Uhr und 08 Uhr dürfen weder Fahrten noch Besichtigungen vorgenommen werden. Wenn sich also eine Sehenswürdigkeit an einem Tag nicht mehr ausgeht, darf eine Teilreisezeit bis 18 Uhr in den Zeitplan eingetragen werden und am nächsten Tag die verbleibende Zeit ab 08 Uhr weitergefahren werden.

Der Einfügeprozess läuft in einer Schleife, in der solange Sehenswürdigkeiten in den Zeitplan eingefügt werden, bis ein lokales Optimum erreicht wurde, das heißt, solange sich noch Sehenswürdigkeiten in den Zeitplan einfügen lassen. Ein stark vereinfachter Pseudo-Code des Einfügeprozesses ist in Quellcode 4.4 gegeben. Als erstes muss jene Sehenswürdigkeit samt ihrem Einfügeplatz gefunden werden, die von ihrer Bewertung und der zusätzlich benötigten Zeit her am besten passt. Dazu wird für jede Sehenswürdigkeit, die sich noch nicht im Zeitplan befindet, versucht, sie an jeder möglichen Position im Zeitplan einzufügen. Für jede mögliche Position wird eine Kennzahl `Ratio` ermittelt. Die Sehenswürdigkeit, die den höchsten `Ratio`-Wert erreicht, wird dann an der Position eingefügt, wo dieser höchste `Ratio` aufgetreten ist. Der `Ratio` einer Sehenswürdigkeit an Platz  $i$  eingefügt, wird berechnet als Verhältnis der quadrierten Qualität  $Q$  der einzufügenden Sehenswürdigkeit zur zusätzlich benötigten Zeit `Shift_Netto`, um die die Reise durch das Einfügen der neuen Sehenswürdigkeit an der Stelle  $i$  nun länger dauern würde als zuvor:

$$Ratio_i = \frac{(Q_i)^2}{Shift\_Netto_i} \quad (4.14)$$

## 4. Technische Beschreibung der Applikation

```
1  solange kein lokales Optimum erreicht:
2  {
3    für jede noch nicht eingefügte Sehenswürdigkeit:
4    {
5      für jede Position im Zeitplan:
6      {
7        überprüfe, ob Position im erlaubten Zeitfenster ist;
8        wenn ja:
9        {
10         berechne Starttag, Endtag, Ankunftszeit, Wait,
11           Besichtigungsstart, Besichtigungsende, MaxShift, Shift und
12           Ratio;
13         überprüfe, ob nachstehende Sehenswürdigkeiten noch möglich
14       }
15     }
16     füge Sehenswürdigkeit mit höchstem Ratio ein
17     für jede nachfolgende Sehensw. im Zeitplan:
18     {
19       update Ankunftszeit, Wait, Besichtigungsstart, Besichtigungsende,
20         MaxShift und Shift
21     }
22     update MaxShift der eingefügten Sehenswürdigkeit
23     für jede vorherige Sehensw. im Zeitplan:
24     {
25       update MaxShift
26     }
27   }
28 }
```

**Src. 4.4:** Stark vereinfachter Pseudo-Code des Einfügeprozesses von Sehenswürdigkeiten in den Zeitplan zur Übersicht, wie er grundsätzlich funktioniert.

Da es wichtig ist, eine Sehenswürdigkeit mit guter Qualität einzufügen und die dafür länger benötigte Zeit im Verhältnis dazu nur eine untergeordnete Rolle spielt, wird der Wert  $Q$  quadriert [Vansteenwegen u. a., 2009]. Die durch das Einfügen an Stelle  $i$  (zwischen Stelle  $h$  und Stelle  $j$ ) länger benötigte Zeit  $Shift\_Netto$  wird berechnet indem man die Fahrtzeit  $c_{hi}$  von der Sehenswürdigkeit an der Stelle  $h$  bis zur neuen Sehenswürdigkeit an Stelle  $i$ , die Wartezeit  $Wait\_Netto$  die man eventuell warten muss, falls man vor dem Besichtigungszeitraum ankommt, die Besichtigungsdauer  $T$  sowie die Fahrtzeit  $c_{ij}$  von der Sehenswürdigkeit an Stelle  $i$  zur nächsten Sehenswürdigkeit an Stelle  $j$ , aufsummiert und davon die alte Strecke  $c_{hj}$  von Sehenswürdigkeit an Stelle  $h$  zu Sehenswürdigkeit an Stelle  $j$  abzieht. Liegt zwischen der Anfahrtszeit und der Besichtigung eine Nacht, so wird diese nicht mit berücksichtigt, da es sich hierbei um keine wirkliche Wartezeit in dem Sinne handelt und daher das Ergebnis verfälschen würde. Denn würde man die Nacht als Wartezeit mitzählen, wäre  $Shift\_Netto$  viel höher als wenn man Anreise und Besichtigung an einem Tag schaffen würde. Somit wären Teilfahrtzeiten mit Übernachtung dazwischen schlechter, was aber nicht der Fall ist. Für  $Shift\_Netto$  ergibt sich also:

$$Shift\_Netto_i = c_{hi} + Wait\_Netto_i + T_i + c_{ij} - c_{hj} \quad (4.15)$$

Da sich durch ein Einfügen der neuen Sehenswürdigkeit an der Stelle  $i$  die nachfolgenden Sehenswürdigkeiten zeitlich nach hinten verschieben würden, ist es für die Berechnung der Ankunftszeiten der nachfolgenden Sehenswürdigkeiten notwendig, die Nächte mitzurechnen. Daher benötigt man den  $Shift\_Brutto$ -Wert, der sich vom  $Shift\_Netto$ -Wert nur dadurch unterscheidet, dass beim  $Shift\_Brutto$ -Wert auch die Nacht-Zeiten mitberücksichtigt werden:

$$Shift\_Brutto_i = c_{hi} + Wait\_Brutto_i + T_i + c_{ij} - c_{hj} \quad (4.16)$$

Natürlich muss für jedes Einfügen gewährleistet werden, dass sich die dahinter befindlichen Sehenswürdigkeiten zeitlich immer noch ausgehen. Jede Sehenswürdigkeit, die im Zeitplan bereits eingefügt wurde, hält daher neben der Variablen `Shift_Netto` und `Shift_Brutto` noch die Variablen `Wait_Netto`, `Wait_Brutto` und die Variable `MaxShift`. `Wait_Netto` gibt an, wie lange man ab der Ankunftszeit  $a_i$  bei der Sehenswürdigkeit an der Stelle  $i$  warten muss, bis die Sehenswürdigkeit besichtigt werden darf. Eine Übernachtung wird dabei nicht berücksichtigt, um das Ergebnis von `Shift_Netto` nicht zu verfälschen. Ab 08 Uhr in der Früh darf eine Anreise beginnen. Hat die Reisegruppe als täglichen Besichtigungsbeginn zum Beispiel 10 Uhr gewählt und man würde, wenn man um 08 Uhr losfahren würde, um 09:40 Uhr bei der Sehenswürdigkeit ankommen, dann wäre auch der Besichtigungsstart  $s_i$  für die Sehenswürdigkeit um 10 Uhr und somit der Wert `Wait_Netto` = 20 Minuten.

Da im “Holiday Recommender” vorerst nur die einzelnen Tage, an denen eine Sehenswürdigkeit besichtigt werden darf oder nicht, aus- oder abgewählt werden können, entsprechen die Öffnungs- und Schließungszeiten der Sehenswürdigkeiten dem täglichen Besichtigungsbeginn bzw. dem täglichen Besichtigungsende. Im Algorithmus wurde darauf geachtet, dass der tägliche Besichtigungsbeginn und die Öffnungszeit einer Sehenswürdigkeit immer in getrennten Variablen gespeichert werden, womit der Algorithmus auch problemlos ohne weitere Wartung funktioniert, wenn sich die Öffnungszeiten nicht auf die gesamte erlaubte Besichtigungsdauer eines Tages ausdehnen würden.

`Wait_Brutto` gibt an, wie lange man ab der Ankunftszeit  $a_i$  bei der Sehenswürdigkeit an der Stelle  $i$  warten muss, bis die Sehenswürdigkeit besichtigt werden darf (Besichtigungsstart  $s_i$ ), wobei hier die Übernachtungszeit mitgerechnet wird, um den `Shift_Brutto`-Wert für das Update der nachfolgenden Sehenswürdigkeiten (sollte die Sehenswürdigkeit wirklich an dieser Stelle eingefügt werden) korrekt vornehmen zu können. Ist die Ankunft bei der Sehenswürdigkeit sowie auch deren Besichtigung am selben Tag, so ergeben sich der `Wait_Netto`-Wert und der `Wait_Brutto`-Wert:

$$\text{Wait\_Netto}_i = \text{Wait\_Brutto}_i = \max\{0, s_i - a_i\} \quad (4.17)$$

Sind die Ankunft und die Besichtigung an verschiedenen Tagen, was passieren kann, wenn man zum Beispiel an einem Dienstag versucht, die Sehenswürdigkeit einzufügen, sich die Anreise am Dienstag auch ausgeht, die Sehenswürdigkeit aber am Dienstag nicht besichtigt werden darf, sondern erst am Mittwoch, dann ergibt sich eine Wartezeit, die sich von der Ankunft  $a_i$  am Dienstag bis zum Besichtigungsbeginn  $s_i$  am Mittwoch erstreckt. In diesem Falle unterscheiden sich `Wait_Netto` und `Wait_Brutto` von einander. Sei  $e$  die Reiseendzeit für jeden Tag, also 18 Uhr und  $b$  die Reisebeginnzeit für jeden Tag, also 08 Uhr und seien  $n$  volle Tage zwischen Ankunftstag und Besichtigungstag (also Tage, an denen weder die Ankunft noch die Besichtigung stattfindet, die aber zwischen Ankunftstag und Besichtigung liegen), dann ergibt sich für `Wait_Netto`:

$$\text{Wait\_Netto}_i = e - a_i + (n \cdot (e - b)) + s_i - b \quad (4.18)$$

Für `Wait_Brutto` werden im Falle dessen, dass Ankunftstag und Besichtigungstag nicht der selbe sind, die gesamten Stunden von der Ankunftszeit bis zum Besichtigungsbeginn (also inklusive der Nachtzeiten) aufsummiert:

$$\text{Wait\_Brutto}_i = 24:00 \text{ Uhr} - a_i + (n \cdot 24 \text{ Stunden}) + s_i - 0:00 \text{ Uhr} \quad (4.19)$$

Die Variable `MaxShift` einer Sehenswürdigkeit gibt an, wie weit diese und die zeitlich dahinter befindlichen Sehenswürdigkeiten nach hinten verschoben werden dürfen, sodass sich alle noch in ihrem Zeitfenster ausgehen. Der `MaxShift` für die Sehenswürdigkeit an der Stelle  $i$  entspricht daher entweder dem Besichtigungsendzeitpunkt  $E_i$  der Sehenswürdigkeit an Stelle  $i$ , wann ein Besuch dieser Sehenswürdigkeit spätestens beendet sein muss (Schließungszeit), minus dem Startzeitpunkt des Besichtigungsbeginns  $s_i$  minus der Besichtigungsdauer  $T_i$  oder der Wartezeit `Wait_Netto` der nächsten

#### 4. Technische Beschreibung der Applikation

Sehenswürdigkeit an der Stelle  $j$  plus dem  $MaxShift$ -Wert der Sehenswürdigkeit an der Stelle  $j$ , je nach dem, welcher Wert der kleinere von beiden ist:

$$MaxShift_i = \min\{E_i - s_i - T_i, Wait\_Netto_j + MaxShift_j\} \quad (4.20)$$

Nun sind alle benötigten Variablen bekannt und es muss überprüft werden, ob die Sehenswürdigkeit an Stelle  $i$  (zwischen Stelle  $h$  und Stelle  $j$ ) wirklich eingefügt werden kann oder nicht. Um die Sehenswürdigkeit einfügen zu können muss gewährleistet sein, dass der berechnete  $Shift\_Netto$ -Wert für die neue Sehenswürdigkeit, die an der Stelle  $i$  eingefügt wird, nicht größer ist als die Summe von  $Wait\_Netto$  und  $MaxShift$  der nachfolgenden Sehenswürdigkeit, die an Stelle  $j$  ist:

$$Shift\_Netto_i = c_{hi} + Wait\_Netto_i + T_i + c_{ij} - c_{hj} \leq Wait\_Netto_j + MaxShift_j \quad (4.21)$$

Nachdem alle  $Ratio$ -Werte für jede noch nicht eingefügte Sehenswürdigkeit an jeder möglichen und erlaubten Einfügestelle berechnet wurden, wird die Sehenswürdigkeit mit dem höchsten  $Ratio$ -Wert an der Stelle des Zeitplans, wo dieser aufgetreten ist, eingefügt. Dabei werden auch alle berechneten Werte (Ankunftszeit,  $Wait\_Netto$ ,  $Wait\_Brutto$ , Besichtigungsstart,  $MaxShift$ ,  $Shift\_Netto$ ,  $Shift\_Brutto$ ,  $Ratio$ , Starttag, Endtag) mit in den Zeitplan eingetragen. Gehen sich Anfahrt und Besichtigung nicht an einem Tag aus, dann wird als letztes am Tag vor der Besichtigung (Starttag) eine Teilfahrzeit eingetragen und die restliche Fahrzeit inklusive Besichtigung am nächsten Tag (Endtag). Zwischen einer Teilreisezeit und der Besichtigung am nächsten Tag darf natürlich keine weitere Sehenswürdigkeit eingefügt werden.

Immer, nachdem eine Sehenswürdigkeit eingefügt wird, müssen die Werte aller nachfolgenden Sehenswürdigkeiten upgedatet werden, beginnend mit der Sehenswürdigkeit direkt nach der eingefügten Sehenswürdigkeit fortlaufend bis zur letzten Sehenswürdigkeit des aktuellen Zeitplans. Die Ankunftszeit für die nachfolgende Sehenswürdigkeit an Stelle  $j$  wird dabei berechnet, indem man zur alten Ankunftszeit  $a_j$  den  $Shift\_Brutto$ -Wert der davor liegenden Sehenswürdigkeit an der Stelle  $i$  addiert:

$$a_{j\_neu} = a_{j\_alt} + Shift\_Brutto_i \quad (4.22)$$

Nach selbigem Muster werden auch die Ankunftszeiten aller nachfolgenden Sehenswürdigkeiten upgedatet (nachdem dann alle anderen Updates der jeweiligen bisherigen Sehenswürdigkeiten abgeschlossen sind - es müssen erst alle Updates einer Sehenswürdigkeit abgeschlossen sein, bevor mit einem Update der darauffolgenden Sehenswürdigkeit begonnen werden darf). Als nächstes werden  $Wait\_Netto$  und  $Wait\_Brutto$  upgedatet. Die neuen  $Wait$ -Werte ergeben sich, indem man von den alten  $Wait$ -Werten die  $Shift$ -Werte der vorherigen Sehenswürdigkeit abzieht. Ist das Ergebnis negativ, ist der neue  $Wait$ -Wert 0:

$$Wait\_Netto_{j\_neu} = \max\{0, Wait\_Netto_{j\_alt} - Shift\_Netto_i\} \quad (4.23)$$

$$Wait\_Brutto_{j\_neu} = \max\{0, Wait\_Brutto_{j\_alt} - Shift\_Brutto_i\} \quad (4.24)$$

Für das Update der  $Shift$ -Werte wird vom  $Shift$ -Wert der davor liegenden Sehenswürdigkeit die eben erneuerte Wartezeit  $Wait$  der upzudatenden Sehenswürdigkeit subtrahiert. Ist das Ergebnis negativ, ist der neue  $Shift$ -Wert 0:

$$Shift\_Netto_{j\_neu} = \max\{0, Shift\_Netto_i - Wait\_Netto_{j\_neu}\} \quad (4.25)$$

$$Shift\_Brutto_{j\_neu} = \max\{0, Shift\_Brutto_i - Wait\_Brutto_{j\_neu}\} \quad (4.26)$$

Der neue Besichtigungsbeginn  $s_{j\_neu}$  ergibt sich aus der Addition des ehemaligen Besichtigungsbeginns und des eben erneuerten  $Shift\_Netto$ -Wertes der upzudatenden Sehenswürdigkeit:

$$s_{j\_neu} = s_{j\_alt} + Shift\_Netto_{j\_neu} \quad (4.27)$$

Selbiges gilt für das neue Besichtigungsende  $e_{j\_neu}$ :

$$e_{j\_neu} = e_{j\_alt} + Shift\_Netto_{j\_neu} \quad (4.28)$$

Für das Update des  $MaxShift$ -Wertes wird vom alten  $MaxShift$ -Wert der aktuelle  $Shift$ -Wert der upzudatenden Sehenswürdigkeit subtrahiert:

$$MaxShift_{j\_neu} = MaxShift_{j\_alt} - Shift\_Netto_{j\_neu} \quad (4.29)$$

Da diese Updates im Laufe des Algorithmus' häufig gemacht werden müssen, ist es wichtig, hierbei so zeitoptimiert wie möglich zu arbeiten. Da bei den Updates immer nur der  $Shift$ -Wert ausschlaggebend für das neue Ergebnis ist, kann das Update abgebrochen werden, sobald der  $Shift\_Netto$ -Wert und der  $Shift\_Brutto$ -Wert einer eben upgedateten Sehenswürdigkeit 0 sind, denn dann ändern sich die Werte der nachfolgenden Sehenswürdigkeiten nicht mehr.

Alle Werte der Sehenswürdigkeiten, die sich im Zeitplan hinter der neuen Sehenswürdigkeit, die an der Stelle  $i$  eingefügt wurde, befinden, sind nun auf dem aktuellen Stand. Nun muss auch der  $MaxShift$ -Wert der gerade eingefügten Sehenswürdigkeit upgedatet werden, da der  $MaxShift$ -Wert auch von den Werten der nachfolgenden Sehenswürdigkeit abhängt. Für dieses Update wird Gleichung 4.20 verwendet, die bereits für das Einfügen der neuen Sehenswürdigkeit verwendet wurde.

Wenn sich der  $MaxShift$ -Wert der gerade eingefügten Sehenswürdigkeit geändert hat, müssen nun auch die  $MaxShift$ -Werte aller vorhergehenden Sehenswürdigkeiten upgedatet werden, solange, bis entweder auch die allererste Sehenswürdigkeit geändert wurde oder der  $MaxShift$ -Wert einer Sehenswürdigkeit gleich bleibt. Auch für dieses Update wird Gleichung 4.20 verwendet, die bereits für das Einfügen der neuen Sehenswürdigkeit verwendet wurde.

### 4.3.5 Der Mischprozess

Wie bereits erwähnt, dient der Mischprozess dazu, um lokalen Optima zu entkommen und somit das Annähern eines globalen Optimums zu ermöglichen. Zuerst werden daher  $Anzahl\_zu\_löschen\_der\_Sehensw$  viele Sehenswürdigkeiten aus dem aktuellen Zeitplan entfernt, wobei gilt:

$$Anzahl\_zu\_löschen\_der\_Sehensw \in [1, Anzahl\_aller\_Sehensw] \quad (4.30)$$

Dieser Wert für die Anzahl zu löschender Sehenswürdigkeiten wird jeweils im vorhergehenden Iterationsschritt ermittelt. Im vorherigen Iterationsschritt ist die Anzahl an Sehenswürdigkeiten, die im nächsten Schritt wieder eingefügt wird, noch nicht bekannt, weswegen das Maximum der Variable  $Anzahl\_zu\_löschen\_der\_Sehensw$  auf die Anzahl aller Sehenswürdigkeiten gesetzt wird.

Da unter Umständen die Anzahl an Sehenswürdigkeiten im aktuellen Zeitplan kleiner ist als die Anzahl aller Sehenswürdigkeiten, muss vor dem Löschen überprüft werden, ob die  $Anzahl\_zu\_löschen\_der\_Sehensw$  kleiner oder gleich der Anzahl an Sehenswürdigkeiten aus dem aktuellen Zeitplan ist. Ist sie größer, werden einfach alle Sehenswürdigkeiten gelöscht, das heißt,  $Anzahl\_zu\_löschen\_der\_Sehensw = Anzahl\_Sehensw\_im\_Zeitplan$ :

$$\begin{aligned} \text{Wenn } Anzahl\_zu\_löschen\_der\_Sehensw \geq Anzahl\_Sehensw\_im\_Zeitplan : \\ Anzahl\_zu\_löschen\_der\_Sehensw = Anzahl\_Sehensw\_im\_Zeitplan \end{aligned} \quad (4.31)$$

## 4. Technische Beschreibung der Applikation

Um Sehenswürdigkeiten zu löschen, wird auch die Variable `Index_Löschbeginn` benötigt, die angibt, ab welcher Position Sehenswürdigkeiten aus dem Zeitplan gelöscht werden sollen. Angenommen, `Index_Löschbeginn=2` und `Anzahl_zu_löschender_Sehensw=5` und es befinden sich zehn Sehenswürdigkeiten im aktuellen Zeitplan. Dann werden die Sehenswürdigkeiten an den Positionen 2, 3, 4, 5 und 6 gelöscht, wenn die erste Sehenswürdigkeit im Zeitplan `Position = 1` hat. Es kann natürlich vorkommen, dass bei anderen Werten das Ende des Zeitplans erreicht wird. In diesem Fall wird der Löschmodus am Beginn des Zeitplans fortgesetzt. Angenommen, `Index_Löschbeginn=8` und `Anzahl_zu_löschender_Sehensw=5` und es befinden sich zehn Sehenswürdigkeiten im aktuellen Zeitplan. Dann werden die Sehenswürdigkeiten an den Positionen 8, 9, 10, 1 und 2 gelöscht.

Nachdem der Löschmodus abgeschlossen ist, müssen `Shift_Netto` und `Shift_Brutto` der Sehenswürdigkeit vor dem gelöschten Set neu berechnet werden, da diese Shift-Werte von den Wegzeiten zur nächsten Sehenswürdigkeit abhängen. Dazu werden Gleichung 4.15 und Gleichung 4.16 verwendet.

Danach werden die Sehenswürdigkeiten nach vorne gereiht. Ist der Löschmodus nicht über das Ende des Zeitplans hinausgegangen, werden alle Sehenswürdigkeiten nach den gelöschten nach vorne geschoben, sodass die Lücke, die die gelöschten Sehenswürdigkeiten hinterlassen haben, wieder geschlossen ist. Ist der Löschmodus über das Ende des Zeitplans hinausgegangen, werden alle Sehenswürdigkeiten nach vorne gereiht. Kann eine Sehenswürdigkeit nicht nach vorne geschoben werden, weil sie aufgrund ihres erlaubten Zeitfensters nicht früher besichtigt werden kann (zum Beispiel wenn sie nur am Freitag besichtigt werden darf, kann sie nicht auf Mittwoch oder Donnerstag vorverlegt werden), verbleibt sie genau am selben Tag zur selben Zeit. In diesem Fall werden auch die nachfolgenden Sehenswürdigkeiten nicht verschoben.

Nachdem die Sehenswürdigkeiten verschoben wurden, müssen auch deren Werte upgedatet werden. Sind die ersten Sehenswürdigkeiten nicht vom Löschmodus betroffen, müssen alle Sehenswürdigkeiten nach dem entfernten Set aktualisiert werden, ansonsten alle. Dabei werden von den Sehenswürdigkeiten von der niedrigsten Position bis zu der an der letzten Position jeweils die Werte `Starttag`, `Endtag`, `Ankunftszeit`, `Wait_Netto`, `Wait_Brutto`, `Besichtigungsstart`, `Besichtigungsende`, `MaxShift`, `Shift_Netto` und `Shift_Brutto` upgedatet. Sind dabei bei einer Sehenswürdigkeit `Shift_Netto=0` und `Shift_Brutto=0` oder ändern sich keine Werte, müssen auch die Werte der nachfolgenden Sehenswürdigkeiten nicht mehr aktualisiert werden. Dazu werden die Formeln aus Gleichung 4.22 bis Gleichung 4.29 verwendet.

Zum Schluss müssen noch die `MaxShift`-Werte der Sehenswürdigkeiten, die vor dem gelöschten Set sind (falls es welche gibt) upgedatet werden, solange, bis entweder auch die allererste Sehenswürdigkeit geändert wurde oder der `MaxShift`-Wert einer Sehenswürdigkeit gleich bleibt. Für dieses Update wird Gleichung 4.20 verwendet, die bereits für das Einfügen einer neuen Sehenswürdigkeit bzw. für das Update der vor der eingefügten Sehenswürdigkeit befindlichen Sehenswürdigkeiten aus dem Einfügemodus verwendet wurde. Ein stark vereinfachter Pseudo-Code des Mischprozesses ist in Quellcode 4.5 gegeben.

```
1 Wenn Anzahl_zu_löschender_Sehensw ≥ Anzahl_Sehensw_im_Zeitplan:
2 {
3     Lösche alle Sehenswürdigkeiten
4 }
5 Ansonsten:
6 {
7     Lösche Anzahl_zu_löschender_Sehensw viele Sehenswürdigkeiten von
8     Index_Löschbeginn an
9     Wenn Ende des Zeitplans erreicht und noch nicht genug gelöscht:
10    {
11        Lösche von Beginn des Zeitplans an weiter
12    }
13 }
14 Berechne Shift_Netto und Shift_Brutto der Sehensw vor dem gelöschten Set
15 Für jede nachfolgende Sehenswürdigkeit:
```

```

15 {
16   Schiebe Sehenswürdigkeit nach vorne
17   Update Starttag, Endtag, Ankunftszeit, Wait, Besichtigungsstart,
      Besichtigungsende, MaxShift, Shift
18 }
19 Für jede vorhergehende Sehenswürdigkeit:
20 {
21   Update MaxShift
22 }

```

**Src. 4.5:** Stark vereinfachter Pseudo-Code des Mischprozesses des Zeitplans zur Übersicht, wie er grundsätzlich funktioniert.

Wie bereits beschrieben, werden iterativ Einfüge- und Mischprozess so lange ausgeführt, bis in 300 Iterationen hintereinander keine bessere Lösung mehr gefunden wird. Ein Beispiel, wie die interne Repräsentation des Zeitplan-Arrays aussehen könnte, ist in Quellcode 4.6 gegeben. In diesem Beispiel dürfen Besichtigungen täglich zwischen 10 und 15 Uhr stattfinden. Mit der Variable `part_drive` wird angezeigt, dass es sich bei diesem Eintrag lediglich um eine Teilanfahrt zur nächsten Sehenswürdigkeit handelt und dieser keine Besichtigung darstellt. Das Ende `end` ist fix auf 18:00 Uhr gesetzt, es zeigt also nur an, dass bis 18 Uhr gefahren werden darf, es heißt nicht, dass auch wirklich bis 18 Uhr gefahren wird. Wie lange man wirklich fährt, ist bei der Ankunftszeit `arrive` der ersten Sehenswürdigkeit des nachfolgenden Tages, die die selbe ID haben muss wie die Teilreise am Ende des Vortages, indirekt abzulesen. In diesem Beispiel beginnt die Anfahrt zu Sehenswürdigkeit 161 bereits am ersten Tag (2015-10-11). Am folgenden Tag findet man die Ankunftszeit `arrive => 16:12` bei dieser Sehenswürdigkeit. Nach einer Anfahrt über Nacht wird also die theoretische Ankunftszeit, ohne Berücksichtigung des richtigen Tages und ohne Berücksichtigung der Nachtruhe angezeigt, da diese Information den Benutzern niemals gezeigt wird, sie dient lediglich der internen Repräsentation. Theoretisch kommt man in diesem Beispiel also um 16:12 Uhr des Vortages an. Daher ergibt sich auch der Besichtigungsstart um 10:00 Uhr. Diese Startzeit könnte dem Benutzer auch angezeigt werden, falls man die Applikation dahin verfeinern möchte. Auch am 12.10. ergibt sich eine Teilreisezeit zu Sehenswürdigkeit 150. Hier ist die theoretische Ankunftszeit um 17:08 Uhr, die sich auf den 12.10. bezieht. Die Warte- und Shift-Zeiten werden in Sekunden angegeben.

```

1 Array
2 (
3   [2015-10-11] => Array
4     (
5       [0] => Array
6         (
7           [sightId] => 167
8           [arrive] => 09:06
9           [wait_netto] => 3240
10          [wait_brutto] => 3240
11          [start] => 10:00
12          [end] => 10:30
13          [maxShift] => 1860
14          [shift_netto] => 8999
15          [shift_brutto] => 8999
16        )
17      [1] => Array
18        (
19          [sightId] => 156
20          [arrive] => 13:29
21          [wait_netto] => 0
22          [wait_brutto] => 0
23          [start] => 13:29
24          [end] => 14:29
25        )

```

#### 4. Technische Beschreibung der Applikation

```
26         [maxShift] => 1860
27         [shift_netto] => 8999
28         [shift_brutto] => 8999
29     )
30
31     [2] => Array
32     (
33         [sightId] => 161
34         [part_drive] => 1
35         [end] => 18:00
36     )
37
38 )
39
40 [2015-10-12] => Array
41 (
42     [0] => Array
43     (
44         [sightId] => 161
45         [arrive] => 16:12
46         [wait_netto] => 13680
47         [wait_brutto] => 64080
48         [start] => 10:00
49         [end] => 12:00
50         [maxShift] => 424
51         [shift_netto] => 32061
52         [shift_brutto] => 82461
53     )
54
55     [1] => Array
56     (
57         [sightId] => 155
58         [arrive] => 12:38
59         [wait_netto] => 0
60         [wait_brutto] => 0
61         [start] => 12:38
62         [end] => 13:08
63         [maxShift] => 424
64         [shift_netto] => 2756
65         [shift_brutto] => 2756
66     )
67
68     [2] => Array
69     (
70         [sightId] => 168
71         [arrive] => 14:22
72         [wait_netto] => 0
73         [wait_brutto] => 0
74         [start] => 14:22
75         [end] => 14:52
76         [maxShift] => 424
77         [shift_netto] => 2756
78         [shift_brutto] => 2756
79     )
80
81     [3] => Array
82     (
83         [sightId] => 150
84         [part_drive] => 1
85         [end] => 18:00
86     )
87
88 )
89
```

```
90 [2015-10-13] => Array
91 (
92     [0] => Array
93     (
94         [sightId] => 150
95         [arrive] => 17:08
96         [wait_netto] => 10264
97         [wait_brutto] => 60664
98         [start] => 10:00
99         [end] => 10:30
100        [maxShift] => 16200
101        [shift_netto] => 0
102        [shift_brutto] => 0
103    )
104 )
105 )
106 )
107 )
```

**Src. 4.6:** Beispiel eines Zeitplans, wie er intern in PHP abgebildet wird.



# Kapitel 5

## Evaluierung der Applikation

Die Benutzerfreundlichkeit der Applikation “Holiday Recommender” wurde im Zuge dieser Arbeit getestet und ausgewertet. In diesem Kapitel werden der Testaufbau, die Durchführung sowie die Ergebnisse präsentiert.

### 5.1 Aufbau des Tests

Um die Benutzerfreundlichkeit des “Holiday Recommenders” zu testen, wurden sieben Gruppen zu je drei Personen gebeten, das System für eine gemeinsame Urlaubsplanung zu verwenden. Da es sich um ein Gruppen-Empfehlungssystem handelt, ist es sinnvoll, die Testpersonen zu Gruppen zusammenzufassen und sie das System in einer Gruppe benutzen zu lassen. Dabei sitzen die Personen aber immer alleine vor dem PC und arbeiten mit der Applikation, wie es auch in der Realität wäre. Es wurde eine Gruppengröße von drei Personen gewählt, da weniger Personen keine Gruppendynamik erzeugen würde und mehr Personen zum einen nicht mehr sinnvoll arbeiten könnten und zum anderen der Zeit- und Organisierungsaufwand zu groß wäre. Da die Einschränkung von Google existiert, dass nicht mehr als acht Sehenswürdigkeiten in den Strecken-Karten der Route angezeigt werden können [Google Developers, 2013a] und die Gefahr besteht, dass MapQuest bei der Anfrage der Routen-Matrix kein Ergebnis bzw. nur ein Teilergebnis retourniert, wenn man zu viele Sehenswürdigkeiten zur Reiseliste hinzufügt, dürfen von der Testgruppe nur wenige Sehenswürdigkeiten eingebracht werden. Damit jedes Mitglied dann zwei bis drei Sehenswürdigkeiten einbringen darf, sollte die Gruppe daher nicht mehr als drei Personen beinhalten. Außerdem würde sich der Zeitaufwand des Tests mit der Anzahl der Gruppenmitglieder in die Länge ziehen, was es schwieriger macht, Testpersonen, die dazu Zeit haben, zu finden.

Es wurden Testpersonen unterschiedlichen Alters und mit unterschiedlichen Vorkenntnissen gesucht. Dadurch werden reale Benutzerverhältnisse geschaffen, da die Applikation sowohl junge als auch ältere Personen zur Zielgruppe hat. In erster Linie wird sie eher von Personen benutzt werden, die sich im Internet und mit Facebook gut auskennen. Da innerhalb der Reisegruppe dann aber auch Personen sein könnten, die davor noch kein Facebook benutzt haben, darf man auch auf diese nicht vergessen. Ebenfalls wurden sowohl Frauen als auch Männer in den Prozess mit einbezogen.

Am Test teilgenommen haben 21 Personen im Alter von 24 bis 59 Jahren. Das Durchschnittsalter der Testpersonen lag bei 35,86 Jahren. Davon waren sieben weibliche und 14 männliche Personen. Mit Facebook kannten sich neun Personen gut aus, zwölf gaben an, sich in Facebook nicht gut oder gar nicht auszukennen. Mit ähnlichen Empfehlungssystemen hatten bereits acht Personen zu tun, 13 hatten noch keine Erfahrungen mit ähnlichen Empfehlungssystemen.

Die Tests wurden teilweise auf einem Stand-PC unter Windows und teilweise auf einem Laptop unter Unix ausgeführt. Der Laptop ist mit einem Intel Pentium Dual Core K8-Prozessor, 1.8 GHz, 4 GB RAM, ausgestattet, auf dem ein FreeBSD 10.1-RELEASE amd64 läuft. Weiters wurden Apache 2.4.12, PHP 5.6, MySQL 5.6.23 und Firefox 35.0.1 verwendet. Der Stand-PC ist mit einem AMD

## 5. Evaluierung der Applikation

Athlon 64 X2 5200+ Dual Core Prozessor, 2.7 GHz, 2 GB Ram, ausgestattet, auf dem ein Windows 7, 64 Bit läuft. Weiters wurden XAMPP 3.2.1, Apache 2.4.9, PHP 5.5, MySQL 5.6.16 und Firefox 36.0.1 verwendet.

### 5.2 Durchführung des Tests

Zu jedem Testtermin waren drei Personen, die gemeinsam eine Gruppe bildeten, anwesend. Nach der Begrüßung wurde den Testpersonen eine kurze mündliche Einführung gegeben. Zuerst wurde der Gruppe erklärt, dass sie heuer gemeinsam für drei Tage auf Urlaub fahren möchten. Die Reisedauer wurde auf drei Tage beschränkt, da nur eine beschränkte Anzahl von Sehenswürdigkeiten eingebracht werden darf aufgrund der Beschränkungen von Google und MapQuest. Damit dann nach der Berechnung der drei Routen auch drei unterschiedliche Routen präsentiert werden können, darf die Anzahl an Urlaubstagen nicht zu groß sein, da sich sonst alle Sehenswürdigkeiten im Zeitplan ausgeben würden. Die Gruppe musste sich dann auf ein Reiseziel sowie den Reisezeitraum einigen. Nachdem sich die Gruppenmitglieder einig geworden sind, wurde ihnen erklärt, dass sie sich dazu entschlossen haben, die Facebook-Applikation "Holiday Recommender" zu verwenden, um den Urlaub zu planen. Danach wurde der Gruppe eine mündliche Kurzinformation über die Applikation gegeben, die als Applikationsbeschreibung des "Holiday Recommenders" auf Facebook im "App Center" von Facebook sichtbar wäre. Diese Applikationsbeschreibung wird in Abbildung 5.1 gezeigt.

Der Holiday Recommender hilft dir und deinen Freunden mit Hilfe von Empfehlungssystemen eine Urlaubsreise zu planen.

Er ist in 3 Phasen unterteilt:

- 1) Sehenswürdigkeiten einbringen: zuerst könnt ihr alle Sehenswürdigkeiten einbringen, die euch interessieren. Du kannst dir dabei auch Sehenswürdigkeiten vorschlagen lassen, die auf deine speziellen Interessen zugeschnitten sind.
- 2) Nachdem ihr Sehenswürdigkeiten eingebracht habt und die Bewertungsphase begonnen hat, könnt ihr alle Sehenswürdigkeiten, die ihr eingebracht habt, bewerten.
- 3) Nach dem Ablauf der Bewertungsphase werden für euch 3 Routen berechnet, die sich in der von euch vorgegebenen Zeit ausgeben, wobei ihr euch dann gemeinsam für eine Route davon entscheidet.

Zuerst legt einer von euch eine neue Reise an. Ihr könnt dabei einen Städteurlaub machen, indem ihr als Start- und Endort eurer Route den selben Ort, an dem euer Urlaub stattfinden soll einträgt oder auch eine Streckenreise machen, indem ihr als Startort eurer Route den Ort eingibt, an dem die Route mit den Sehenswürdigkeiten starten soll und als Endort den Ort, an dem eure Reise, also die Route, endet. Ladet dann eure Freunde, die euch auf der Reise begleiten, zur Reise ein.

**Abbildung 5.1:** Eine Kurzinformation über den "Holiday Recommender", die als Applikationsbeschreibung auf Facebook im "App Center" von Facebook sichtbar wäre. Diese Information wurde den Testpersonen vor dem Testbeginn mitgeteilt.

Nach dieser Kurzbeschreibung des Systems wurde ein freiwilliges Gruppenmitglied gebeten, den Anfang zu machen und die Reise anzulegen sowie zu initialisieren. Sobald dieses auch eine maximale Anzahl von drei Sehenswürdigkeiten eingebracht hat und signalisierte, fertig zu sein, wurden die beiden anderen Reisemitglieder händisch zur Datenbank hinzugefügt, da die Einladung von Freunden zur Reise in der Applikation noch nicht implementiert ist. Für die Testkandidaten wurden drei Test-Accounts auf Facebook angelegt, von denen je Gruppe ein Mitglied jeweils einen Test-Account zugewiesen bekam.

Während eine Person am Computer arbeitete, durften die anderen nicht zusehen, damit sie nicht von der am Computer sitzenden Person lernen. Auch wurde nur dann Hilfe geleistet, wenn die Person ausdrücklich darum gebeten hat und es notwendig war. Waren alle drei Gruppenmitglieder einmal an der Reihe und haben die Vorschlagsphase erfolgreich absolviert, wurde händisch in die Bewertungsphase und nach dieser händisch die nächste Phase vorgeschaltet, da es im Rahmen des Tests nicht möglich gewesen wäre, die Tage, die man wirklich gebraucht hätte, bis das jeweilige, eingestellte Phasen-Enddatum überschritten worden wäre, abzuwarten.

Nachdem sich die Gruppe am Ende auf eine Route geeinigt hatte und der Computertest damit beendet war, bekamen die Testpersonen einen Fragebogen zur Evaluierung des "Holiday Recommenders" vorgelegt.

### 5.3 Feedback-Fragebogen

Nach der Durchführung des Tests wurde jeder einzelnen Testperson ein Fragebogen überreicht. Dieser Fragebogen, der sich "System Usability Scale" (SUS) nennt und von Brooke [1996] entworfen wurde, enthält zehn Fragen bezüglich der Benutzerfreundlichkeit des Systems. Diese zehn Fragen werden als Aussagesätze formuliert, wobei die Testperson den Grad der Übereinstimmung mit dem jeweiligen Satz in einer Fünf-Punkte-Skala angibt. Die fünf Punkte der Skala reichen von "Keine Übereinstimmung" bis "Starke Übereinstimmung". Der exakte Fragebogen kann in Abbildung 5.2 eingesehen werden.

Dieser Fragebogen ermöglicht es, schnell einen Überblick darüber zu bekommen, wie die Menschen die Benutzerfreundlichkeit des Systems empfinden. Er hat sich über die Jahre als sehr einfaches und vertrauenswürdige Werkzeug bewährt, um die Benutzerfreundlichkeit von Systemen zu testen. SUS wurde mittlerweile sogar in kommerzielle Evaluierungs-Toolkits wie "Morae" eingebaut und mehrere Publikationen weisen es bereits als "Industrie Standard" aus. [Brooke, 2013]

Zum originalen Fragebogen wurden zusätzlich am Ende vier Fragen zu statistischen Zwecken hinzugefügt.

## 5. Evaluierung der Applikation

### Benutzerfreundlichkeits-Skala des Systems

System Usability Scale  
© Digital Equipment Corporation, 1986.

	Keine Übereinstimmung				Starke Übereinstimmung
1. Ich glaube, dass ich dieses System öfter verwenden würde	<input type="checkbox"/>				
	1	2	3	4	5
2. Ich fand das System unnötig kompliziert	<input type="checkbox"/>				
	1	2	3	4	5
3. Ich fand, dass das System einfach zu benutzen war	<input type="checkbox"/>				
	1	2	3	4	5
4. Ich glaube, ich brauche Hilfe eines Technikers um dieses System benutzen zu können	<input type="checkbox"/>				
	1	2	3	4	5
5. Ich finde, die einzelnen Funktionen des Systems sind gut integriert	<input type="checkbox"/>				
	1	2	3	4	5
6. Ich finde, es waren zu viele Inkonsistenzen in diesem System	<input type="checkbox"/>				
	1	2	3	4	5
7. Ich glaube, dass die meisten Menschen schnell lernen, dieses System zu benutzen	<input type="checkbox"/>				
	1	2	3	4	5
8. Ich finde, dass das System sehr umständlich zu benutzen ist	<input type="checkbox"/>				
	1	2	3	4	5
9. Ich fühlte mich sehr sicher mit dem Umgang des Systems	<input type="checkbox"/>				
	1	2	3	4	5
10. Ich musste einiges lernen, bevor ich mit dem System zurecht kam	<input type="checkbox"/>				
	1	2	3	4	5

Alter: \_\_\_\_\_

Ich bin:  weiblich  männlich

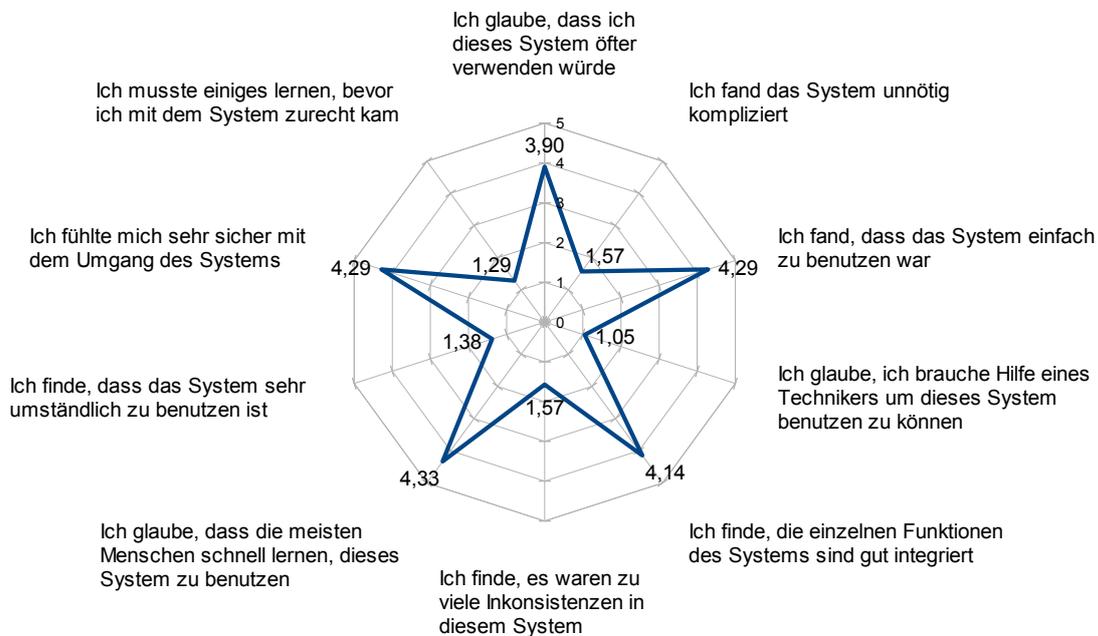
Ich kenne mich in facebook gut aus:  ja  nein

Ich habe Erfahrungen mit ähnlichen Empfehlungssystemen:  ja  nein

**Abbildung 5.2:** Der SUS Fragebogen, der von Brooke [1996] entworfen und um vier Zusatzen-  
gaben erweitert wurde, den die Testpersonen nach der Verwendung des "Holiday  
Recommenders" zur Evaluierung des Systems bekommen haben.

## 5.4 Ergebnis des Tests

Im Durchschnitt dauerte ein Testdurchlauf mit drei Personen insgesamt 1,5 Stunden. Neben der Tätigkeiten der Testpersonen fällt in diese Zeit auch das händische Hinzufügen der Testaccounts der zweiten und dritten Testpersonen zur Reise sowie das Ändern der Namen der Testpersonen in der Datenbank. Grundlegend kamen alle Testpersonen recht schnell mit dem System zurecht. Aufgefallen ist, dass manche Personen anfangs vergaßen, ihre Daten zu speichern und ihre eingegebenen Daten damit verloren gingen. Viele Kandidaten wünschten sich die Implementierung der Schnellbewertung der Sehenswürdigkeiten direkt in der Reiseliste während der Bewertungsphase. Vier Personen haben angemerkt, dass sie bei der Detail-Anzeige der Sehenswürdigkeiten gerne mehr Informationen über die Sehenswürdigkeit gehabt hätten. Von der Grundidee der Applikation waren durchgehend alle Testkandidaten begeistert, auch wenn einzelne Testpersonen bei der Bewertung angegeben haben, dass sie das System nicht oft benutzen würden, da sie entweder selten auf Urlaub fahren oder Facebook grundlegend nicht verwenden möchten. Die durchschnittlichen Bewertungen der einzelnen Fragen können in Abbildung 5.3 eingesehen werden.



**Abbildung 5.3:** Ergebnis der Evaluierung des "Holiday Recommenders", an der 21 Personen in sieben Dreiergruppen teilgenommen haben.



# Kapitel 6

## Konklusion

Dieses Kapitel bietet eine Zusammenfassung dessen, was im Rahmen dieser Arbeit erreicht wurde. Des Weiteren wird im Folgenden auf Einschränkungen der Applikation hingewiesen. Zum Schluss wird noch eine Übersicht über sinnvolle Ideen zur Weiterentwicklung des “Holiday Recommenders” angeboten.

### 6.1 Ergebnis

Die meisten bisherigen Empfehlungssysteme, die dabei helfen, Urlaube zu planen, sind auf Einzelpersonen ausgelegt, das heißt, dass bei den Empfehlungen nur die Interessen einer einzigen Person berücksichtigt werden. “INTRIGUE” [Ardissono u. a., 2001] bietet zwar ein Empfehlungssystem für Gruppen an, jedoch agiert hier nur eine Person mit dem Programm und einzelne Gruppenmitglieder werden in homogene Subgruppen zusammengefasst und somit also nicht als Individuen behandelt.

Bei den bisherigen Routen-Algorithmus werden oft keine Öffnungszeiten berücksichtigt und meist ist eine Berechnung über mehrere Tage nicht möglich. Bei den Algorithmen, die für mehrere Tage ausgelegt sind, müssen jeweils Start- und Endpunkt für jeden Tag im Vorhinein genau feststehen bzw. auf eine bestimmte Region begrenzt werden.

Soweit der Autorin bekannt ist, gibt es noch keinen Urlaubsplaner auf Empfehlungsbasis speziell für Gruppen, der ein vollständiges Planungssystem darstellt, das es ermöglicht, sowohl Sehenswürdigkeiten unter Berücksichtigung der Interessen aller Gruppenmitglieder auszuwählen als auch eine Reiseroute zu berechnen, die auf die Bedürfnisse aller Mitglieder zugeschnitten ist, wobei alle Gruppenmitglieder mit dem System agieren und von der Findung der Sehenswürdigkeiten bis hin zu den Routenerstellungen individuell berücksichtigt werden. Die erste Errungenschaft dieser Arbeit ist daher die Planung und Entwicklung eines solchen Systemes.

Die zweite Errungenschaft ist die Verbesserung der Routenberechnung dahingehend, dass Start- und Endpunkt jeder Tagesteilstrecke (mit Ausnahme des Startpunktes des ersten Tages und des Endpunktes des letzten Tages) nicht im Vorhinein bekannt sein müssen, da diese vom Algorithmus selbst passend bestimmt werden, wodurch das Problem der Optimierung der Tagesteilstrecken gelöst wurde. Die Lösung dieses Problems ist deshalb von Bedeutung, weil die Reisegruppe nun nicht mehr für jeden Tag schon vor der Berechnung der Route wissen muss, wo sie übernachten möchte, wodurch in weiterer Folge die Berechnung der Gesamtstrecke einer mehrtägigen Reise optimiert werden kann und keine Einschränkungen mehr bezüglich der Wahl der Tagesteilstrecken existieren.

Zum besseren Verständnis stelle man sich eine Reise vor, bei der die Gruppe in Dublin (Irland) mit dem Flugzeug landet, mit einem Mietauto mehrere Tage lang bis Shannon (Irland) fährt und von dort aus wieder mit dem Flugzeug die Heimreise antritt. Für die Berechnung der Routen durch den “Holiday Recommender” müssen dabei nur zwei Orte angegeben werden, nämlich der Startort (Dublin) und der Endort (Shannon). Da die Gruppe vor der Routenberechnung noch nicht weiß, welche von ihnen vorausgewählten Sehenswürdigkeiten tatsächlich in die Reiseroute aufgenommen werden und welche nicht,

## 6. Konklusion

wissen sie im Vorhinein auch nicht, wie lange sie an einem Ort verweilen werden, da das wiederum davon abhängt, wie viele Sehenswürdigkeiten bei diesem Ort in die Reiseroute aufgenommen werden. Werden zum Beispiel viele Sehenswürdigkeiten aufgrund der Bewertungen und passenden Öffnungszeiten in Dublin aufgenommen, wird die Gruppe dort eventuell zwei Tage verweilen, werden nur wenige aufgenommen, wird die Gruppe vielleicht schon früher weiterreisen. Noch extremer wäre es, wenn die Gruppe für eine bestimmte Sehenswürdigkeit einen größeren Umweg machen müsste. Sie könnte sich nämlich dazu entscheiden, gleich in der Nähe der abseits gelegenen Sehenswürdigkeit zu übernachten. Da die Gruppe aber im Vorhinein nicht weiß, ob diese abgelegene Sehenswürdigkeit überhaupt in der Reiseroute inkludiert sein wird, wäre es fatal, wenn sie vorher angeben müsste, ob sie in dieser Gegend übernachten wird. Würde die Gruppe eine Nächtigung dort im Vorhinein wählen, würde diese abgelegene Sehenswürdigkeit sogar deshalb vielleicht in die Reiseroute aufgenommen werden, obwohl sie in einer optimierten Route anderenfalls gar nicht aufgenommen worden wäre, was nicht nur eine weniger optimierte Route zur Folge hätte, sondern auch eine bewusste Beeinflussung der Route durch ein Gruppenmitglied hervorrufen könnte. Möchte zum Beispiel nur ein Mitglied eine abseits gelegene Sehenswürdigkeit besuchen und die andern nicht, könnte dieses Mitglied das System zu seinem Vorteil ausnutzen, indem es als Endpunkt eines Tages den Ort bei dieser Sehenswürdigkeit auswählt. Die sich daraus ergebende Problematik, sich eine faire Strategie überlegen zu müssen, wie Start- und Endpunkt jeder Tagesteilstrecke in der Gruppe fair gewählt werden können, fällt somit beim "Holiday Recommender" ebenfalls weg.

Zusammenfassend kann man sagen, dass die Angabe von Start- und Endpunkt jeder Tagesteilstrecke in den meisten Fällen eine weniger optimale Gesamtroute ergeben würde, da sie durch diese Angaben in ihrer Flexibilität stark eingeschränkt wäre. Es würden möglicherweise interessante Sehenswürdigkeiten durch diese Einschränkung wegfallen und dafür stattdessen eventuell weniger gute in die Route aufgenommen werden. Aufgrund der optimierten, flexiblen, automatischen Wahl von Start- und Endpunkt jeder Tagesteilstrecke durch den Algorithmus des "Holiday Recommenders" wird dies verhindert. Ebenfalls wird eine bewusste Manipulation der Route durch ein Gruppenmitglied zum Nachteil der anderen Mitglieder verhindert.

Mit der Integration des "Holiday Recommenders" in das soziale Netzwerk Facebook trifft die Applikation den Puls der Zeit, da bereits viele Menschen Gruppenevents über Facebook organisieren und eine Zusammenarbeit zwischen den einzelnen Gruppenmitgliedern durch die Umgebung von Facebook zusätzlich erleichtert wird. Des Weiteren bietet Facebook viele zusätzliche Informationen über seine Mitglieder, die zur Verbesserung und Erweiterung der integrierten Empfehlungssysteme des "Holiday Recommenders" herangezogen werden könnten.

Mit dem "Holiday Recommender" wurde also eine Applikation geschaffen, die folgendes ermöglicht:

- Urlaubsplanung für eine Gruppe von Menschen
- Urlaubsreisen weltweit ohne Einschränkung auf ein bestimmtes Gebiet
- Streckenreisen sowie Urlaube an einem einzigen Ort
- Individuelle Vorschläge von Sehenswürdigkeiten für jedes Gruppenmitglied
- Freie Entscheidung, welche Sehenswürdigkeiten als mögliche Kandidaten für die Reise aufgenommen werden sollen
- Händische Angabe von eigenen Sehenswürdigkeiten als mögliche Kandidaten für die Reise
- Bewertung jeder aufgenommenen Sehenswürdigkeit durch jedes Gruppenmitglied
- Berechnung von drei Routen mittels verschiedener Entscheidungsstrategien
- Berücksichtigung von Öffnungszeiten bei der Routenberechnung

- Berücksichtigung der gewünschten Aufenthaltsdauer bei jeder Sehenswürdigkeit
- Berücksichtigung eines gewünschten täglichen Besichtigungszeitrahmens
- Flexible, automatische Wahl der Tagesteilstrecken durch den Algorithmus bei mehrtägigen Streckenreisen
- Auswahl einer von drei Routen, je nachdem, welche Entscheidungsstrategie bevorzugt wird
- Nutzung der Funktionen von Facebook durch die Einbettung der Applikation in dieses soziale Netzwerk
- Einfache Handhabung aufgrund einer benutzerfreundlichen Oberfläche

Im Rahmen dieser Arbeit wurden für den “Holiday Recommender” sowohl die theoretischen Grundlagen entwickelt, als auch eine real einsetzbare Grundversion des Programms implementiert. Darüber hinaus wurde seine Benutzerfreundlichkeit im Rahmen eines Tests evaluiert. Daraus geht hervor, dass den Testpersonen die grundlegende Idee gefällt, und sie das System größtenteils für benutzerfreundlich halten, auch wenn Verbesserungsvorschläge suggeriert wurden und ein Ausbau des Systems sowie eine Ergänzung um zusätzliche Funktionen gewünscht wurden.

## 6.2 Einschränkungen der Applikation

Aufgrund dessen, dass sowohl die kostenlose Version der Google Maps API, als auch die kostenlose Community Edition der MapQuest API Einschränkungen unterliegen und beide durch den “Holiday Recommender” genutzt werden, ergeben sich auch Einschränkungen bei der Verwendung der Applikation.

Folgende Einschränkungen von Google bzw. MapQuest betreffen den “Holiday Recommender”:

- “Die Verwendung des Google Geocoding APIs unterliegt einer Anfragebegrenzung von 2500 Geolokalisierungsanfragen pro Tag” [Google Developers, 2013b]. Dies betrifft Anfragen, die der Erlangung geografischer Koordinaten des Start- und Endortes einer Urlaubsrouten sowie aller Sehenswürdigkeiten, die zu einer Reiseliste hinzugefügt werden, dienen.
- Die Verwendung der Google Places API erlaubt maximal 1000 Anfragen innerhalb von 24 Stunden [Google Developers, 2015]. Dies beinhaltet Autocomplete-Anfragen, die abgesendet werden, wenn ein Benutzer einen Start- oder Endort einer Urlaubsrouten eintippt bzw. eine eigene Sehenswürdigkeit eintippt. Ebenfalls betroffen ist die Suche nach passenden Sehenswürdigkeiten. Hierfür werden pro Klick auf den “Sehenswürdigkeiten vorschlagen”-Button einige Anfragen gesendet: für den Start- und Endort, wenn diese verschieden sind, sowie für jede Kategorie bzw. teilweise auch Unterkategorien. Will sich ein Benutzer Details zu einer Sehenswürdigkeit anschauen, wird eine Detailanfrage zur Sehenswürdigkeit abgeschickt, die ebenfalls zu dieser Beschränkung zählt.
- Bei der Suche nach Sehenswürdigkeiten in der Umgebung des Start- bzw. Endortes ist nur ein maximaler Suchradius von 50 km möglich [Google, 2014].
- Auf der Google-Karte, auf der die Reiseroute angezeigt wird, können nicht mehr als acht Sehenswürdigkeiten eingezeichnet werden, da Google nicht mehr als acht Wegpunkte erlaubt [Google Developers, 2013a].
- Bei der Anfrage zur Erlangung der Distanzen aller Punkte zu einander (Routen-Matrix) können maximal 25 Orte angegeben werden [MapQuest API Route Matrix, 2014]. Das heißt, neben Start- und Endort können bis zu 23 Sehenswürdigkeiten übergeben werden.

## 6. Konklusion

- Bei der Anfrage zur Erlangung der Distanzen aller Punkte zueinander darf eine Gesamtdistanz von 8000 Meilen (=12.872 Kilometer) nicht überschritten werden [MapQuest API Locations, 2014]. Diese errechnet sich aus der Addition der Distanz des ersten Punktes zum zweiten zur Distanz des zweiten Punktes zum dritten und so weiter.

Zusätzlich zu diesen offiziellen Einschränkungen treten noch weitere Probleme in der Zusammenarbeit mit Google und MapQuest auf. Bei der Suche nach Sehenswürdigkeiten in einem Radius von 50 Kilometern um einen gegebenen Punkt herum liefert Google manchmal keine Ergebnisse zurück, obwohl es welche gäbe. Schränkt man den Suchradius dann zum Beispiel auf 20 Kilometer ein, werden sehr wohl Ergebnisse zurückgeliefert. Hierbei dürfte es sich um einen Fehler in der Google API handeln. Schickt man zu viele solcher Suchanfragen in zu kurzer Zeit ab, werden dann ebenfalls oft zu einigen Anfragen keine Ergebnisse retourniert. Daher wurde im “Holiday Recommender” eine Wartezeit zwischen den einzelnen Anfragen implementiert, was die Wartezeit für den Benutzer bis zum Erhalt der Liste der Sehenswürdigkeiten erheblich verlängert.

Wenn man über MapQuest Routen-Matrizen anfordert, kommt es häufig vor, dass man in der Matrix eine oder mehrere Null-Zeilen und -Spalten erhält, also keine Distanzen. Diese Gefahr erhöht sich mit der Anzahl der Sehenswürdigkeiten auf der Reiseliste, obwohl die erlaubten 8000 Meilen noch nicht überschritten wurden. Dieses Phänomen trat zu Beginn der Entwicklungszeit des “Holiday Recommenders” häufig auf, besserte sich dann deutlich, trat aber gegen Ende der Entwicklungszeit wieder vermehrt auf. Selbst die Beispiele auf der Webseite der MapQuest API liefern oft eine Null-Zeile zurück. Versucht man es zu einem späteren Zeitpunkt wieder, tritt sie nicht mehr auf. Hierbei dürfte es sich also um einen Fehler in der MapQuest API handeln.

### 6.3 Weiterentwicklung für die Zukunft

Der “Holiday Recommender” wurde im Rahmen dieser Arbeit aufgrund des großen Umfangs und dem damit verbundenen Zeitaufwand nur in seinen Grundfunktionen implementiert. Daher wurden ein paar kleinere Details, sowie einige nette Erweiterungen, nicht mehr hinzugefügt. Damit der “Holiday Recommender” eigenständig auf Facebook als Applikation laufen kann, müssen noch folgende Dinge implementiert werden:

- Verlinkung des Administrator-Profiles einer Reise auf der Startseite
- Freunde zur Reise hinzufügen
- Anzeigen der Freunde, die Mitglieder der Reise sind
- Sehenswürdigkeit auf der Pinwand posten

Die Applikation hat großes Potenzial und bietet somit auch viele Möglichkeiten zur Weiterentwicklung. Gute Ideen für den zukünftigen Ausbau wären (unter anderen):

- Warnung anzeigen, wenn man den Tab wechselt oder ein Popup schließt, ohne gespeichert zu haben, damit keine Daten verloren gehen, bzw. die Speicher-Buttons durch eine automatische Speicherung ersetzen und einen Rückgängig-Button einbauen, damit fälschlich getätigte Änderungen wieder zurückgenommen werden können.
- Die Sterne-Schnell-Bewertung direkt in der Reiseliste implementieren.
- Reiserouten anderen “Holiday Recommender”-Benutzern zur Verfügung stellen, mit entsprechenden Anmerkungen jener Gruppe, die diese Tour gemacht hat.
- Mehr Details zu den einzelnen Sehenswürdigkeiten auf der Details-Seite anbieten.

- Vorschlägen von Hotels, die sich auf der Route am Ende jedes Tages befinden.
- Miteinplanen einer Mittagspause.
- Individuelle, genaue Öffnungszeiten pro Sehenswürdigkeit ermöglichen. Dies ist vom Routenalgorithmus her prinzipiell schon möglich.
- Anzeigen der vom Algorithmus vorgeschlagenen Besichtigungszeiten jeder Sehenswürdigkeit in der Routenansicht. Der Routenalgorithmus berechnet diese bereits, jedoch werden sie derzeit nicht angezeigt, da sie in der Praxis meist nicht genau einzuhalten sind.
- Einstellen von Routenoptionen wie: Fahrzeugart (Auto, Zug, Rad, usw.), welche Routen man bevorzugt (schnellste Route, schönste Route, kürzester Weg, usw.), ob gewisse Straßen vermieden werden sollen (Autobahn, Mautstraße, usw.), uvm.
- Genaue Routenbeschreibung anzeigen: wann man wo abbiegen muss, wie die Straßen heißen, usw.
- Merken, welche Sehenswürdigkeiten wie oft von Personen ausgeschlossen wurden (“nicht mehr anzeigen”) bzw. ausgewählt wurden. Mit Hilfe dieser Daten könnte man dann die Sehenswürdigkeiten in Zukunft reihen. Diese Protokollierung verbraucht allerdings viel Speicher, da Sehenswürdigkeiten aus aller Welt betroffen sind. Außerdem dauert es sehr lange, bis dazu aussagekräftige Daten vorliegen.
- Modifikation von Routen, falls die Benutzer am Ende mit den Routen nicht einverstanden sind, bzw. ein Umsortieren von Sehenswürdigkeiten erlauben, wenn alle Gruppenmitglieder damit einverstanden sind.
- Auswertung von Facebook-Daten zur Verbesserung der integrierten Empfehlungssysteme.
- Ausbau der Applikation, sodass sie die Benutzeranforderungen auf Handys besser erfüllt.
- Erstellen einer Webseite, auf der der “Holiday Recommender” von Facebook entkoppelt läuft.



# Abbildungsverzeichnis

3.1	Startseite Screenshot . . . . .	33
3.2	Verwaltungs-Hauptseite Screenshot . . . . .	34
3.3	Menüpunkt Reisedaten Screenshot . . . . .	35
3.4	Kalenderfunktion Screenshot . . . . .	36
3.5	Menüpunkt Meine Interessen Screenshot . . . . .	38
3.6	Menüpunkt Sehenswürdigkeiten Phase 1 Screenshot . . . . .	39
3.7	Menüpunkt Sehenswürdigkeiten Phase 2 Screenshot . . . . .	41
3.8	Details einer Sehenswürdigkeit Screenshot . . . . .	42
3.9	Vorgeschlagene Sehenswürdigkeiten Screenshot . . . . .	45
3.10	Reiseroute Phase 3 Screenshot . . . . .	48
3.11	Reiseroute Phase 4 Screenshot . . . . .	50
3.12	Notizen Screenshot . . . . .	51
5.1	Applikationsbeschreibung des Holiday Recommenders . . . . .	80
5.2	SUS Fragebogen . . . . .	82
5.3	Evaluierungsergebnis des Holiday Recommenders . . . . .	83



# Quellcodeverzeichnis

4.1	Algorithmus um Sehenswürdigkeiten zu erhalten . . . . .	58
4.2	Algorithmus um Sehenswürdigkeiten auszuwählen und zu präsentieren . . . . .	60
4.3	Übersicht des Routenalgorithmus . . . . .	65
4.4	Übersicht des Einfügeprozesses . . . . .	70
4.5	Übersicht des Mischprozesses . . . . .	74
4.6	Beispiel eines internen Zeitplans . . . . .	75



# Literaturverzeichnis

- Almeida, Ana [2008]. “Personalized Sightseeing Tours Recommendation System”. *The 4th International Symposium on Management, Engineering and Informatics: MEI 2008* (Juni 2008). <http://www.iiis.org/cds2008/cd2008sci/MEI2008/PapersPdf/M371SD.pdf> (siehe Seite 21).
- Amer-Yahia, Sihem u. a. [2009]. “Group Recommendation: Semantics and Efficiency”. In: *Proceedings of the VLDB Endowment, Band 2, Ausgabe 1*. VLDB Endowment, 1. Aug. 2009, Seiten 754–765. doi:10.14778/1687627.1687713. <http://dl.acm.org/citation.cfm?id=1687713> (siehe Seite 11).
- Andrews, Keith [2006]. *Human Computer Interaction: Vorlesung*. 2006. <http://courses.iicm.tugraz.at/hci> (siehe Seite 14).
- Andrews, Keith [2012a]. *Information Architecture and Web Usability: Vorlesung*. 2012. <http://courses.iicm.tugraz.at/iaweb> (siehe Seite 14).
- Andrews, Keith [2012b]. *Writing a Thesis: Guidelines for Writing a Master’s Thesis in Computer Science*. Graz University of Technology, Austria. 22. Okt. 2012. <http://ftp.iicm.edu/pub/keith/thesis/> (siehe Seiten vii, ix).
- Appreciate® [2011]. *Appreciate: A Facebook Application*. 10. Okt. 2011. <https://www.facebook.com/AppreciateApps> (siehe Seite 15).
- Ardissono, Liliana u. a. [2001]. “INTRIGUE: presentation of information tourist on the Web”. *Proc. Workshop su Intelligenza Artificiale per i Beni Culturali e le Biblioteche Digitali, VII Congress AI\*IA* (2001), Seiten 66–70. [http://www.di.unito.it/~goy/papers/aiaa01\\_ws.pdf](http://www.di.unito.it/~goy/papers/aiaa01_ws.pdf) (siehe Seiten 23, 85).
- Ardissono, Liliana u. a. [2002]. “Tailoring the Recommendation of Tourist Information to Heterogeneous User Groups”. In: *Hypermedia: Openness, Structural Awareness, and Adaptivity*. Band 2266. Springer, 2002, Seiten 280–295. ISBN 978-3-540-43293-7. doi:10.1007/3-540-45844-1\_26. [http://dx.doi.org/10.1007/3-540-45844-1\\_26](http://dx.doi.org/10.1007/3-540-45844-1_26) (siehe Seite 23).
- Ardissono, Liliana u. a. [2003]. “Intrigue: Personalized Recommendation Of Tourist Attractions For Desktop And Handset Devices”. In: *Applied Artificial Intelligence*. Band 17, Ausgabe 8-9. Taylor und Francis, 2003, Seiten 687–714. doi:10.1080/713827254. <http://dx.doi.org/10.1080/713827254> (siehe Seiten 12, 23).
- Baatarjav, Enkh-Amgalan, Santi Phithakkitnukoon und Ram Dantu [2008]. “Group Recommendation System for Facebook”. In: *On the Move to Meaningful Internet Systems: OTM 2008 Workshops, Band 5333*. Springer, 2008, Seiten 211–219. ISBN 978-3-540-88874-1. doi:10.1007/978-3-540-88875-8\_41. [http://dx.doi.org/10.1007/978-3-540-88875-8\\_41](http://dx.doi.org/10.1007/978-3-540-88875-8_41) (siehe Seite 15).
- Boratto, Ludovico, Salvatore Carta und Michele Satta [2010]. “Groups Identification and Individual Recommendations in Group Recommendation Algorithms”. *Proceedings of the Workshop on the Practical Use of Recommender Systems, Algorithms and Technologies (PRSAT 2010)* (Sep. 2010),

- Seiten 27–34. [http://www.academia.edu/3441423/Groups\\_identification\\_and\\_individual\\_recommendations\\_in\\_group\\_recommendation\\_algorithms](http://www.academia.edu/3441423/Groups_identification_and_individual_recommendations_in_group_recommendation_algorithms) (siehe Seiten 3, 10, 14).
- Brooke, John [1996]. “SUS: a ‘quick and dirty’ usability scale”. In: *Usability Evaluation In Industry*. Taylor & Francis Ltd., 1996, Seiten 189–194. ISBN 9780748404605. <http://www.crcpress.com/product/isbn/9780748404605> (siehe Seiten ix, 81, 82).
- Brooke, John [2013]. “SUS: A Retrospective”. *Journal Of Usability Studies (JUS)* Band 8, Ausgabe 2 (Feb. 2013), Seiten 29–40. [https://uxpa.org/sites/default/files/JUS\\_Brooke\\_February\\_2013\\_0.pdf](https://uxpa.org/sites/default/files/JUS_Brooke_February_2013_0.pdf) (siehe Seite 81).
- Cakiroglu, Seda und Aysenur Birturk [2010]. “Suggest Me a Movie: A Multi-Client Movie Recommendation Application on Facebook”. In: *Computer and Information Sciences - Proceedings of the 25th International Symposium on Computer and Information Sciences*. Ankara, Türkei: Springer, Aug. 2010, Seiten 333–338. ISBN 978-90-481-9793-4. doi:10.1007/978-90-481-9794-1\_63. [http://link.springer.com/chapter/10.1007/978-90-481-9794-1\\_63](http://link.springer.com/chapter/10.1007/978-90-481-9794-1_63) (siehe Seite 16).
- Castillo, Luis u. a. [2008]. “SAMAP: An user-oriented adaptive system for planning tourist visits”. *Expert Systems with Applications* Band 34, Ausgabe 2 (Feb. 2008), Seiten 1318–1332. ISSN 0957-4174. doi:10.1016/j.eswa.2006.12.029. <http://www.sciencedirect.com/science/article/pii/S0957417406004209> (siehe Seiten 2, 27).
- Chaudhari, Parag, Shraddha Deshmukh und Tushar Mhaskar [2013]. *Lets Hangout! A Group Based Information Recommender System*. University of Southern California, USA, 2013. <http://www.isi.edu/~szekely/contents/cs548/2013-spring/letshangout.pdf> (siehe Seite 15).
- Cheverst, Keith u. a. [2000]. “Developing a Context-aware Electronic Tourist Guide: Some Issues and Experiences”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. Den Haag, Niederlande: ACM Press, New York, 2000, Seiten 17–24. ISBN 1-58113-216-6. doi:10.1145/332040.332047. <http://doi.acm.org/10.1145/332040.332047> (siehe Seite 2).
- CityTripPlanner [2015]. *City Trip Planner*. 7. Feb. 2015. <http://www.citytripplanner.com/> (siehe Seite 22).
- Conitzer, Vincent und Tuomas Sandholm [2002]. “Complexity of Mechanism Design”. In: *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence*. Alberta, Kanada: Morgan Kaufmann Publishers Inc., 2002, Seiten 103–110. ISBN 1-55860-897-4. <http://dl.acm.org/citation.cfm?id=2073876.2073889> (siehe Seite 24).
- Conitzer, Vincent und Tuomas Sandholm [2003]. *Applications of Automated Mechanism Design. Early version: the UAI-03 Bayesian Modeling Applications Workshop*. Acapulco, Mexiko, 2003. <https://jflap.org/~conitzer/amdapplicationsUAIworkshop03.pdf> (siehe Seite 24).
- Cook, William J. [2011]. *In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation*. 1. Auflage. Princeton University Press, 2011. ISBN 9780691152707. <http://press.princeton.edu/titles/9531.html> (siehe Seite 61).
- Croissant, Yannick [2015]. *Ajaxload*. 30. März 2015. <http://www.ajaxload.info/> (siehe Seite ix).
- Delgado, Joaquin und Richard Davidson [2002]. “Knowledge Bases and User Profiling in Travel and Hospitality Recommender Systems”. *Information & Communication Technologies in Tourism 2002: ENTER Conference Proceeding* (Jän. 2002), Seiten 1–16. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.102.5909> (siehe Seite 4).
- Dunlop, Mark D. u. a. [2004]. “Design and development of Taeneb City Guide - From Paper Maps and Guidebooks to Electronic Guides”. In: *Information and communication technologies in tourism 2004, the 11th ENTER International Conference in Cairo, Egypt*. Springer, 2004, Seiten 58–64. ISBN

- 3-211-20669-8. <http://personal.cis.strath.ac.uk/~mdd/research/publications/> (siehe Seite 2).
- Fox, Maria und Derek Long [2001]. “Hybrid STAN: Identifying and managing combinatorial optimisation sub-problems in planning”. In: *Proceedings of International Joint Conference on Artificial Intelligence*. Band 1. Morgan Kaufmann, 2001, Seiten 445–452. ISBN 1-55860-777-3. <http://strathprints.strath.ac.uk/1935/> (siehe Seite 27).
- Gartrell, Mike u. a. [2010]. “Enhancing group recommendation by incorporating social relationship interactions”. In: *GROUP '10 Proceedings of the 16th ACM international conference on Supporting group work*. Sanibel Island, Florida, USA: ACM Press, New York, Nov. 2010, Seiten 97–106. ISBN 9781450303873. doi:10.1145/1880071.1880087. <http://dl.acm.org/citation.cfm?id=1880087> (siehe Seiten 11, 12, 14, 47, 63).
- Godart, Jean-Marc [2001]. “Combinatorial Optimisation for Trip Planning”. *Belgian Journal of Operations Research, Statistics and Computer Science (JORBEL)* Band 41, Ausgabe 1-2 (2001), Seiten 59–68. <http://www.ig.fpms.ac.be/node/346> (siehe Seite 21).
- Godart, Jean-Marc [2006]. “Multicriteria Selection of Attractions and Hotels for Sightseeing Tours”. *Proceedings of ECCO 2006* (Mai 2006) (siehe Seite 21).
- Golden, Bruce L., Larry Levy und Rakesh Vohra [1987]. “The orienteering problem”. *Naval Research Logistics (NRL)* Band 34, Ausgabe 3 (Juni 1987), Seiten 307–318. ISSN 1520-6750. doi:10.1002/1520-6750(198706)34:3<307::AID-NAV3220340302>3.0.CO;2-D. [http://dx.doi.org/10.1002/1520-6750\(198706\)34:3%3C307::AID-NAV3220340302%3E3.0.CO;2-D](http://dx.doi.org/10.1002/1520-6750(198706)34:3%3C307::AID-NAV3220340302%3E3.0.CO;2-D) (siehe Seiten 5, 26).
- Google [2014]. *Google API - Nearby Search Requests*. 16. Sep. 2014. <https://developers.google.com/places/documentation/search?hl=de#PlaceSearchRequests/> (siehe Seiten 57, 87).
- Google API [2013]. *Google Maps API*. 31. Jän. 2013. <https://developers.google.com/maps/> (siehe Seiten ix, 57).
- Google Developers [2013a]. *Google Developers - Directions API Waypoints*. 31. Jän. 2013. <https://developers.google.com/maps/documentation/javascript/directions#Waypoints> (siehe Seiten 47, 79, 87).
- Google Developers [2013b]. *Google Developers Dokumentation - Geocoding*. 31. Jän. 2013. <https://developers.google.com/maps/documentation/geocoding/> (siehe Seite 87).
- Google Developers [2014]. *Google Developers Dokumentation - Distanzmatrix*. 19. Nov. 2014. <https://developers.google.com/maps/documentation/distancematrix/> (siehe Seite 64).
- Google Developers [2015]. *Google Developers - Places API Usage Limits*. 31. März 2015. <https://developers.google.com/places/webservice/usage> (siehe Seite 87).
- Hashimoto, Hideki, Mutsunori Yagiura und Toshihide Ibaraki [2008]. “An iterated local search algorithm for the time-dependent vehicle routing problem with time windows”. *Discrete Optimization* Band 5, Ausgabe 2 (Mai 2008), Seiten 434–456. ISSN 1572-5286. doi:10.1016/j.disopt.2007.05.004. <http://www.sciencedirect.com/science/article/pii/S1572528607000485> (siehe Seite 28).
- Jameson, Anthony [2004]. “More than the sum of its members: challenges for group recommender systems”. In: *AVI '04 Proceedings of the working conference on Advanced visual interfaces*. Bari, Italien: ACM Press, New York, Mai 2004, Seiten 48–54. ISBN 1581138679. doi:10.1145/989863.989869. <http://dl.acm.org/citation.cfm?id=989869> (siehe Seiten 23–25, 43).
- Jameson, Anthony und Barry Smyth [2007]. “Recommendation to Groups”. In: *The Adaptive Web*. Band 4321. Springer, 2007, Seiten 596–627. ISBN 978-3-540-72078-2. doi:10.1007/978-3-540-72079-9\_20. [http://dx.doi.org/10.1007/978-3-540-72079-9\\_20](http://dx.doi.org/10.1007/978-3-540-72079-9_20) (siehe Seiten 2, 4, 10, 23–25).

- Jannach, Dietmar u. a. [2011]. In: *Recommender Systems: An introduction*. Cambridge University Press, 2011. ISBN 978-0-521-49336-9. <http://www.recommenderbook.net/recommender-systems-introduction> (siehe Seiten 1, 9).
- jQuery Foundation [2015]. *jQuery*. 30. März 2015. <https://jquery.com/> (siehe Seite ix).
- Kinoshita, Takamasa u. a. [2006]. “A Personal Navigation System for Sightseeing across Multiple Days”. *Proc. of the 3rd International Conference on Mobile Computing and Ubiquitous Networking (ICMU)* (2006), Seiten 254–259. <http://www.icmu.org/icmu2006/pdf/ICMU2006-1568990584.pdf> (siehe Seite 29).
- Knabben, Frederico [2015]. *CKEditor*. 26. März 2015. <http://ckeditor.com/> (siehe Seiten ix, 51).
- Kramer, Ronny, Marko Modsching und Klaus Ten Hagen [2006]. “A City Guide Agent Creating and Adapting Individual Sightseeing Tours Based on Field Trial Results”. *International Journal of Computational Intelligence Research* Band 2, Ausgabe 2 (2006), Seiten 191–206. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.106.4444> (siehe Seite 2).
- Lazaruk, Szymon u. a. [2012]. “Semantic Web Recommendation Application”. In: *Proceedings of the Federated Conference on Computer Science and Information Systems, 2012*. Wrocław, Polen: PTI, IE-EE, Sep. 2012, Seiten 1055–1062. ISBN 978-83-60810-51-4. <https://fedcsis.org/proceedings/2012/> (siehe Seiten 4, 15).
- Lourenço, Helena R., Olivier C. Martin und Thomas Stützle [2003]. “Iterated Local Search”. In: *Handbook of Metaheuristics*. Band 57. Springer US, 2003, Seiten 320–353. ISBN 978-1-4020-7263-5. doi:10.1007/0-306-48056-5\_11. [http://dx.doi.org/10.1007/0-306-48056-5\\_11](http://dx.doi.org/10.1007/0-306-48056-5_11) (siehe Seite 26).
- MapQuest API [2014]. *MapQuest API*. 9. Nov. 2014. <http://developer.mapquest.com/> (siehe Seiten ix, 57).
- MapQuest API Locations [2014]. *MapQuest API - Specifying Locations*. 21. Nov. 2014. <http://open.mapquestapi.com/directions/#locations> (siehe Seiten 64, 88).
- MapQuest API Route Matrix [2014]. *MapQuest API - Route Matrix*. 21. Nov. 2014. <http://www.mapquestapi.com/directions/#matrix> (siehe Seiten 64, 87).
- Maruyama, Atsushi u. a. [2004a]. “A Personal Tourism Navigation System to Support Traveling Multiple Destinations with Time Restrictions”. *Advanced Information Networking and Applications (AINA)* Band 2 (2004), Seiten 18–21. doi:10.1109/AINA.2004.1283747. [http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=1283747](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1283747) (siehe Seite 29).
- Maruyama, Atsushi u. a. [2004b]. “P-Tour: A Personal Navigation System for Tourism”. In: *Proc. of 11th World Congress on ITS*. 2004, Seiten 18–21. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.58.9019> (siehe Seite 29).
- Masthoff, Judith [2004]. “Group Modeling: Selecting a Sequence of Television Items to Suit a Group of Viewers”. *User Modeling and User-Adapted Interaction* Band 14, Ausgabe 1 (2004), Seiten 37–85. ISSN 0924-1868. doi:10.1023/B:USER.0000010138.79319.fd. <http://dl.acm.org/citation.cfm?id=962428> (siehe Seiten 11, 13).
- Masthoff, Judith [2010]. “Group Recommender Systems: Combining Individual Models”. In: *Recommender Systems Handbook*. Springer US, 2010, Seiten 677–702. ISBN 978-0-387-85819-7. doi:10.1007/978-0-387-85820-3\_21. [http://dx.doi.org/10.1007/978-0-387-85820-3\\_21](http://dx.doi.org/10.1007/978-0-387-85820-3_21) (siehe Seite 63).
- Mittenthal, John und Charles Noon [2007]. “An Insert/Delete Heuristic for the Travelling Salesman Subset-Tour Problem with One Additional Constraint”. *Journal of the Operational Research Society*

- (*JORS*) Band 43, Ausgabe 3 (Juli 2007), Seiten 277–283. ISSN 0160-5682. doi:10.1057/jors.1992.37. <http://dx.doi.org/10.1057/jors.1992.37> (siehe Seite 5).
- Nagata, Munenobu u. a. [2006]. “A Method to Plan Group Tours with Joining and Forking”. In: *Simulated Evolution and Learning - 6th International Conference*. Springer, 2006, Seiten 881–888. ISBN 978-3-540-47332-9. doi:10.1007/11903697\_111. [http://dx.doi.org/10.1007/11903697\\_111](http://dx.doi.org/10.1007/11903697_111) (siehe Seite 29).
- Natural Digit S.A., Belgien [2015]. *YourTour*. 14. Feb. 2015. <http://www.yourtour.com/> (siehe Seite 21).
- Niaraki, Abolghasem Sadeghi und Kyehyun Kim [2009]. “Ontology based personalized route planning system using a multi-criteria decision making approach”. *Expert Systems with Applications* Band 36, Ausgabe 2, Teil 1 (2009), Seiten 2250–2259. ISSN 0957-4174. doi:10.1016/j.eswa.2007.12.053. <http://www.sciencedirect.com/science/article/pii/S0957417407006902> (siehe Seite 27).
- Openclipart [2015]. *Open Clip Art*. 30. März 2015. <http://openclipart.org/> (siehe Seite ix).
- Popescu, George [2011]. *Validation of Game Theoretic Algorithms for Group Recommender Systems*. École polytechnique fédérale de Lausanne (EPFL), Schweiz, Project Report, Juli 2011 (siehe Seite 17).
- Popescu, George [2013]. “Group Recommender Systems as a Voting Problem”. In: *Online Communities and Social Computing, Band 8029*. Springer, 2013, Seiten 412–421. ISBN 978-3-642-39370-9. doi:10.1007/978-3-642-39371-6\_46. [http://dx.doi.org/10.1007/978-3-642-39371-6\\_46](http://dx.doi.org/10.1007/978-3-642-39371-6_46) (siehe Seiten 17, 18).
- Popescu, George und Pearl Pu [2011]. *Group Recommender Systems as a Voting Problem*. École polytechnique fédérale de Lausanne (EPFL), Schweiz, Technical report, Feb. 2011 (siehe Seite 17).
- Prokop, Lukas [2014]. *Grundlagen der Informatik: Skriptum zur Vorlesung*. Technische Universität Graz, IST. 29. Okt. 2014, Seiten 65–68. <http://gdi.ist.tugraz.at/skriptum.pdf> (siehe Seite 61).
- Quijano-Sánchez, Lara, Juan A. Recio-García und Belén Díaz-Agudo [2011]. “Group recommendation methods for social network environments”. *3rd Workshop on Recommender Systems and the Social Web 5th ACM International Conference on Recommender Systems, RecSys’11 (2011)*. <http://www.dcs.warwick.ac.uk/~ssanand/RSWeb11/4Quijano-Sanchez.pdf> (siehe Seiten 3, 16).
- Righini, Giovanni und Matteo Salani [2009]. “Decremental state space relaxation strategies and initialization heuristics for solving the Orienteering Problem with Time Windows with dynamic programming”. *Computers & Operations Research* Band 36, Ausgabe 4 (Apr. 2009), Seiten 1191–1203. doi:10.1016/j.cor.2008.01.003. <http://dx.doi.org/10.1016/j.cor.2008.01.003> (siehe Seite 5).
- Rogers, Seth und Pat Langley [1998]. “Personalized Driving Route Recommendations”. In: *In Proceedings of the American Association of Arti Intelligence Workshop on Recommender Systems*. Madison, WI, 1998, Seiten 96–100. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.51.1720> (siehe Seite 26).
- Schafer, J.Ben u. a. [2007]. “Collaborative Filtering Recommender Systems”. In: *The Adaptive Web*. Band 4321. Springer, 2007, Seiten 291–324. ISBN 978-3-540-72078-2. doi:10.1007/978-3-540-72079-9\_9. [http://dx.doi.org/10.1007/978-3-540-72079-9\\_9](http://dx.doi.org/10.1007/978-3-540-72079-9_9) (siehe Seite 9).
- Sebastia, Laura u. a. [2009a]. “A Group Recommender System for Tourist Activities”. In: *E-Commerce and Web Technologies*. Band 5692. Springer, 2009, Seiten 26–37. ISBN 978-3-642-03963-8. doi:10.1007/978-3-642-03964-5\_4. [http://dx.doi.org/10.1007/978-3-642-03964-5\\_4](http://dx.doi.org/10.1007/978-3-642-03964-5_4) (siehe Seiten 19, 20).
- Sebastia, Laura u. a. [2009b]. “e-Tourism: A Tourist Recommendation And Planning Application”. *International Journal on Artificial Intelligence Tools* Band 18, Ausgabe 05 (Sep. 2009), Seiten 717–

738. doi:10.1142/S0218213009000378. <http://www.worldscientific.com/doi/abs/10.1142/S0218213009000378> (siehe Seiten 2, 19, 20).
- Shiraishi, Takayuki u. a. [2005]. “A Personal Navigation System with a Schedule Planning Facility Based on Multi-Objective Criteria”. In: *Proc. of 2nd International Conference on Mobile Computing and Ubiquitous Networking (ICMU)*. 2005, Seiten 104–109. <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.59.5613> (siehe Seite 29).
- Silver, Edward A. [2004]. “An Overview of Heuristic Solution Methods”. *Journal of the Operational Research Society* Band 55 (Mai 2004), Seiten 936–956. doi:10.1057/palgrave.jors.2601758. <http://dx.doi.org/10.1057/palgrave.jors.2601758> (siehe Seite 27).
- Soo, Von-Wun und Shu-Hau Liang [2001]. “Recommending a Trip Plan by Negotiation with a Software Travel Agent”. In: *Cooperative Information Agents V*. Band 2182. Springer, 2001, Seiten 32–37. ISBN 978-3-540-42545-8. doi:10.1007/3-540-44799-7\_5. [http://dx.doi.org/10.1007/3-540-44799-7\\_5](http://dx.doi.org/10.1007/3-540-44799-7_5) (siehe Seiten 5, 28).
- Souffriau, Wouter und Pieter Vansteenwegen [2010]. “Tourist Trip Planning Functionalities: State-of-the-Art and Future”. In: *Current Trends in Web Engineering*. Band 6385. Springer, 2010, Seiten 474–485. ISBN 978-3-642-16984-7. doi:10.1007/978-3-642-16985-4\_46. [http://dx.doi.org/10.1007/978-3-642-16985-4\\_46](http://dx.doi.org/10.1007/978-3-642-16985-4_46) (siehe Seiten 2, 21, 26).
- Stackoverflow [2015]. *Stackoverflow*. 31. März 2015. <http://stackoverflow.com/> (siehe Seite ix).
- Stettinger, Martin [2013]. “Intelligent Services for Group Decision Making”. Magisterarbeit. Technische Universität Graz, Österreich, 2013 (siehe Seite 3).
- Stettinger, Martin, Alexander Felfernig, Gerhard Leitner und Stefan Reiterer [2015]. “Counteracting Anchoring Effects in Group Decision Making”. *23rd Conference on User Modeling, Adaptation, and Personalization (UMAP 2015)* (Juni 2015), Seiten 118–130. doi:10.1007/978-3-319-20267-9. [http://www.researchgate.net/publication/275098678\\_Counteracting\\_Anchoring\\_Effects\\_in\\_Group\\_Decision\\_Making](http://www.researchgate.net/publication/275098678_Counteracting_Anchoring_Effects_in_Group_Decision_Making) (siehe Seite 24).
- Stettinger, Martin, Alexander Felfernig, Gerhard Leitner, Stefan Reiterer und Michael Jeran [2015]. “Counteracting Serial Position Effects in the CHOICLA Group Decision Support Environment”. *Proceedings of the 20th ACM Conference on Intelligent User Interfaces (IUI 2015)* (Apr. 2015), Seiten 148–157. doi:10.1145/2678025.2701391. <http://dl.acm.org/citation.cfm?id=2701391> (siehe Seite 14).
- Stream Media Pte Ltd [2012]. *Free Games Recommended Applorer: A Facebook Application*. 2012. <http://applorer.com/> (siehe Seite 15).
- Sun, Yuxian und Lyndon Lee [2004]. “Agent-based personalised tourist route advice system”. *ISPRS: 20th International Society for Photogrammetry and Remote Sensing (ISPRS) Congress, Istanbul* (2004), Seiten 319–324. <http://195.130.87.21:8080/dspace/handle/123456789/593> (siehe Seiten 5, 27).
- Symeonidis, Panagiotis, Dimitrios Ntempos und Yannis Manolopoulos [2014]. In: *Recommender Systems for Location-based Social Networks*. Springer, 2014. ISBN 978-1-4939-0285-9. doi:10.1007/978-1-4939-0286-6. <http://dx.doi.org/10.1007/978-1-4939-0286-6> (siehe Seite 1).
- Tsiligirides, Theodore [1984]. “Heuristic Methods Applied to Orienteering”. *Journal of the Operational Research Society (JORS)* Band 35, Ausgabe 9 (Sep. 1984), Seiten 797–809. doi:10.1057/jors.1984.162. <http://dx.doi.org/10.1057/jors.1984.162> (siehe Seite 26).
- Vansteenwegen, Pieter und Dirk Van Oudheusden [2007]. “The Mobile Tourist Guide: An OR Opportunity”. *OR Insight* Band 20, Ausgabe 3 (Juli 2007), Seiten 21–27. ISSN 0953-5543. doi:10.1057/ori.2007.17. <http://dx.doi.org/10.1057/ori.2007.17> (siehe Seite 5).

- Vansteenwegen, Pieter u. a. [2009]. “Iterated local search for the team orienteering problem with time windows”. In: *Computers & Operations Research, Band 36, Ausgabe 12*. Elsevier Ltd., März 2009, Seiten 3281–3290. doi:10.1016/j.cor.2009.03.008. <http://www.sciencedirect.com/science/article/pii/S030505480900080X> (siehe Seiten 22, 30, 61, 62, 68–70).
- Vansteenwegen, Pieter u. a. [2011]. “The City Trip Planner: An expert system for tourists”. *Expert Systems with Applications* Band 38, Ausgabe 6 (2011), Seiten 6540–6546. ISSN 0957-4174. doi:10.1016/j.eswa.2010.11.085. <http://dx.doi.org/10.1016/j.eswa.2010.11.085> (siehe Seiten 4, 5, 22).
- Wikipedia [2014]. *Ajax (Programmierung)*. 29. Okt. 2014. [http://de.wikipedia.org/wiki/Ajax\\_\(Programmierung\)](http://de.wikipedia.org/wiki/Ajax_(Programmierung)) (siehe Seite 57).
- Youyou, Wu, Michal Kosinski und David Stillwell [2015]. “Computer-based personality judgments are more accurate than those made by humans”. *Proceedings of the National Academy of Sciences* Band 112, Ausgabe 4 (2015), Seiten 1036–1040. doi:10.1073/pnas.1418680112. <http://www.pnas.org/content/early/2015/01/07/1418680112> (siehe Seite 16).