



Stefan Galler, BSc

## **Conversational Agent for eHealth**

# **A natural language interface in eHealth to support behavioral change**

### **MASTER'S THESIS**

to achieve the university degree of

Master of Science

Master's degree programme: Software Development and Business Management

submitted to

**Graz University of Technology**

Supervisor

Assoc.Prof. Dr. Christian Gütl

Institute for Information Systems and Computer Media (IICM)





Stefan Galler, BSc

## **Dialogorientierter Agent für eHealth**

# **Benutzeroberfläche in Natürlicher Sprache zur Unterstützung von Verhaltensänderungen in eHealth**

### **MASTERARBEIT**

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Masterstudium Softwareentwicklung – Wirtschaft

eingereicht an der

**Technischen Universität Graz**

Betreuer

Assoc.Prof. Dr. Christian Guetl

Institut für Informationssysteme und Computermedien (IICM)

Graz, August 2017



# Conversational Agent for eHealth

*A natural language interface in eHealth to support behavioral change.*

**Stefan Galler**



## **Adviser**

Christian Gütl

*Graz University of Technology*

## **Co-Adviser**

David Glance

*University of Western Australia*

**August, 2017**



# Abstract

Chronic diseases are increasingly becoming a problem in western civilization. Around half of the citizens of the European Union (EU) over 18 were considered overweight in 2014. Most of the time, a bad lifestyle is the reason for this issue. Changing a persons lifestyle is possible, but strong support is needed.

Throughout the last years, technology became increasingly important in the area of health. Especially the Internet allowed to easily reach people with personalized and interactive information. This combination of health and technology – often referred to as eHealth – is also used to support the change towards a healthier lifestyle.

A technology that has been developed for decades is conversational agents (CAs). These are computer programs that solely use natural language to communicate with the user. Although they are applied to multiple fields, CAs as part of eHealth are relatively unexplored.

In this thesis, a CA was designed, implemented and integrated to an existing eHealth platform. The aim of the CA was to proactively approach its user in order to ask for health information. This information was then further processed to generate motivational messages, which were then sent to the user.

The acceptance of this prototype was tested in a user study including seven participants. Happiness and anger were the prevailing feelings on the Computer Emotion Scale (CES). An average score of 72.1 was reached on the System Usability Scale (SUS). The number of interactions between the users and the CA decreased between week 1 and week 2 of the study; on average 11.7 (standard deviation (SD) 6.1) interaction took place in the first week, where 6.9 (SD 6.9) interactions occurred on average in week 2.





# Kurzzusammenfassung

Chronische Krankheiten werden in der westlichen Welt zunehmend zu einem Problem. Rund die Hälfte der über 18-Jährigen Einwohner der European Union (EU) litten 2014 an Übergewicht. Dies ist meist durch einen schlechten Lebensstil begründet. Es ist möglich diesen Lebensstil zu verbessern. Dies bedarf aber große Unterstützung.

In den letzten Jahren haben verschiedene Technologien im Gesundheitsbereich an Wichtigkeit gewonnen. Insbesondere das Internet ermöglicht das Zurverfügungstellen von personalisierten und interaktiven Materialien. Diese Kombination zwischen Gesundheit und Technologien (häufig auch als eHealth bezeichnet) wird auch zur Unterstützung bei der Verbesserung von Lebensgewohnheiten verwendet.

Eine Technologie, die seit vielen Jahrzehnten entwickelt wird, sind Dialogorientierte Agenten. Dies sind Computerprogramme, die darauf spezialisiert sind, mit dem Benutzer rein in natürlicher Sprache zu kommunizieren. Obwohl diese Technologie in vielen Gebieten verwendet wird, ist ihre Anwendung im Gesundheitsbereich noch relativ wenig erforscht.

In dieser Arbeit wurde ein Dialogorientierter Agent geplant, umgesetzt, und schließlich in eine vorhandene Gesundheitsplattform integriert. Die Aufgabe dieses Dialogorientierten Agenten war es, den Benutzer aktiv nach Gesundheitsdaten zu fragen, und diese in weiterer Folge für die Generierung von motivierenden Nachrichten zu verwenden.

Die Akzeptanz dieses Prototypen durch die Benutzer wurde durch eine Studie mit sieben Teilnehmern getestet. Glück und Wut waren die überwiegend vorkommenden Emotionen, die durch die Computer Emotion Scale (CES) gemessen wurden. Eine durchschnittliche Bewertung von 72.1 wurde auf der System Usability Scale (SUS) erreicht. Die Anzahl der Interaktionen zwischen Benutzer und System gingen zwischen erster und zweiter Studienwoche zurück. Durchschnittlich 11.7 (Standardabweichung 6.1) Interaktionen fanden in der ersten Woche statt, in der zweiten waren es 6.9 (Standardabweichung 6.9).



**AFFIDAVIT**

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 indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

---

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 entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

---

Datum

---

Unterschrift



# Acknowledgments

I want to thank my parents for supporting me throughout my studies no matter I was in Austria, or abroad.

My greatest appreciation to Christian Gütl, who gave me the opportunity to write my thesis in Australia. He always offered helpful feedback on my work and guided me through the process of this project.

For the support and warm welcome at the University of Western Australia, I want to thank David Glance. Furthermore, many thanks to John Pham who helped me with the technical problems concerning the Self eHealth platform.

Finally, my gratitude also belongs to Janine Wenzel for her encouragement throughout the duration of writing this thesis.





# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation and Background . . . . .	1
1.2	Outline . . . . .	3
<b>2</b>	<b>eHealth</b>	<b>5</b>
2.1	Definitions . . . . .	5
2.2	History and Development . . . . .	6
2.3	Applications . . . . .	8
2.3.1	Health Communication . . . . .	8
2.3.2	Clinical Decision Support Systems . . . . .	8
2.3.3	Electronic Health Records . . . . .	9
2.3.4	Internet-based Interventions . . . . .	9
2.4	Common Theories in eHealth Interventions . . . . .	10
2.4.1	Social Cognitive Theory . . . . .	10
2.4.2	Transtheoretical Model . . . . .	11
2.4.3	Theory of Planned Behavior . . . . .	11
2.4.4	Health Belief Model . . . . .	11
2.5	Summary . . . . .	12
<b>3</b>	<b>Conversational Agents</b>	<b>13</b>
3.1	Definition . . . . .	13
3.2	Classification . . . . .	15
3.3	The Turing Test . . . . .	18
3.4	Early Conversational Agents . . . . .	18
3.4.1	ELIZA . . . . .	18
3.4.2	PARRY . . . . .	20
3.5	The Loebner Prize . . . . .	22
3.5.1	The Beginnings . . . . .	23
3.5.2	AIML: A.L.I.C.E and Mitsuku . . . . .	23
3.5.3	ChatScript: Suzette, Rosette, and Rose . . . . .	24
3.6	Intelligent Personal Assistants . . . . .	25

3.6.1	Apple Siri . . . . .	26
3.6.2	Google Now & Goolge Assistant . . . . .	26
3.6.3	Amazon Alexa . . . . .	26
3.6.4	Microsoft Cortana . . . . .	26
3.7	Data-driven Approaches . . . . .	27
3.8	Summary . . . . .	27
<b>4</b>	<b>Conversational Agents in Health</b>	<b>29</b>
4.1	Combining eHealth and Conversational Assistants . . . . .	29
4.2	Conversational Agents as Information Sources . . . . .	30
4.3	Conversational Agents for Behavior Change . . . . .	30
4.4	Summary . . . . .	31
<b>5</b>	<b>System Architecture</b>	<b>33</b>
5.1	System Goals . . . . .	33
5.2	Analysis of the Self eHealth Platform . . . . .	33
5.2.1	Functionality . . . . .	34
5.2.2	Usage . . . . .	35
5.2.3	Technology . . . . .	37
5.3	General Requirements . . . . .	38
5.4	Proposal of Alternative Concepts . . . . .	40
5.4.1	Concept Proposal 1: Health Assistant . . . . .	40
5.4.2	Concept Proposal 2: Motivation Diagnosis . . . . .	40
5.4.3	Concept Proposal 3: Motivational Conversational Agent . . . . .	41
5.5	Prototype Concept . . . . .	41
5.6	Technology Decisions . . . . .	42
5.6.1	Conversational Agent Frameworks . . . . .	43
5.6.2	Backend and Logic . . . . .	46
5.6.3	Frontend . . . . .	48
5.6.4	Technology Used . . . . .	48
5.7	Integration and Usability . . . . .	48
5.7.1	User Interface Design . . . . .	49
5.7.2	Embodiment . . . . .	49
5.8	Architecture . . . . .	50
5.9	Summary . . . . .	50
<b>6</b>	<b>Prototype Implementation</b>	<b>53</b>
6.1	Technology Introduction . . . . .	53
6.1.1	ChatScript . . . . .	53
6.1.2	Django . . . . .	55

6.1.3	NLTK . . . . .	55
6.1.4	pandas . . . . .	56
6.1.5	User Interface . . . . .	56
6.2	Topic Selection . . . . .	57
6.3	Message Generation . . . . .	58
6.3.1	Compilation . . . . .	58
6.3.2	Control Script . . . . .	59
6.3.3	Macros . . . . .	59
6.3.4	Topic Scripts . . . . .	59
6.4	Data Interface . . . . .	61
6.5	Feedback Generation . . . . .	63
6.6	Data Models . . . . .	64
6.7	Sebo Heartbeat . . . . .	65
6.8	User Interface . . . . .	65
6.9	Conversation Design . . . . .	67
6.10	Maintainability and Administration . . . . .	67
6.11	Summary . . . . .	69
<b>7</b>	<b>User Study</b>	<b>73</b>
7.1	Originally Planned Study . . . . .	73
7.2	Study Setup . . . . .	73
7.3	Questionnaire . . . . .	74
7.3.1	Computer Emotion Scale . . . . .	74
7.3.2	System Usability Scale . . . . .	75
7.3.3	System Specific Questions . . . . .	75
7.4	Implementation . . . . .	76
7.5	Summary . . . . .	76
<b>8</b>	<b>Results and Discussion</b>	<b>77</b>
8.1	Results . . . . .	77
8.1.1	Participants . . . . .	77
8.1.2	Computer Emotion Scale . . . . .	77
8.1.3	System Usability Scale . . . . .	78
8.1.4	System Specific questions . . . . .	80
8.1.5	Other Feedback By Participants . . . . .	82
8.1.6	Behavior Analysis . . . . .	82
8.2	Discussion and Suggested Improvements . . . . .	83
8.3	Limitations . . . . .	86
8.4	Summary . . . . .	87

<b>9 Lessons Learned</b>	<b>89</b>
9.1 Literature . . . . .	89
9.2 Implementation . . . . .	89
9.3 Outcome . . . . .	90
<b>10 Conclusion and Future Work</b>	<b>91</b>
10.1 Conclusion . . . . .	91
10.2 Future Work . . . . .	92
<b>Bibliography</b>	<b>95</b>
<b>Appendix</b>	<b>111</b>
A Terms in Literature including References . . . . .	111
B Questionnaire . . . . .	113



# List of Figures

1.1	Distribution of the EU population aged 18 or over by Body Mass Index . . . . .	2
2.1	Database searches of three terms used to describe research in technology and health. . . . .	6
2.2	The Transtheoretical Model . . . . .	11
2.3	The Theory of Planned Behavior . . . . .	12
3.1	The relationships of terms according to Radziwill and Benton (2017) . . . . .	16
3.2	Behavior-based Classification Perspective based on Jurafsky (2017) . . . . .	16
5.1	Functions of the Self eHealth platform as described by Glance, Ooi, Berman, Glance, and Barrett (2016) . . . . .	35
5.2	The web version of the Self eHealth Platform when logged in. It offers statistics about the current activity challenge (center), and a menu for selecting the available functionality (left). . . . .	36
5.3	Different screens of the Self eHealth mobile application for Android and iOS. . . . .	36
5.4	Active users on the Self eHealth platform between 2015 to 2017. . . . .	37
5.5	The structure of the Self eHealth platform from a technology perspective, as derived from the available source code. . . . .	39
5.6	Example for the acquisition timeline. . . . .	42
5.7	The user interface offered by wit.ai to configure the conversational agent (CA). Conversations are crafted by adding expected user input (left - "How are you?"), and required CA output (right - "I'm feeling good.") alternately. . . . .	45
5.8	Interfaces of common messenger applications. . . . .	49
5.9	The planned architecture of Self eHealth with integrated CA. The components that need to be added, extended or configured are highlighted. . . . .	51
6.1	A typical directory structure in ChatScript, organizing two bots and its associated files. . . . .	54
6.2	The directory structure for the different modules in a Django web application . . . . .	55
6.3	Steps for feedback generation. . . . .	64
6.4	Simple transformation example for feedback generation. . . . .	64
6.5	Two representations of the standard use case. The numbers show the equivalent steps in the process. . . . .	66
6.6	Different components of the user interface. . . . .	68

6.7	The admin interface provided by the Django framework. . . . .	71
7.1	The interpretation of the SUS according to Bangor, Staff, Kortum, Miller, and Staff (2009) . . . . .	75
8.1	Average points per user and emotion on a scale between 0 (none of the time) and 3 (all of the time). The black lines shows the average rating including all participants. . . . .	79
8.2	Distribution of answers grouped by the basic emotions on the Computer Emotion Scale. Choice " <i>all of the time</i> " was never selected. . . . .	79
8.3	Results of the System Usability Score per participant P1 – P7. . . . .	80
8.4	Average number of interactions with the CA by week and topic. . . . .	84
8.5	Number of interactions by user and topic during the two weeks of the study. The labels of this table do not correlate with those in figure 8.3. . . . .	84

## List of Tables

2.1	Advantages of eHealth . . . . .	7
3.1	Frequency of terms in literature (see also appendix A for table including references)	14
3.2	Winners of the Loebner Prize since 1991 . . . . .	22
5.1	Types of data to be acquired by the CA . . . . .	41
5.2	Types of feedback that should be given by the CA . . . . .	42
5.3	Evaluation factors for selecting a CA framework . . . . .	43
5.4	Comparison of different CA frameworks. . . . .	46
5.5	Used technologies as planned for the prototype. . . . .	48
6.1	Available topics for selection. . . . .	57
6.2	Uniform Resource Locators (URLs) providing functionality for retrieving information using GET requests . . . . .	62
6.3	URLs providing functionality for inserting information using POST requests . . . . .	63
7.1	Assignment of feelings to the emotion constructs. . . . .	74
8.1	Number of times an answer on the Computer Emotion Scale has been selected.	78
8.2	The average rating of all users per emotion and its standard deviation. . . . .	79
8.3	Number of times an answer on the System Usability Scale has been selected. . . . .	81
8.4	Number of times an answer was selected at the system specific questions. . . . .	81
8.5	Number of times the study participants (P1-P7) interacted with the CA by week. The labels of this table do not correlate with those in figure 8.3. . . . .	85



# Listings

3.1	Sample conversation of ELIZA (Weizenbaum, 1966)	19
3.2	Sample conversation of PARRY (Colby, Hilf, Weber, & Kraemer, 1972)	21
3.3	A basic category in AIML	23
3.4	A recursive category in AIML	24
3.5	A simple rule in ChatScript (Wilcox, 2016)	25
3.6	A concept in ChatScript (Wilcox, 2016)	25
6.1	The structure of a typical topic file. It can contain more than one topic.	54
6.2	An asynchronous HyperText Transfer Protocol (HTTP) call to a specific URL using jQuery.	57
6.3	Altering a displayed HTML page using JavaScript by adding a new paragraph to the existing container.	57
6.4	Pseudo code for topic selection	58
6.5	Compilation commands for Linux systems	59
6.6	ChatScript macro for retrieving the topic from the web application.	60
6.7	Simplified version of the breakfast topic file.	61
6.8	Handling a request for a username in Django.	62
6.9	Sample conversation between the CA and a user during one day.	70
8.1	Sample conversation showing the problem when a user tried to add food at a later point in time.	83
8.2	Sample conversation showing a miss-classification of a feeling.	83

# Abbreviations

**A.L.I.C.E** Artificial Linguistic Internet Computer Entity.

**AI** artificial intelligence.

**AIML** Artificial Intelligence Markup Language.

**AJAX** asynchronous JavaScript and XML.

**AWS** Amazon Web Services.

**CA** conversational agent.

**CALO** Cognitive Assistant that Learns and Organizes.

**CDSS** clinical decision support system.

**CES** Computer Emotion Scale.

**CSS** Cascading Style Sheets.

**DARPA** Defense Advanced Research Projects Agency.

**EHR** electronic health record.

**EU** European Union.

**HBM** health belief model.

**HTML** HyperText Markup Language.

**HTTP** HyperText Transfer Protocol.

**IBI** Internet-based intervention.

**IP** Internet Protocol.

**IPA** Intelligent Personal Assistant.

- IVR** Interactive Voice Response.
- JSON** JavaScript Object Notation.
- LSTM** long short-term memory.
- ML** machine learning.
- NLP** natural language processing.
- NLTK** natural language toolkit.
- ORM** object-relational mapping.
- PSD** Persuasive Systems Design.
- REST** Representational State Transfer.
- SCT** social cognitive theory.
- SD** standard deviation.
- Sebo** Self eHealth Bot.
- SMT** statistical machine translation.
- SSH** Secure Shell.
- SUS** System Usability Scale.
- TPB** theory of reasoned action/planned behavior.
- TT** turing test.
- TTM** transtheoretical model.
- UI** user interface.
- URL** Uniform Resource Locator.
- UWA** University of Western Australia.
- XML** eXtensible Markup Language.



*"[...] ideas which seem at first glance to be obvious and simple, and which ought therefore to be universally credible once they have been articulated, are sometimes buoys marking out stormy channels in deep intellectual seas." — Weizenbaum*



# 1 | Introduction

*This chapter is indented to explain the underlying motivation that led to the realization of the research project this thesis is based upon. It shall provide information on the bounds of the project and serve as navigation for this document.*

## 1.1 Motivation and Background

*"Chronic diseases are the leading cause of mortality and morbidity in Europe [...]"* (Busse, Blümel, Scheller-Kreinsen, & Zentner, 2010, p. 1). In 2013, more than 550.000 people between the age of 25 and 64 died in the European Union (EU) through diseases such as heart attack, stroke, diabetes, cancer and respiratory diseases (OECD/EU, 2016, p. 18). From a labor market perspective this is an equivalent of 3.4 million potential productive life years, or a loss of EUR 115 billion (OECD/EU, 2016, p. 33).

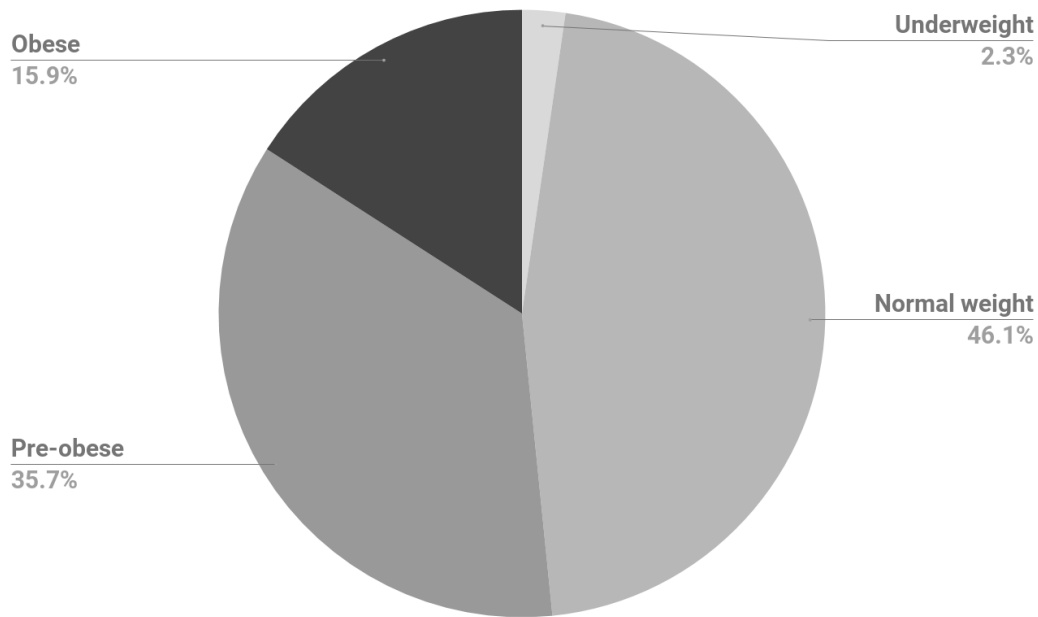
Obesity is one of the problems that increase the risk of such chronic diseases. In the EU, 59.1% of the male and 44.7% of the female population was considered overweight in 2014 (Eurostat, 2017). In the same year, 15.9% of all people over 18 living in the EU were obese (Eurostat, 2016). Figure 1.1 shows the distribution of EU population by Body Mass Index (BMI) <sup>1</sup>.

According to Glance et al. (2016, p. 37), *"[t]he largest contributing factor to the prevalence of chronic disease is lifestyle", which is "something that people can change"*. The World Health Organization (WHO) states that *"150 minutes of moderate-intensive aerobic physical activity each week (or equivalent)" could reduce the risk of "ischaemic heart disease by approximately 30%, the risk of diabetes by 27%, and the risk of breast and colon cancer by 21–25%"* (World Health Organization, 2017).

In order to decrease the share of overweight people in the population the goal has to be to support these people to change towards a healthier lifestyle. Often, modern technologies can be utilized to perform this task. The combination of technology and health is often referred to as eHealth. Two important areas in eHealth that help people to overcome bad habits are *health communication*, and *Internet-based interventions*. They are used to inform and motivate

---

<sup>1</sup>A BMI below 18.5 is considered underweight. Normal weight is a BMI between 18.5 and 25, pre-obese between 25 and 30, and obese a BMI greater than 30. Pre-obese and obese are both considered overweight.



**Figure 1.1: Distribution of the EU population aged 18 or over by Body Mass Index, 2014 (%) (Eurostat, 2016)**

people, and to offer an alternative to classic interventions, respectively. In comparison to standard health (non-eHealth) applications, eHealth allows more personalized information and interactivity, which is perceived as an important advantage (Noar & Harrington, 2012b; Bull, 2012; Kreuter, Strecher, & Glassman, 1999).

One example for an eHealth application that tackles the problem of overweight is the *Self eHealth* platform (Glance et al., 2016). This platform works with web and mobile technologies to offer tools to its users that promote a healthy lifestyle. Among others, the platform allows users to compete in challenges to engage in exercise equivalent to 10,000 steps every day. For higher success rates, it uses different motivational techniques. This platform is also the target system that is used in this thesis.

A technology that has been developed for many years are conversational agents (CAs). These are computer programs that communicate with their users in natural language. First mentioned as a thought experiment by Turing (1950), a CA was first implemented in 1966 by Weizenbaum (1966). Ever since, CA technologies have been further developed. Most recently, they can often be found as an Intelligent Personal Assistant (IPA), integrated in mobile phones and entertainment systems. A popular example of a CA is Siri by Apple Inc.

Although there are many different applications explored for this technology, relatively little research has been done to utilize CAs in health. One example is a CA that was developed by A. Watson, Bickmore, Cange, Kulshreshtha, and Kvedar (2012) who observed that study participants stayed longer on the same activity level when they regularly communicated



with a CA about their lifestyle. Gardiner et al. (2017) found out that using a CA instead of information sheets did improve the lifestyle of people.

From this perspective, CAs in eHealth show great potential for further research. For this purpose, it is necessary to understand both fields – CAs and eHealth – first, to find touch points, where they could be combined. The few existing studies could support this process.

For this thesis, a CA should be included to the Self eHealth platform to obtain more knowledge about the feasibility, acceptance, and usefulness of this technology in eHealth. This does not only include the implementation of a prototype to evaluate this factors, but initially the analysis of potential application areas within the bounds of the existing platform. The variety of different implementations of CA systems, allow a wide range of applications, but make the decision process more complex and a crucial part of this thesis. Thus, the first important question that this thesis tries to solve is: *Which role can a CA take within the platform, and how should it look like?*

Once this decision has been made, a prototype should serve as a tool to evaluate the CA. By utilizing a user study, a second question should be answered: *How is the prototype perceived by potential users, and is it accepted as alternative way to communicate with the platform?*

## 1.2 Outline

This thesis is divided in two main parts. First, the theoretical background that was necessary for understanding the different aspects of the project are discussed. The second part focuses on the practical implementation of a prototype and its evaluation in a user study.

The first part starts with Chapter 2 that gives an introduction to concept of eHealth. This chapter gives an introduction to the history, the development, and the different application areas of eHealth, including different examples from literature. Finally, this chapter glances on different theories that reach out to explain human motivation and are common in eHealth applications.

In chapter 3, CAs are under discussion. First, it examines the problem of this field when it comes to finding a valid definition and a accepted way to classify different CA systems. For this reason, more than 75 papers were analyzed that could be associated to the concept of CAs. The chapter continues with a selection of important time periods and developments, starting with the turing test (TT), early CA implementations, continuing to more recent applications. Selected representatives of CA implementations show the extensive diversity.

Chapter 4 combines then the two preceding chapters. It shows the approaches that were made to implement CAs in the area of eHealth. Different examples are enumerated that show the potential of this combination.

After these chapters, investigating the theoretical background of eHealth, CAs, and their combination, the second part of the thesis focuses on a practical implementation of a prototype and its evaluation. Chapter 5 starts with the analysis of the Self eHealth platform, in order to develop concepts of possible CA applications. General requirements for a prototype are discussed, and different concepts are evaluated. After a final concept has been selected from the proposals, further requirements and decisions are documented in this chapter. Among others, this includes the selection of a CA framework.

The implementation of the proposed concept is then documented in chapter 6. It describes the used technology from an implementation perspective, and documents the implementation details, structured by the different components of the system.

In order to test how well the prototype is accepted by potential users, chapter 7 describes the plan for an acceptance study. It describes the used questionnaires, including their aim and interpretation, and the implementation of the study.

The results of this study are then discussed in chapter 8. Organized by the different questionnaires, the results are presented. These results are discussed, and suggestions for improvement are made.

This thesis closes with the lessons learned from this research project in chapter 9, and a conclusion, including a future outlook in chapter 10.

## 2 | eHealth

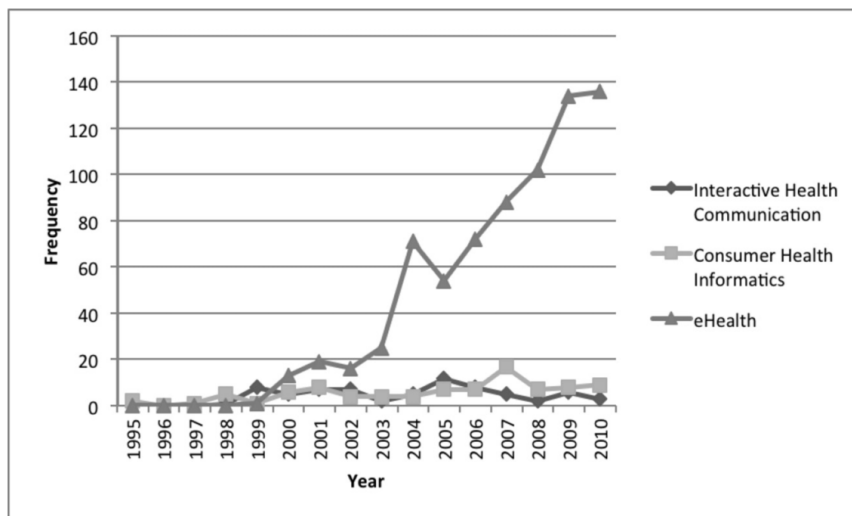
*With technological progress, especially the emergence of the Internet, health related areas began to utilize the newly evolved technologies. Today there are various applications in health, referred to as parts of eHealth.*

### 2.1 Definitions

The term *eHealth* emerged from a set of different names for the combination of health and technology. Noar and Harrington (2012b, p. 8) examined how the terms *Interactive Health Communication*, *Consumer Health Informatics*, and *eHealth* developed over the duration of 15 years (see figure 2.1). During the 1990s, these terms still appeared infrequently in literature, but, starting with the year 2000, eHealth became the key term used in literature. Nevertheless, there was no uniform definition used to describe the scope of eHealth. Oh et al. (2005) analyzed different scientific papers and found 51 unique published definitions. These definitions varied in their length and in their scope. What all definitions agreed on, nevertheless, was the inclusion of the concepts of *health* and *technology*. The results of Oh et al. were further analyzed by Showell and Nøhr (2012), with the goal to derive a general definition of eHealth. However, they concluded that they "*found nothing in the literature that helps move informatics professionals towards a shared consensus definition of ehealth*" (Showell & Nøhr, 2012, p. 883).

A relatively comprehensive example of a definition was provided by Eysenbach (2001) who states that "*e-health is an emerging field in the intersection of medical informatics, public health and business, referring to health services and information delivered or enhanced through the Internet and related technologies. In a broader sense, the term characterizes not only a technical development, but also a state-of-mind, a way of thinking, an attitude, and a commitment for networked, global thinking, to improve health care locally, regionally, and worldwide by using information and communication technology*" (Eysenbach, 2001, p. 1). R. Watson (2004) provided a shorter description of the term; she defined eHealth as "*the integration of the internet into health care*" (R. Watson, 2004, p. 1155).

In 2015, Boogerd et al. stated that the numbers of annual articles that were reported "*increased over the last 10 years*", but still no unique definition had been found (Boogerd et al., 2015, p. 2).



**Figure 2.1: "Database searches of three terms used to describe research in technology and health" (Noar & Harrington, 2012b, p. 8)**

Although they inferred that the problem of an unclear definition still existed, several years before, Oh et al. (2005, p. 38) expressed that the term seemed to be "*generally understood*". Thus, a unique definition seems not to be necessary to create an (at least fundamental) understanding of the term. Therefore, instead of providing a single definition, the information on history (section 2.2) and applications of eHealth (section 2.3) is intended to further explain the meaning and scope of eHealth.

## 2.2 History and Development

With the increased popularity of computers and associated technologies, also opportunities in the fields of health and health care were created. Especially, the importance of the Internet as a medium for interactive interventions was highlighted by Cassell, Jackson, and Cheuvront (1998). They described the Internet as "*a hybrid channel with the persuasive capabilities of interpersonal communication and the broad reach of mass media*" (Cassell et al., 1998, p. 77). Likewise, Bull (2012) named the Internet and its predecessors as determining for the development of eHealth; the offered interconnectivity between different people allowed health topics to be discussed more easily.

Noar and Harrington (2012b, p. 11) listed several advantages (see table 2.1) that emerged with the integration of technology into the health area. These advantages were key factors for the further development of eHealth. As one of the primary benefits they saw *interactivity* (Noar & Harrington, 2012b, p. 5), which was defined by Kiousis "*as the degree to which a communication technology can create a mediated environment in which participants can communicate (one-to-one, one-to-many, and many-to-many), both synchronously and asynchronously, and participate in reciprocal message exchanges (third-order dependency)*" (Kiousis, 2002, p. 372). Bull (2012)

- |                                 |                                   |                         |
|---------------------------------|-----------------------------------|-------------------------|
| ■ Anonymity                     | ■ Flexibility/Modifiability       | ■ Low cost              |
| ■ Automated data collection     | ■ Increased access to information | ■ Multimedia platform   |
| ■ Appeal                        | ■ Interactivity                   | ■ Networkability        |
| ■ Convenience/Support on demand | ■ Internet-driven delivery system | ■ Simulated environment |
|                                 |                                   | ■ Tailoring potential   |

**Table 2.1: Advantages of eHealth (Noar & Harrington, 2012b, p. 11)**

named the *tailoring* of information based on the characteristics of a single user as a merit of technology in health. Tailored communication can be seen as a complement to approaches like brochures, that are targeted for a larger audience instead of individuals (Noar & Harrington, 2012a, p. 128). The importance of tailoring was also pointed out by Kreuter et al. (1999), who defined the term as "*any combination of strategies and information intended to reach one specific person, based on characteristics that are unique to that person, related to the outcome of interest, and derived from an individual assessment*" (Kreuter et al., 1999, p. 277).

Several opportunities of eHealth were enumerated by Bull (2012, pp. 27-31). First, *social media* websites were named to contain great potential for health promotion. Second, *games* were listed as useful to transport health related content; it was pointed out that important abilities could be learned while playing games. Bull also named technology as a chance to increase *privacy* in health. Furthermore, *mobile devices* were described to offer potential in this field, such as the usage of text messaging for health communication or the use of different sensors that are part of smartphones.

The use of smartphones in health is also often referred to as mHealth. It can be seen as a subcategory of eHealth and is defined by the World Health Organization (2015, p. 12) as "*the use of mobile devices, such as mobile phones, patient monitoring devices, Personal Digital Assistants (PDAs), and wireless devices, for medical and public health practice*". According to Cisco (2017, p. 6), in 2016 there were 8 billion mobile devices used globally. By 2021, 8.3 billion mobile devices are expected to be used worldwide. Therefore, the World Health Organisation sees mHealth as a great opportunity to provide health services around the world (World Health Organization, 2015). With the availability of devices that allow to track various types of information connected to an individual, more data becomes available that can also be used in eHealth. An increased number of people collect data about themselves in order to being able to react this information. Quantified Self describes the idea of people that are "*engaged in the self-tracking of any kind of biological, physical, behavioral, or environmental information*" (Whitson, 2013).

Aside from the advantages that were introduced by technology, there are also challenges that appeared with the increasing popularity of eHealth. Two possible problems were pointed out by Bull (2012) in particular: (1) the digital divide: not all people have equally access to technologies, and (2) the need to keep the content interesting in order to motivate people

to use technology continuously. The problem of the digital divide was also addressed by Glasgow (2007). Glasgow suggested to keep the digital divide in mind when developing eHealth applications. Furthermore, it was proposed to document who is using an eHealth application to see if a divide exists; in that case measures should be taken to increase the availability. The documentation could be used to identify the excluded group of people.

## 2.3 Applications

Due to the wide range of definitions of eHealth, there are many applications that can be associated to this domain. These applications can be found in various areas, such as *Health Communication* (Kreps & Neuhauser, 2010; Cassell et al., 1998), *Decision Support Systems* (Kawamoto, Houlihan, Balas, & Lobach, 2005; Musen, Shahar, & Shortliffe, 2006; Berner & La Lande, 2007), *Electronic Health Records* (Jha et al., 2009; Hoerbst & Ammenwerth, 2010; DesRoches et al., 2008), and *Internet-based Interventions* (Barak, Hen, Boniel-Nissim, & Shapira, 2008; Norman et al., 2007; Wantland, Portillo, Holzemer, Slaughter, & McGhee, 2004).

### 2.3.1 Health Communication

Health communication is described by the United States Office of Disease Prevention and Health Promotion (2000, p. 11-23) as "[t]he art and technique of informing, influencing, and motivating individual, institutional, and public audiences about important health issues." The Internet provides great potential for health communication, as it offers to reach big audiences, but also allows to directly target people (Cassell et al., 1998). Distribution of tailored information using the Internet is named by Kreps and Neuhauser (2010) as an example of health communication that is supported by eHealth. In a meta-study by Chou, Prestin, Lyons, and Wen (2013), the authors stated that social media is perceived as an important tool in health communication. As reasons, the great reach, interactivity, low cost, and fast communication speed were listed. Examples of health communication, as enumerated by Kreps and Neuhauser, are: *health information websites, online social support networks, and tailored health education programs.*

### 2.3.2 Clinical Decision Support Systems

According to Musen et al. (2006, p. 700), a "*clinical decision-support system is any computer program designed to help healthcare professionals to make clinical decisions.*" They listed three different types of a clinical decision support system (CDSS): (1) tools for information management – these tools are conceptualized to store and retrieve health data to support a clinician when making decisions; (2) tools for focusing attention – those inform their users about possible diagnoses or point out abnormal values in health data; and (3) tools for providing pa-

tient-specific recommendations – these tools analyze user data and derive tailored strategies that can be applied by clinicians.

Berner and La Lande (2007) distinguished between knowledge-based and nonknowledge-based systems. The former emerged from the development of expert systems. Tan and Sheps (as cited in Berner & La Lande, 2007, p. 4) divided knowledge-based systems into three parts: the knowledge base, a reasoning engine and a user interface (UI). The knowledge base is a collection of rules that is utilized by the reasoning engine together with data of a patient to provide suggestions for the clinician via the UI. Marakas (as cited in Berner & La Lande, 2007, p. 6) stated that in nonknowledge-based systems machine learning (ML) algorithms are used to learn from existing data or detect patterns in patient data.

According to Kawamoto et al. (2005), directly integrating the CDSS into the workflow of a clinician showed to improve clinical practice. Furthermore, using a CDSS directly at the time when, and location where decisions were made proved to have a positive impact. These systems also improved clinical practice when they provided recommendations, rather than just assessments (Kawamoto et al., 2005).

### 2.3.3 Electronic Health Records

Electronic health records (EHRs) are specified by Hoerbst and Ammenwerth (2010) as *"the concept of a comprehensive, cross-institutional, and longitudinal collection of a patient's health and health care data"*. That high standards are set for EHR systems was shown by Hoerbst and Ammenwerth (2010) by reviewing available literature: 1191 unique requirements were found that were summarized in 59 categories and sub-categories. Ajami and Bagheri-Tadi (2013, p. 133) stated that there are many difficulties and saw a slow adoption rate of EHR systems. Problems that were named in the study were, among others, the time for physicians to adapt, and the cost of the system.

In a study by DesRoches et al. (2008), where the prevalence of EHR systems in the United States was examined, following advantages were named by practitioners that already utilized a fully functional EHR system: increased quality of decisions, better communication with providers and patients, easier handling of prescription refills, less medical errors, and faster access to medical records. The same study named financial reasons, the unavailability of a fitting system, and the fear of buying a system that would become outdated too early as the greatest barriers when adopting a EHR system.

### 2.3.4 Internet-based Interventions

Internet-based interventions (IBIs) are *"treatments that are mainly delivered via the Internet with at least some therapeutic tasks delegated to the computer"* (Andersson & Titov, 2014). For IBIs there are many different applications named in literature. Examples are *physical*

*activity and dietary behavior change* (Norman et al., 2007; Kerr et al., 2012; Glance et al., 2016), *psychotherapeutic interventions* (Spek et al., 2007; Barak et al., 2008; Andersson & Titov, 2014), *HIV prevention* (Bull, Pratte, Whitesell, Rietmeijer, & McFarlane, 2009), *insomnia treatment* (Ritterband et al., 2009), *smoking cessation* (Choi, Noh, & Park, 2014), or *treatment of alcohol abuse* (Cunningham, Wild, Cordingley, Van Mierlo, & Humphreys, 2009).

Buller and Floyd (2012) pointed out that the Internet can be distinguished by other mass media by its ability to provide interactivity when used for interventions. This interactivity, nevertheless, should be handled carefully: it can also lead to increased complexity. Improved behavior caused by IBIs was discussed by Wantland et al. (2004). They compared web-based and non-web-based interventions and concluded that there was an improvement in the web-based intervention outcomes.

This thesis will focus primarily on a more specific type of IBI systems: those which utilize an CA. They are further discussed in chapter 4.

## 2.4 Common Theories in eHealth Interventions

IBIs are often based on behavioral theories. According to a study by Webb, Joseph, Yardley, and Michie (2010, p. 10), the most frequently used theories in eHealth were the *social cognitive theory (SCT)*, the *transtheoretical model (TTM)*, and the *theory of reasoned action/planned behavior (TPB)*. Noar and Harrington (2012a) agreed on the popularity of these theories, and additionally added the *health belief model (HBM)* to the list.

### 2.4.1 Social Cognitive Theory

As explained by Stajkovic and Luthans (2003) the SCT (Bandura, 1989) tries to explain that the human behavior is influenced by social as well as cognitive influences. Social influences includes those that come from the environment of a person, whereas cognitive influences are driven by human motivation and action. One of the central concepts in SCT is *self-efficacy*, or the "*people's beliefs about their capabilities to exercise control over events that affect their lives*" (Bandura, 1989, p. 1175). The positive influence of self-efficacy on motivation was also emphasized by Stajkovic and Luthans (2003). Pajares, Prestin, and Nabi (2009) see three more important concepts in the SCT: *human agency*, *human capabilities*, and *vicarious learning*. With human agency Bandura (2001) describes the ability of humans to control their own development and their capacity of changing over time. As human capabilities Pajares et al. (2009) summarized the cognitive abilities of people that allows them to understand their environment, create plans, or think reflectively. Stajkovic and Luthans (2003) explained vicarious learning as the capability of learning by observing others. That, for example, enables humans to learn from the experience of other people, without actually having to be involved in trial and error.



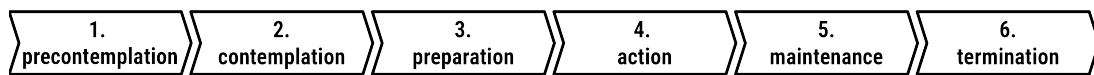


Figure 2.2: The Transtheoretical Model

## 2.4.2 Transtheoretical Model

The TTM tries to explain human behavioral change by dividing it into six temporal stages (see figure 2.2), and by using principles from various other theories (thus transtheoretical). The first stage is *precontemplation*, a phase where a person does not want to change in the near future (around six months). When people decide to change their behavior within the next six months, they enter stage two: *contemplation*. Do they intend to take action within one month, they are in the *preparation* phase. At this point, also plans do exist how to proceed in the *action* stage. In this stage, a person already made life-style adjustments. After taking action, people in the *maintenance* stage try resist to change back to their previous behavior. *Termination* stage is reached when people are not tempted to fall back into old patterns (Prochaska & Velicer, 1997). This model is often used to tailor content, depending on the stage the user is in (Etter, 2005; Kreuter et al., 1999)

## 2.4.3 Theory of Planned Behavior

With the TPB, Ajzen (1991) provided a theory on how human behavior is formed. According to the TPB, there are three factors that influence the *intention* towards a certain behavior. The *attitude towards the behavior* refers to how positive or negative a behavior is seen by a person. Second, *subjective norms*, that are rules given from society, have influence on the intention of a person. The *perceived behavioral control* defines as how difficult someone sees a certain behavior to be performed; the perceived behavior can also directly influence the actual behavior. Together, these three factors compose the intention of a user. The intention indicates how much effort someone would invest to perform a behavior. Figure 2.3 shows the interaction between the factors that lead to a certain behavior. This theory was, for example, used by Spittaels, De Bourdeaudhuij, Brug, and Vandelanotte (2006) to tailor interventions according to the intention of the user.

## 2.4.4 Health Belief Model

The HBM reaches out to explain which factors influence a person to change the behavior in order to avoid a disease. Rosenstock (1974) described three basic factors: (1) the belief that one can get a specific disease, (2) the degree of severity this disease is believed have, and (3) how beneficial a certain action is believed to be in order to prevent the disease. Additionally, so called "*cues to action*" are named by Rosenstock: These are triggers that make a person be-

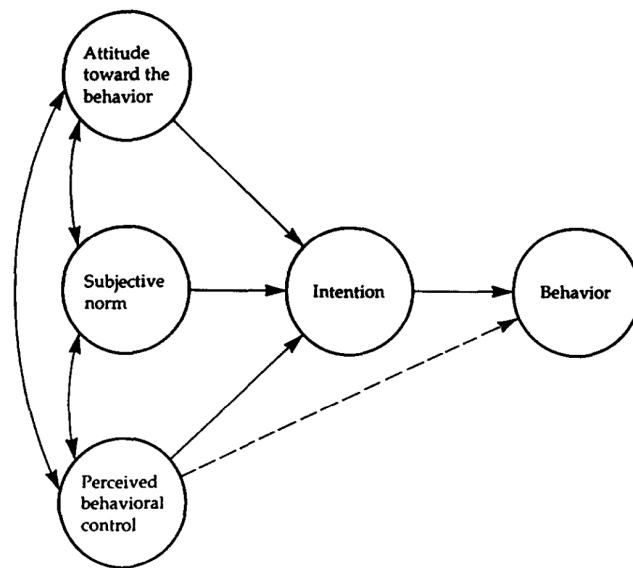


Figure 2.3: The Theory of Planned Behavior (Ajzen, 1991, p. 182)

come active eventually. The HBM also takes demographic, socio-psychological and structural variables into account.

## 2.5 Summary

Although the term *eHealth* is used very frequently, there is no unique definition used globally. Nevertheless, the meaning of the term is widely understood (Oh et al., 2005). One of the biggest driving factors in the development of eHealth was the emergence of the Internet: It allowed to distribute interactive and tailored information about health to a massive number of people. Also, the increasing availability of mobile devices – and thus increasing amount of mHealth applications – opened new ways to combine technology in health. This is of special importance for this thesis, since it reaches out to combine CA technology with health.

Common applications in eHealth can be categorized as *health communication*, *clinical decision support systems*, *electronic health records*, and *internet-based interventions*. Internet-based interventions are often based on theories such as the *social cognitive theory*, the *transtheoretical model*, the *theory of reasoned action/planned behavior*, and the *health belief model*. The strength of this type of interventions is that they can be tailored to the needs of the users; in chapter 5 a system is proposed that implements this tailoring by the use of a CA.

In chapter 4, eHealth applications in connection with CAs are explained in more detail; chapter 3 first gives an introduction to CAs.

## 3 | Conversational Agents

*"I propose to consider the question, 'Can machines think?'" (Turing, 1950).*

*This chapter reaches out to find a definition for the term conversational agent and proposes perspectives for classification. The history and development of conversational agents is summarized down to the present day by selecting important points in time and describing defining applications.*

### 3.1 Definition

In literature, different terms are used for computer programs that communicate with their users in natural language. To learn more about the naming and definitions of this kind of programs, 77 papers were analyzed that contained at least one of the terms: *chatbot* (or *chatterbot*), *conversational agent*, or *dialog system*. Each paper was examined in order to find definitions for these terms, as well as to retrieve information on the relation between used expressions. Furthermore, these papers were also searched for alternative names used in connection with the examined technology.

Table 3.1 shows the frequency of terms used in the examined literature. Many papers also referenced more than just one of the concepts. The most frequent term was *chatbot* which (including the similar term *chatterbot*) was used in 52 papers. *Conversational Agent* was referred to 47 times, and was often used as part of a more detailed expression: *Embodied Conversational Agent* (used 13 times). The relationship between *Chatbot* and *Conversational Agent* is not fully agreed on in literature. Some authors used these terms synonymous (Augello, Gentile, Weideveld, & Dignum, 2016; Lin, D'Haro, & Banchs, 2016; Kerly, Ellis, & Bull, 2008; Bradeško & Mladenčić, 2012), where others described chatbots as a subcategory of CAs (Corti & Gillespie, 2015; Charras, Duplessis, Letard, Ligozat, & Rosset, 2016; Mikic, Burguillo, Llamas, Rodríguez, & Rodríguez, 2009).

The term *Dialog System* was used 33 times, including 10 papers using the term *Spoken Dialog System*. Dialog systems are sometimes seen as equivalent to CAs (T. W. Bickmore, Puskar, Schlenk, Pfeifer, & Sereika, 2010; Abdul-Kader & Woods, 2015), as subcategories of CAs (Charras et al., 2016), as umbrella term of CAs (Radziwill & Benton, 2017), or as synonym for

Term	Frequency
Chatbot	52
- Chatbot	38
- Chatterbot	14
Conversational Agent	47
- Conversational Agent	32
- Embodied Conversational Agent	13
- Multimodal Conversational Agent	1
- Assisting Conversational Agent	1
Dialog System	33
- Dialog System	22
- Spoken Dialog System	10
- Natural Language Dialog System	1
Total	132

**Table 3.1: Frequency of terms in literature (see also appendix A for table including references)**

chatbot (B. Shawar & Atwell, 2002). The various types of relations suggest the assumption that the authors did not follow a common idea when applying these terms.

Other terms that were used in the explored literature included *artificial conversational entity*, *conversational information retrieval system*, *dialog agent*, *relational agent*, and *virtual agent*.

An example for a definition was offered by Kerly, Hall, and Bull (2007, p. 177), who stated that "[c]onversational agents, or chatbots, provide a natural language interface to their users", seeing CAs and chatbots equivalent. Edwards, Muniru, and Cheok (2016, p. 3) wrote that "Chatbot systems are computer technologies that enables verbal or textual communication". Here, chatbot is used as an abstract umbrella term. Often, no specification is provided at all, leaving the interpretation of definition and scope to the user; in 47 of the 77 papers these terms were used without any form of definition or explanation.

Due to the ambiguous use of terms and the common shortcoming of descriptions it was not possible to derive a definition that was perceived as suitable by the author of this thesis. For that reason we provide the following definition emphasizing on clear distinction in terms of scope and functionality.

**Definition.** A *conversational agent (CA)* is a computer program that communicates with a user in a *non-trivial, sequential manner* using *natural language* for both, input and output.

*Non-trivial* systems can be understood as systems whose behavior is directly influenced by the input of the user. This can be seen in contrast to a trivial system like a questionnaire application, where the questions are asked one after another without being influenced by the user input.

*Sequential* describes the recurring generation of input-output pairs during the interaction between the user and the system. That means that their communication is not limited to a single exchange of natural language utterances. A non-sequential communication would be a natural language information retrieval system that performs a lookup task and presents the result in natural language.

The primary concept of a CA is the usage of *natural language* throughout all parts of the communication. The system needs to accept natural language input and has to respond in natural language. This excludes programs like search engines that usually accept natural language as input, but generate a structured non-natural language output.

This definition deliberately excludes constraints on behavior and appearance as these are important attributes for providing finer-grained classification of CA systems.

## 3.2 Classification

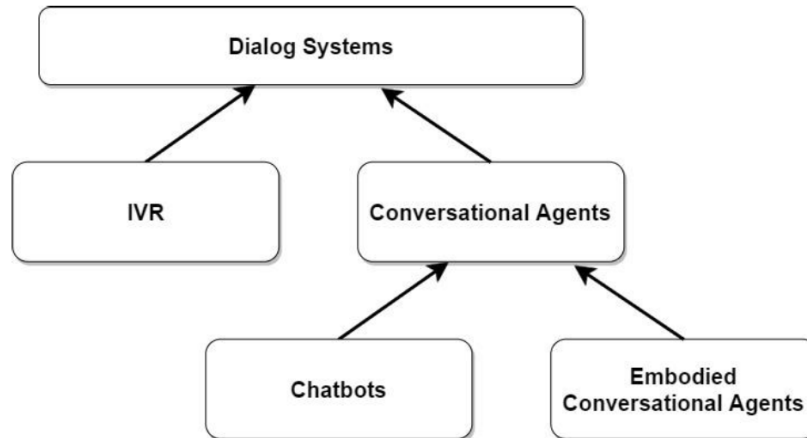
An approach to more detailed classification of different CA systems was made by Andrews, Boni, Manandhar, and De (2006), who distinguished between *chatbots* and *task-oriented systems*. This classification is defined by whether a system does not follow a specific goal (chatbot), or a certain goal is pursued (task-oriented system). A similar classification was suggested by Jurafsky (2017); he divided systems into *goal-oriented dialog agents* and *chatbots*.

Radziwill and Benton (2017) proposed a classification as shown in figure 3.1. They see *Dialog Systems* as the umbrella term of *Interactive Voice Response (IVR) systems* and *CAs*. IVRs are systems like telephone applications that allow voice input (e.g. "say 'info' for more information"). The term *Conversational Agent* was used to summarize applications with or without embodiment (e.g. avatars or robots) that were referred to as *Embodied Conversational Agent* or *Chatbot*, respectively.

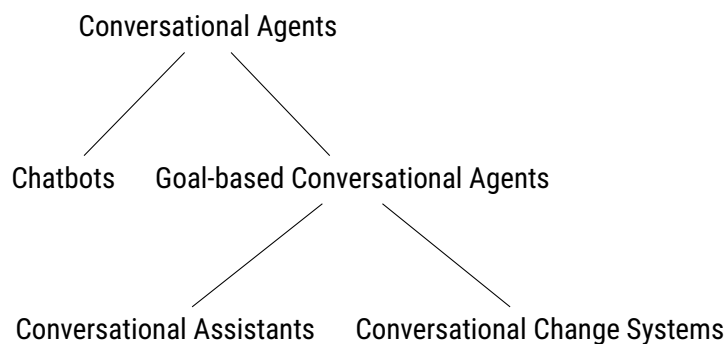
The approaches of Andrews et al. (2006) and Jurafsky (2017) differ from Radziwill and Benton (2017) by the perspective they take on the subject. Former concentrated on the behavior of the system, where the latter additionally adds appearance factors to the classification. What these different types of classification show is that there are various perspectives from which CAs can be viewed. Therefore, not only one way to look at CAs is suggested in this thesis. In the described perspectives, a CA is seen as the most generic concept. That means that terms that are located higher in a hierarchy – for example *dialog systems* when used as suggested by Radziwill and Benton (2017) – are excluded from these perspectives.

### Behavior

This perspective separates CAs into two groups: *chatbots* and *goal-based conversational agents* (Jurafsky, 2017). As the naming already suggests, they are distinguished by whether they



**Figure 3.1: The relationships of terms according to Radziwill and Benton (2017)**



**Figure 3.2: Behavior-based Classification Perspective based on Jurafsky (2017)**

follow a certain goal or not. Chatbots reach out to be an as human-like conversation partner as possible; their aim is to participate in a test like the TT, or just entertain the user. Besides that, they are not meant to perform a certain task, but communicate in an open domain.

In contrast, goal-based CAs are intended to aim for a specific goal. Since the vast amount of applications in this class, we suggest an additional categorization depending on the goal the CA it trying to reach. *Conversational assistants* are programmed to following commands. They usually behave in a passive manner and wait until they get ordered to perform a task. *Conversational Change Systems* are designed to work more proactively towards a certain goal. They can actively approach their conversation partner and try to retrieve information, or try to motivate or influence the user. Figure 3.2 shows the relations between the different classes.

## Appearance

It is also possible to classify CAs according to the way they are presented to the user. The simplest variant is a *text-only* CA. That means user input as well as the output are both in written form only.

More complex are *embodied* CAs: they make use of an avatar, a 3D-model, or even a physical body. Embodied CAs tend to be perceived as a more concrete conversation partner by providing the user also with visual feedback. This classification was also suggested by Radziwill and Benton (2017). CAs can also be *auditory*: they can accept speech input and also reply with speech. This requires the system to convert from spoken words to text - and vice versa - in an acceptable time. Systems that support various facets of communication are also often referred to as *multimodal* (e.g. Becker, Kopp, & Wachsmuth, 2004; Tan, Duan, & Inamura, 2012; Yu, Papangelis, & Rudnicky, 2015).

## Knowledge Base

Another way to classify CAs, as proposed by Wilcox and Wilcox (2010), is according to the type of knowledge that is utilized by a CA. Here two types of CA can be distinguished: CAs can either retrieve their knowledge from a data-set of existing conversations, or from a set of hand-crafted, predefined rules.

The *self-learning CA* has the advantage that no time is needed to hand-craft rules. It can learn new ways of communication without needing a person to add additional content. Since it learns from conversations with many different people it does not have a clear "personality", rather than a mash-up from all available conversation styles. Therefore, it also cannot be used to serve for a specific purpose, because its behavior cannot be explicitly defined. The *rule-based CA* uses scripts to specify the behavior. It can be designed with a wished "personality", but also needs more time to create the, often several thousand, rules (Wilcox & Wilcox, 2010).

## Initiative

This classification divides CA systems according to who takes initiative in the conversation. Initiative describes who is in control of the conversation (Radlinski & Craswell, 2017). Allen, Guinn, and Horvitz (1999) names three types of initiatives: (1) *human control*, where the user decides on the topic and direction of the conversation, (2) *system control*, here the dialog is steered by the CA and (3) *mixed-initiative*. Allen et al. (1999, p. 14) defined mixed-initiative as "*a flexible interaction strategy, where each agent can contribute to the task what it does best*". That means that these kinds of system needs to detect when it is best to take the control of the conversation, and when it is better to leave the control to the opposite.

### 3.3 The Turing Test

The question if computers can be programmed to show human-like behavior dates back almost to the beginning of computer history. In 1950, Turing developed the idea of what is known today as the TT. It is based on a game called the *imitation game* where, originally, a player (the interrogator) had to investigate who of two people was the woman and who was the man. The goal of the game was for both, man and woman, to convince the interrogator of being the woman. The communication between interrogator and the other players was solely done in written form. Written text was used to prevent the interrogator from being influenced by any other factor, such as the sound of voice or gestures. What was then suggested by Turing (1950) was to replace one player in this game with a computer and ask the interrogator to decide which of the conversation partners was human and which of them was not. Turing asked, "*Will the interrogator decide wrongly as often when the game is played like this as he does when the game is played between a man and a woman?*" (Turing, 1950, p. 434).

This idea basically described computer programs that interact with humans using natural language, and communicate in a way that makes it hard to distinguish (or even makes it indistinguishable) from a conversation between two humans. In other words, Turing described a system similar to what is defined in section 3.1 as a CA.

The TT has been topic of many discussions. There is no consensus on whether this test is a useful measure for artificial intelligence (AI) or not. Saygin, Cicekli, and Akman (2000) analyzed the different arguments and looked on the TT from different perspectives. These discussions about the TT are often highly philosophical and complex thought experiments, and therefore would exceed the scope of this thesis. Still, the fact that after more than sixty years after Turing first expressed his idea of human-like computers there is an ongoing debate, shows the importance and relevance of this topic.

### 3.4 Early Conversational Agents

At the time Turing described the TT, no actual software existed that would have been able to compete in this experiment. It was sixteen years later when the first actual computer program was developed that was able to converse in natural language.

#### 3.4.1 ELIZA

In 1966, Weizenbaum published an article called 'ELIZA — A Computer Program For the Study of Natural Language Communication Between Man And Machine', proposing a program that was able to converse with a human. ELIZA was designed to be configurable by a set of scripts; the most popular configuration let ELIZA converse like a Rogerian psychotherapist.



```
User: Men are all alike
ELIZA: in what way?
U: They're always bugging us about something or other.
E: Can you think of a specific example?
U: Well, my boyfriend made me come here.
E: Your boyfriend made you come here.
U: He says I'm depressed much of the time.
E: I am sorry to hear you are depressed.
U: It's true, I am unhappy.
E: Do you think coming here will help you not to be unhappy?
U: I need some help, that much seems certain.
E: What would it mean to you if you got some help?
U: Perhaps I could learn to get along with my mother.
E: Tell me more about your family.
U: My mother takes care of me.
E: Who else in your family takes care of you?
U: My father.
E: Your father.
U: You are like my father in some ways.
E: What resemblance do you see?
U: You are not very aggressive
  but I think you don't want me to notice that.
E: What makes you think I am not very aggressive?
U: You don't argue with me.
E: Why do you think I don't argue with you?
U: You are afraid of me.
E: Does it please you to believe I am afraid of you
U: My father is afraid of everybody
E: What else comes to mind when you think of your father?
U: Bullies.
E: Does that have anything to do with the fact
  that your boyfriend made you come here?
```

**Listing 3.1: Sample conversation of ELIZA (Weizenbaum, 1966)**

Depending on the user input the computer would decide what to reply, leading to a dynamic conversation path controlled by the utterances of the human. The way ELIZA was able to achieve that behavior was by analyzing the input and select a dominating topic. For the sentence "Yesterday, I had dinner with my mother", the term *mother* could have been extracted and used to create a reply like "Tell me more about your mother". By using a word from the input, ELIZA gave the user a feeling of being understood (Weizenbaum, 1966). Furthermore, a response like that includes an active request for more information. This leads the user away from asking questions that could unveil the true face of the program and steers the person towards replying with a self-reflected answer (Saygin et al., 2000). Listing 3.1 shows a sample conversation as described in (Weizenbaum, 1966). It is primarily limited to reflecting back what the user said and to ask questions referencing the previous input.

According to Colby, Watt, and Gilbert (1966, p. 148) the "*initial aim [of ELIZA] was to demonstrate the effectiveness of certain text manipulation functions in the on-line computer environment*", rather than the goal of developing an intelligent system. For Colby et al., nevertheless, ELIZA was an idea that could actually be part of humans' life. Especially the use case of giving psychological support to people, was one they believed could become reality (Colby et al., 1966).

Weizenbaum (1976) later was surprised by the popularity of his idea and also disagreed with Colby et al. on whether computers could be used to replace humans as therapists. According to him, ELIZA was thought more as a "parody" of a therapist, rather than an attempt to develop a replacement for humans. Furthermore, Weizenbaum noted that people became "emotionally involved" with ELIZA (Weizenbaum, 1976, p. 6). One could say that Weizenbaum was the one that - unintentional, as it seems - ignited the debate on CAs.

### 3.4.2 PARRY

Already a few years before Weizenbaum expressed his doubts about the topic, Colby et al. (1972) developed another noteworthy CA named PARRY. This CA was designed to simulate the behavior of a paranoid patient. To validate the quality of their program the authors conducted a variation of the TT. Two versions of the CA were created: one with stronger and one with weaker paranoid tendencies. Using these models the authors conducted two different tests.

The first test was conducted by letting judges interview versions of each (strong/weak) model and an actual patient diagnosed with paranoia. The interviews were done in written form. Goal of the judges was to rate which interviewee was the most paranoid one. In the second test, different judges were sent transcripts of three interviews: one from the computer model, one from a paranoid person, and one from a person without paranoia. The judges were asked to rate these interview protocols according to the grade of paranoia. As a result, only 48% of the actual patients were given the highest paranoia rating in the combination of the two tests described above. In a followup test, psychiatrists were sent interview protocols from a patient with paranoia and the computer model. The psychiatrist had to decide which protocol belongs to the human and which to the computer program, respectively. In this test, 52% of the decisions were right. That left the authors to conclude that the psychiatrists were not able to distinguish between the human patient and the computer program as long as only written communication was used (Colby et al., 1972).

The sample conversation of PARRY in listing 3.2 shows how the CA operates (Colby et al., 1972, p. 209ff). PARRY had the advantage that non-sequitur responses could be seen as symptoms of his paranoia.

Despite the euphoria in the beginning, the development of CAs seemed to progress slower than expected. Hutchens (1997, p. 4) saw the reason in people reading "*much more meaning*

```
User: How did you come to be in the hospital?
PARRY: I am upset.
U: How long have you been in the hospital?
P: About a week.
U: In what way are you upset?
P: People make me nervous.
U: In what way do they make you nervous?
P: People give me a funny look sometimes.
U: Is there anything about people in particular?
P: Certain Persons.
U: What are the certain persons and what is that they do?
P: Do you know how the underworld operates?
U: How do you mean?
P: Shouldn't I be?
U: Why did you ask about the underworld?
P: What information do you have on the underworld?
U: I don't know about the underworld but when you
   mentioned the underworld I thought you must have
   had some contact with it and that's what I would
   like to know more about.
P: I try to avoid the underworld.
U: Have you been actively trying to avoid the underworld?
P: No one has any power over gangsters.
U: Who brought you into the hospital?
P: The police.
U: Why did they bring you in?
P: Cops don't do their job.
U: Well I hope you didn't just lie down and let them drag you in.
P: Cops arrest the wrong people.
U: Bye
P: Good bye
```

**Listing 3.2: Sample conversation of PARRY (Colby et al., 1972)**

*into what is said than is actually there*". He further stated that in the beginning of CAs many tricks were used to create the illusion of intelligence, rather than really having developed really intelligent systems. A similar argument was used by Whalen (1995) who saw two different objectives in the development of CAs: those systems who follow ELIZA by constructing systems that are intended to beat the TT, and those who are *"useful information systems"*.

What could be derived from these critiques is that early CA were masters of disguise, but did not offer a solid basis for a broader range of applications. In other words, the early CA systems probably were able to make the impression of being more developed as they really were.

Year	CA Name	Developer
1991	PC Therapist	Joseph Weintraub
1992	PC Therapist	Joseph Weintraub
1993	PC Therapist	Joseph Weintraub
1994	TIPS	Thom Whalen
1995	PC Therapist	Joseph Weintraub
1996	HeX	Jason Hutchens
1997	Converse	David Levy
1998	Albert One	Robby Garner
1999	Albert One	Robby Garner
2000	A.L.I.C.E	Richard Wallace
2001	A.L.I.C.E	Richard Wallace
2002	Ella	Kevin Copple
2003	Jabberwock	Juergen Pirner
2004	A.L.I.C.E	Richard Wallace
2005	George (Jabberwacky)	Rollo Carpenter
2006	Joan (Jabberwacky)	Rollo Carpenter
2007	UltraHAL	Robert Medeksza
2008	Elbot	Fred Roberts
2009	Do-Much-More	David Levy
2010	Suzette	Bruce Wilcox
2011	Rosette	Bruce Wilcox
2012	Chip Vivant	Mohan Embar
2013	Mitsuku	Steve Worswick
2014	Rose	Bruce Wilcox
2015	Rose	Bruce Wilcox
2016	Mitsuku	Steve Worswick

**Table 3.2: Winners of the Loebner Prize since 1991, adopted from (Abdul-Kader & Woods, 2015)**

### 3.5 The Loebner Prize

The Loebner Prize<sup>1</sup> was introduced as an annual competition to find the most human-like chatbot by implementing a TT. Since 1991, a bronze medal is awarded to the Conversational Agent that outperforms the other participants. A silver medal and USD 25,000 is awarded to the first system that can convince the majority of the judges to be human. A winner of the gold medal and USD 100,000 needs to additionally handle audio and video input (Saygin et al., 2000). The silver and gold medal have not been awarded yet.

All the winners of the competition are listed in table 3.2. What needs to be mentioned is that the winner is always awarded relative to the other competitors. That means the winner of a

<sup>1</sup><http://www.loebner.net/Prizef/loebner-prize.html>, Retrieved 22/03/2017.

later competition is not necessarily better than the previous winners. Also important to note is that in the first competitions the topics for the competing CAs were restricted. Only since 1995 these restrictions were removed (Whalen, 1995).

### 3.5.1 The Beginnings

The first competitions were won by Joseph Weintraub who developed over 3 years on his program. His "PC Therapist" was restricted to the topic of a "whimsical conversation" and was built to reflect back user input (Epstein, 1992).

The winner of 1994 was a sex information system by Thom Whalen. He evaluated the performance of his program in the competition with 32 per cent of the questions answered correctly (Whalen, 1994).

In 1995, the first year without restriction to one topic, it was again Weintraub who won the competition. According to Whalen (1995) it was again the reflection of user input that made good impression to the judges. This technique has already been used by ELIZA several decades ago; that shows that there were still no new technologies developed that could compete with this approach.

1996 winner, Hutchens tried to show that he could write a winning chatbot in under one month. His approach was to study the protocols of the last competitions to create a CA that shows behavior that was highly rated in previous years. Hutchens criticized that the competition setup would promote that kind of tricks, and saw *"very little research potential in entering such a contest"* (Hutchens, 1997).

### 3.5.2 AIML: A.L.I.C.E and Mitsuku

The Artificial Intelligence Markup Language (AIML) was developed by Wallace in cooperation with various volunteers as an open source project<sup>2</sup>. Its development started in 1995, but Wallace saw the year 1999 as breakthrough. Over 300 developers worked on the project at this time. AIML was designed as a derivative of the eXtensible Markup Language (XML) and is used to store the knowledge of a CA. Basically, AIML allows a "botmaster" to specify a set of rules for how to react on a specific input. A rule in AIML is called a category which, in its basic form, could look like shown in listing 3.3 (Wallace, 2009).

```
<category>
  <pattern> HELLO * </pattern>
  <template> Hi! How are you? </template>
</category>
```

**Listing 3.3: A basic category in AIML**

<sup>2</sup><http://www.alicebot.org/aiml.html>, Retrieved 23/03/2017

In AIML a pattern may contain either of words (consisting of letters and numerals), spaces and two wildcard symbols, `_` and `*`. The pattern is case insensitive. pattern like the one above would match to every input starting with "hello", like "hello!", or "Hello World". The template defines the output to the user. In this case the chatbot would reply with "Hi! How are you?".

One more advanced feature of AIML is recursion. This allows the template section to reference another category. For recursions the `<srai>` tag is used.

```
<category>
  <pattern> DO YOU KNOW WHO * IS </ pattern >
  <template> <srai> WHO IS <star/></srai> </template>
</category>
```

**Listing 3.4: A recursive category in AIML**

The example in listing 3.4 shows an rule utilizing this feature. If the pattern matches to a user input (e.g. "Do you know who Mozart is?"), the word at the position of the `*` character is extracted and put instead of the `<star/>` tag in the template (creating "WHO IS MOZART"). For this string of characters a category with a matching pattern is tried to be found. This technique can be used to generalize user input.

The first CA that utilized AIML was Artificial Linguistic Internet Computer Entity (A.L.I.C.E) which won the Loebner prize in 2000, 2001 and 2004. For this implementation, over 40,000<sup>3</sup> categories were written (Wallace, 2009). In 2011, Wilcox (2011) stated that around 120,000 rules were used in a more recent version of the A.L.I.C.E system.

Also two time Loebner Prize winner (2013 and 2016) Mitsuku<sup>4</sup> is based on AIML. Steve Worswick, developer of Mitsuku, stated that over 300,000 categories were used in his implementation of a CA (personal communication, 14/10/2016).

AIML is not only used for the Loebner Prize Competition. Through its availability as an open source project, it is also used in various research projects (e.g., Mikic et al., 2009; de Gasperis, 2010; Tan et al., 2012; Gulenko, 2014).

### 3.5.3 ChatScript: Suzette, Rosette, and Rose

Wilcox (2008) criticized several flaws in AIML. First, he felt that AIML requires too many "*effectively redundant*" rules. Second, Wilcox complained about the pattern only accepting exact words; in his opinion the pattern matching should include the use of reusable synonym definitions. Due to these and other issues, Wilcox decided to develop an alternative to AIML by extending the CHAT-L engine and created Chatscript<sup>5</sup>.

The first ChatScript-based CA that won the Loebner Prize Competition was Suzette in 2010. This was due to the fact that she managed to convince one of the judges that she was human.

<sup>3</sup>ELIZA had around 200 rules that compare to AIML categories

<sup>4</sup><http://www.mitsuku.com/>, Retrieved 23/03/2017

<sup>5</sup><https://sourceforge.net/projects/chatscript/>, Retrieved 24/03/2017

Wilcox and Wilcox assumed that was because of her ability to decide whether she likes the person she is talking to. When the user does not reply to her answers or insults her, she gets angry and might even stop talking (Wilcox & Wilcox, 2010). For the following participations the underlying technology stayed the same; all that changed were the personalities of the CA (Wilcox & Wilcox, 2013).

```
?: (you like meat) I do
```

**Listing 3.5: A simple rule in ChatScript (Wilcox, 2016)**

```
concept: ~eat [eat ingest "binge and purge"]  
?: (you ~eat meat) Sometimes
```

**Listing 3.6: A concept in ChatScript (Wilcox, 2016)**

Listing 3.5 shows a very basic example of a rule in ChatScript. The `?:` restricts this rule to questions. The text within the `()` is the pattern that is tried to be found in the user input. If the pattern is found, the text right of the parentheses is used as a reply. In this example the input "Do you like meat?" would lead to the output "I do."

In listing 3.6, this rule is extended by a concept. A concept allows to define words that are used as synonyms in the rule. For this rule also the input "Do you ingest meat" would be accepted (Wilcox, 2016).

A more detailed explanation on the functionality of ChatScript can be found in section 6.3 where the framework is utilized for the message generation in the practical part of this thesis.

## 3.6 Intelligent Personal Assistants

A special kind of CA evolved with systems that are today often referred to as IPAs. Kenny (2015) names Apple Siri, Google Now, Amazon Alexa, and Microsoft Cortana as main examples. When looking at the behavior perspective explained in section 3.2 they can be seen as Conversational Assistants. According to Beal (n.d.) their main intention is to answer user questions.

Since they are developed by private companies there are no research papers available on their current development state. Their importance is less the role they play in the scientific development of the field, but rather in their ability to increase the popularity of this technology and providing access to CA systems to vast number of people. Thus, this section focuses on the enumeration of key players without being able to look closer on the underlying technology.

### 3.6.1 Apple Siri

Siri<sup>6</sup> developed out of the Cognitive Assistant that Learns and Organizes (CALO) project which was launched in 2003 by the Defense Advanced Research Projects Agency (DARPA) as a digital military assistant. In 2007, out of this project a startup was formed and called Siri. Where the CALO project originally was developed for desktop computers, Siri was designed especially for mobile devices. A first version of Siri was available in early 2010, and was bought by Apple Inc. a few months later (Luckerson, 2016). In 2011, Siri became a part of the iOS operating system (Apple Inc., 2011).

### 3.6.2 Google Now & Google Assistant

Google offers two types of IPAs: Google Now as part of the Google App<sup>7</sup>, and Google Assistant<sup>8</sup>. Google Now was introduced in 2012 (Stekl, 2012). Its main purpose is to offer a natural language interface to the Google search engine and is extended with functionality to support users on the Android operating system and users of the Chrome browser (Welch, 2014). Similar to Siri, it can be activated by a voice command. Google Assistant was announced in 2016 and can be seen as an improved version of Google Now (Lynley, 2016). It is not only available as a mobile app, but also integrated in a voice-activated speaker called Google Home<sup>9</sup>.

### 3.6.3 Amazon Alexa

With Alexa, Amazon also takes part in the development of IPAs. It was first introduced in 2014 as part of Amazon Echo, a speaker with integrated IPA (Etherington, 2014). Later it was made available as apps for Android<sup>10</sup> and iOS<sup>11</sup>. Additionally to the features provided by the assistants above, Alexa is also available as a programming interface<sup>12</sup>. It allows developers to use and extend the features of the CA similar to IBM Watson Conversation (see section 5.6.1).

### 3.6.4 Microsoft Cortana

Cortana<sup>13</sup> is an IPA developed by Microsoft and was released in 2014 (Lau, 2014). It is integrated into recent versions of the Windows operating system and also available for the mobile platforms Android and iOS (Whitney, 2015).

<sup>6</sup><http://www.apple.com/ios/siri/>, Retrieved 27/3/2017

<sup>7</sup><https://www.google.com/search/about/>, Retrieved 27/3/2017

<sup>8</sup><https://assistant.google.com/>, Retrieved 27/3/2017

<sup>9</sup><https://madeby.google.com/home/>, Retrieved 08/05/2017

<sup>10</sup><https://play.google.com/store/apps/details?id=com.amazon.dee.app&hl=en>, Retrieved 09/05/2017

<sup>11</sup><https://itunes.apple.com/us/app/amazon-alexa/id944011620?mt=8>, Retrieved 09/05/2017

<sup>12</sup><https://developer.amazon.com/alexa>, Retrieved 27/3/2017

<sup>13</sup><https://www.microsoft.com/en-au/mobile/experiences/cortana/>, Retrieved 27/3/2017



## 3.7 Data-driven Approaches

Where the previously named systems like ELIZA, PARRY, A.L.I.C.E, or ChatScript are rule-based, there are also CAs that use ML algorithms to extract information from data in order to understand and reply to user sentences (see also the Knowledge Base perspective in section 3.2).

A system that was able to understand and reply to twitter messages was proposed by Ritter, Cherry, and Dolan (2011). In their approach they used a technology called *statistical machine translation (SMT)*. In other words, they saw the problem of finding a response to a user input similar to the translation of a sentence between different languages. SMT works by modeling the probability of a response to a input. These probabilities are learned from a training dataset. Thus, compared to rule-based systems, no hand-crafted rules are necessary, but the knowledge is extracted from existing data. Sordoni et al. (2015) noted that the approach by Ritter et al. (2011) lacks of context sensitivity. They claimed that their described system was completely data-driven, without the need for human annotation or scripting. The Recurrent Neural Network Language Model (Mikolov, Karafiat, Burget, Cernocky, & Khudanpur, 2010) was used as a basis for their approach. A user input and a context (previous input-output pairs) are used as a conditions for the conditional probability of a response. As a training and test data, around 4,000 input-context-output-triplets were extracted from Twitter. In a more recent approach, Xu, Liu, Guo, Sinha, and Akkiraju (2017) trained a long short-term memory (LSTM) network (a special case of recurrent neural network) with around 1 million twitter conversations.

## 3.8 Summary

In this chapter different names and classifications from literature were analyzed in order to find a commonly used terminology. As a result, many different terms were found, where none showed prevailing use. For this reason, *conversational agent* was selected as an umbrella term and defined to limit its scope. Three attributes were listed that define a CA: (1) non-triviality, (2) sequential interaction between user and system, and (3) the use of natural language for input and output. This definition was important for the planning of a prototype (see chapter 5) since it provided clear requirements for the system.

For a more detailed analysis of existing systems, different classifications were described. Due to the large amount of available systems, it is often necessary to further divide the applications into classes according to their attributes. The perspectives provide additional detail when looking at existing systems, as well as when planning a new system.

The history of CA systems started as early as 1950, where Alan Turing described the TT. In the years after, first computer programs (ELIZA, PARRY) were developed that were able

to communicate in natural language. In the the 1990s, the Loebner Prize, a competition implementing the TT, was started. Although, its influence on the development of CA was (and is) part of controversial discussions, it offered a platform for CA technology. Especially, technologies like AIML and ChatScript are strongly tied to this competition. Recently, IPAs made CA systems available to a vast amount of people. Global private companies integrated CAs into their products. The arguably most popular representative of IPAs is Apple's Siri.

In contrast to common rule-based technologies, the availability of huge data sets containing conversations led to increased interest in data-driven approaches for CA development. ML-algorithms are used in this applications to learn from existing data.

A specific field of CAs is described in the next chapter. The definition and perspectives described in this chapter are used to analyze eHealth applications that utilize CAs.

## 4 | Conversational Agents in Health

*This chapter combines the areas of eHealth and conversational agents. Modern applications integrate conversational agents to support certain areas of eHealth. This applications are discussed in the following sections.*

### 4.1 Combining eHealth and Conversational Assistants

Through the growing popularity of CA systems, they were also introduced into the field of health. The two previous chapters discussed the areas of *eHealth* and *conversational agents*. In this chapter the combination of these fields is discussed.

In chapter 2, two key benefits of technology in health were outlined: *interactivity* and *tailoring*. Interactivity, according to Kiousis (2002) relies on the possibility for synchronous and asynchronous communication and "*reciprocal message exchanges*". These exchanges are explicitly mentioned in the definition of a CA in section 3.1 as its *sequential* manner. Also a reference to tailoring can be found in the CA definition: the *non-trivial* behavior means that its reaction is directly based on the input of the user. Thus, its reaction is "*intended to reach one specific person*", what matches with the definition of tailoring as described by Kreuter et al. (1999, p. 277). This alignment in the definitions allows the assumption that CAs are especially suitable for eHealth applications.

Another reason to combine CAs and eHealth was named by Crutzen, Peters, Portugal, Fisser, and Grolleman (2011). They investigated the use of a CA in order to reach a adolescent target group. They concluded that the CA was perceived as faster and more anonymous than information lines and search engines, when accessing health information.

T. Bickmore and Giorgino (2006) saw great potential in CA systems, because of their ability to emulate one-on-one interactions between a health provider and a patient. Furthermore, they also emphasize the strength of CAs when it comes to tailoring. Finally, T. Bickmore and Giorgino stated that time constraints for a patient could be removed, and the general availability of health systems could be increased by CA systems. Especially, in the applications of *health communication* (e.g. Rigg, 2017; Vincent, 2016; T. W. Bickmore, Utami, Matsuyama, & Paasche-Orlow, 2016) and *internet-based interventions* (e.g. A. Watson et al., 2012; Grolleman,

van Dijk, Nijholt, & van Emst, 2006; Lokman & Zain, 2009), CAs are applied. The next sections focus on these two areas and discuss example applications.

## 4.2 Conversational Agents as Information Sources

T. Bickmore and Giorgino (2006) described one-on-one communication as the gold standard for patient interaction. In reality this type of interaction is heavily time restricted. This can cause patients to be afraid of asking a health professional for information. Another problem described by T. Bickmore and Giorgino was that many people not even have the access to health professionals. CA applications can be utilized to increase the accessibility of health information.

One concrete example for a system of this kind was tested by the United Kingdom's National Health service. The goal of the provided CA is to extend a current telephone health service and increase availability. When communicating with the CA, the problem of the user is evaluated and further steps are proposed (Vincent, 2017; Rigg, 2017). A similar system is developed by Chinese search engine Baidu. It uses deep learning algorithms to provide diagnoses from the data retrieved while interacting with users (Vincent, 2016).

A CA as interface for health information is also provided within the HealthVault project of Microsoft<sup>1</sup>. The HealthVault project is a comprehensive collection of technology powered health applications and also provides an EHR system. For accessing the stored patient information or a diagnostic symptom checker a CA is used. CA systems to provide health information are also provided by applications like Dr AI<sup>2</sup>, Florence<sup>3</sup>, medwhat<sup>4</sup>, or Your.MD<sup>5</sup>. These are only a few examples of projects that are available on the Internet. It is important to note that these systems are driven by private companies and information about effectiveness is not available.

## 4.3 Conversational Agents for Behavior Change

CA systems for behavior change can also be seen as an instance of a Persuasive System which is defined as *"a computerized software or information system designed to reinforce, change or shape attitudes or behaviours or both without using coercion or deception"* (Oinas-Kukkonen & Harjumaa, 2008, p. 202).

For the implementation of Persuasive Systems, Oinas-Kukkonen and Harjumaa (2009) suggest the Persuasive Systems Design (PSD) Model. Especially the implementation of, what they refer to as Dialog Support, can be seen as a strength of CAs: Actions like praising, reminding

<sup>1</sup><https://www.healthvault.com/en-us/health-bot/>, Retrieved 15/05/2017

<sup>2</sup><https://medium.com/@HealthTap/dr-a-i-80b4cf06be30>, Retrieved 15/05/2017

<sup>3</sup><https://florence.chat>, Retrieved 15/05/2017

<sup>4</sup><https://medwhat.com>, Retrieved 15/05/2017

<sup>5</sup><https://www.your.md/>, Retrieved 15/05/2017

the user of their target, or adopting a social role are tasks that can be implemented in a CA. In a study by Lehto and Oinas-Kukkonen evaluating different Persuasive Systems they remark that *"dialogue support was rather weak overall"* (Lehto & Oinas-Kukkonen, 2015, p. 131), which suggests potential for the application of CA systems.

Andrews et al. (2006) point out that it is important to keep a balance between a controlled and open conversation flow. A too restricted system would not show enough empathy for being able to persuade a user. Too open systems have the problem of being too less (or not at all) focused on its goal: to convince the person. Furthermore, it is suggested for the system to take the initiative in the conversation. Two layers are proposed to be used in such systems: (1) a planning layer which is responsible for consistency and goal-orientation of the system and (2) the interaction layer that is used to create the actual interaction using CA techniques.

A. Watson et al. (2012) developed a CA to support physical activity among overweight adults. The CA was designed to communicate with their users in verbal and non-verbal form. This communication followed a tailored conversation path, and restricted the input to a set of natural-language options. The participants of the study were asked to communicate with the CA three times a week; those conversations lasted between 5 and 10 minutes on average. As a result, A. Watson et al. concluded that the CA did not increase the amount of activity between start and end of the study, but it motivated the participants to stay on the same activity level, whereas the control group reduced their physical activity throughout the study.

A CA system to promote a healthy lifestyle was described by Gardiner et al. (2017). The participants of a study were asked to communicate with the CA on a daily basis, and talk about health food, diabetes prevention, or stress management. The control group received the same information that was provided by the CA as information sheets. As a result, users of the CA system showed a significant decrease in the use of alcohol for stress management. Furthermore, the fruit consumption of those participants who used the CA increased significantly.

## 4.4 Summary

In this chapter the areas of eHealth and CAs were combined. CA systems can make further use of the potential of eHealth, especially interactivity and user specific tailoring can be implemented using a CA. These strengths are also emphasized when planning the prototype in chapter 5.

The two main application areas of CAs in health are health communication and internet-based interventions. Health information systems are of strong interest for private companies, and many different applications can be found in this sector; their effectiveness will need be proved in future. CA systems focusing on behavior change are more commonly described and tested in scientific literature. Here it has been shown that a CA can positively influence a persons lifestyle.



## 5 | System Architecture

*This chapter shows the process of designing a CA for the Self eHealth platform. Starting with the description of the idea, it continues with the analysis of the existing system to extract requirements. Three alternative concepts are discussed before selecting one. In the following sections, this prototype is planned and different decisions are documented.*

### 5.1 System Goals

The Self eHealth platform offers its users various features around personal health. These features are accessible on a web site as well as in mobile apps.

As an alternative way to interact with the platform, this thesis examines the integration of a CA. This CA should communicate with the user in order to support the provided functionality of the system, and improve the user experience through personalized messages; the CA should further increase the positive influence the platform has on the health behavior of the user. No preliminary requirements were available for this task; the selection of an application area was completely unrestricted. For this reason, a detailed analysis of the Self eHealth platform was necessary to derive possible concepts that could be implemented as a prototype.

Requirements should be extracted as part of the analysis process; these requirements should be extended after selecting a concrete concept for a CA. A prototype should be designed around these requirements. Additionally, technological and design specific decisions needed to be discussed. This prototype should then be implemented (see chapter 6) and tested in a study (see chapter 7) to examine its acceptance and impact on the behavior of the users.

### 5.2 Analysis of the Self eHealth Platform

For the analysis three main perspectives were selected in order to understand the underlying system: (1) functionality, (2) usage, and (3) technology. The functionality perspective covered the general features of the platform. It looked on what functions were offered to the user and

how these were accessible. How those provided features were actually accessed was assessed by the usage perspective. This included questions like "Is there a consistent user base?", and "Are there features that are used more than others?". Finally, to plan the prototype in a way it can be integrated into the existing system as optimal as possible, the technological point of view looked at frameworks, programming languages and libraries that were utilized in the existing project.

As described by Glance et al. (2016), the platform was created in order to support an activity challenge, where *"participants were asked to walk, run or engage in other exercise to achieve an equivalent of 10,000 steps a day"* (Glance et al., 2016, p. 38). As a basic use case, a person would regularly synchronize a fitness tracker with the platform in order to add an activity entry to the record. The user would then be provided with performance statistics and comparisons to other participants of the challenge. Participants of an activity challenge were also able to form groups and compete with other teams. These activity challenges were designed to run for a limited time period. During this time, additional benefits, like organized events, were provided. To participate in a challenge, an enrollment fee between AUD 10 and AUD 25 was charged per attendant.

## 5.2.1 Functionality

Apart from the use case explained above, many other features were provided. Figure 5.1 shows how the functions of Self eHealth were categorized by Glance et al. (2016). The platform included the functionality to coordinate research projects<sup>1</sup>. For this purpose, users could sign up to an available project from within the application. Participants were then offered further information about the project and access to assessment questionnaires on the platform.

Several different fitness tracking devices could be connected to the platform in order to automatically collect data about their daily activities<sup>2</sup>. This data was then displayed using various forms of graphical representation. Furthermore, personal health data, such as weight, blood pressure or cholesterol values, could be added to the system. To motivate the participants of a challenge, gamification elements were integrated. Examples for such elements are leaderboards, badges, and prizes. The users were also able to see social media entries in the application (Glance et al., 2016).

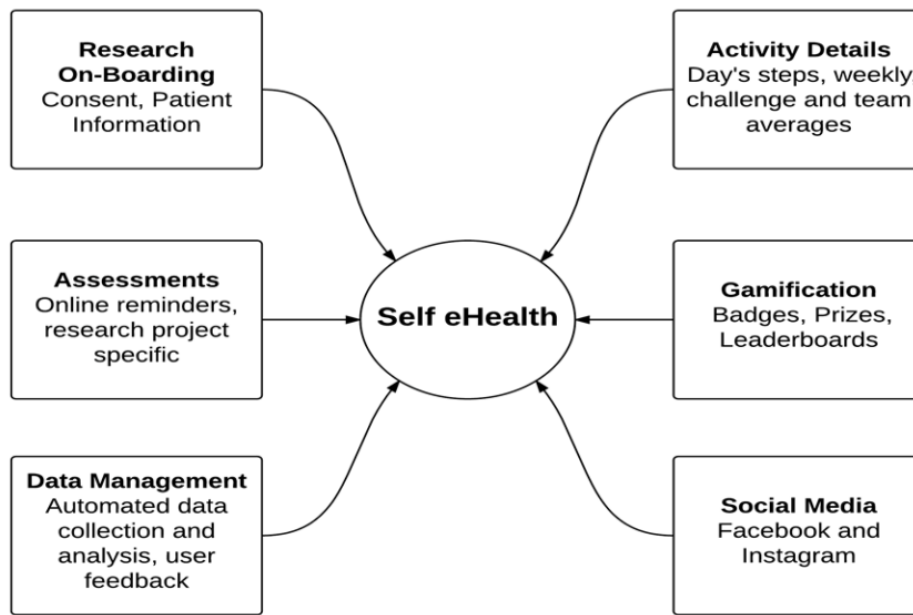
The Self eHealth platform was accessible to the user as a web and a mobile application. Both versions supported the features described above, but they were structured in a different way. Especially, the features on the the overview pages (see figures 5.2 and 5.3a for the web and the

---

<sup>1</sup>One example was the "Gaming for Exercise - Pokemon Go Challenge Research Project", where people used the Self eHealth platform for tracking their steps while playing Pokemon Go. Additionally to tracking the steps of the users, the platform also offered the participants the possibility to review the project description, the consent form, and access questionnaires that they were asked to fill out.

<sup>2</sup>Supported where devices for/from: Fitbit, Jawbone, Microsoft Health, Misfit, Google Fit, Garmin, and Apple Inc.





**Figure 5.1: Functions of the Self eHealth platform as described by Glance et al. (2016)**

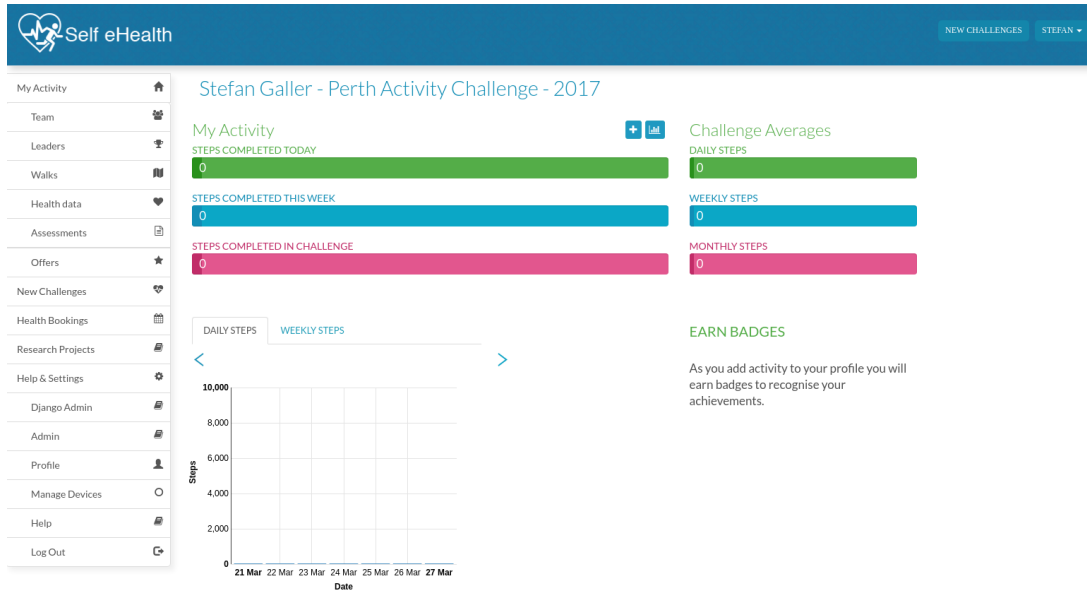
mobile version respectively) differed considerably: Where both versions provided statistics on the number of steps walked, the mobile app also allowed the user to add minutes of mindfulness, portions of fruit and vegetables, and sleep quality. These are features that were not included in the web version at all. Also, a food diary (see figure 5.3c) which allowed the user to keep track of meals was only included in the mobile version.

Users of the platform could also book appointments with health practitioners. They could select from various offers and make an appointment directly from within the website or the mobile app.

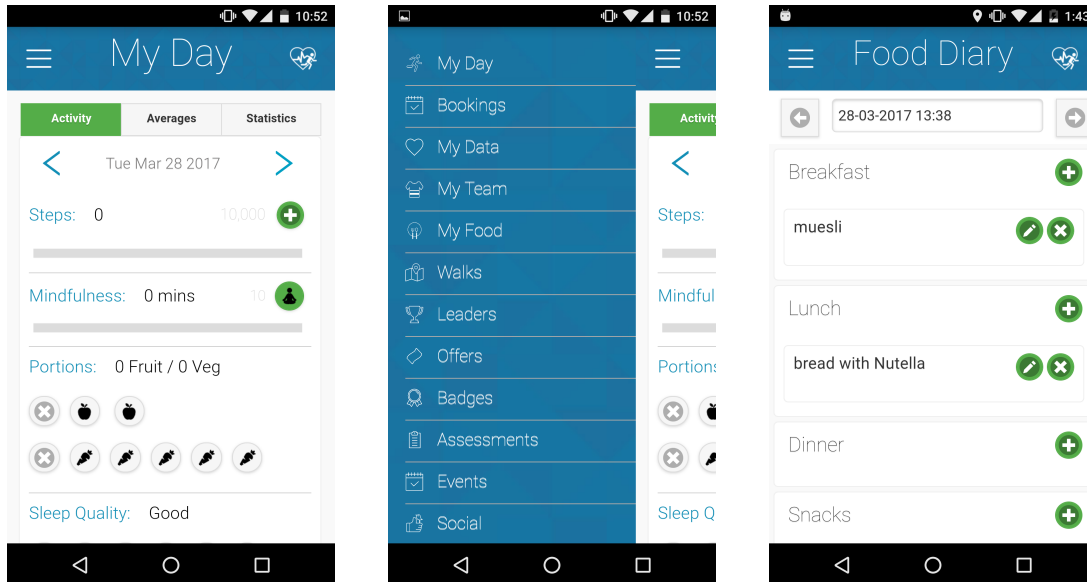
For administration purposes the platform provided two areas for managing the system. One was designed for general functions such as user management and basic statistics, and the second area was restricted to the technical staff and offered access to all stored data.

## 5.2.2 Usage

Since the platform was developed especially to support participants of activity challenges, during these challenges the system was used most frequently. As figure 5.4 shows, the number of active users (participants who regularly add activities) strongly increased at the begin of those challenges, slightly decreased during the challenges, and dropped significantly after the challenges finished. Following events are marked in the figure: (1) the UWA Self eHealth Challenge, held from May till September 2015, (2) the Perth Activity challenge 2016 which was partly held simultaneously with (3) the UWA Activity Challenge 2016. The numbers at period (4) were distorted by an error, where certain data has been lost.



**Figure 5.2:** The web version of the Self eHealth Platform when logged in. It offers statistics about the current activity challenge (center), and a menu for selecting the available functionality (left).

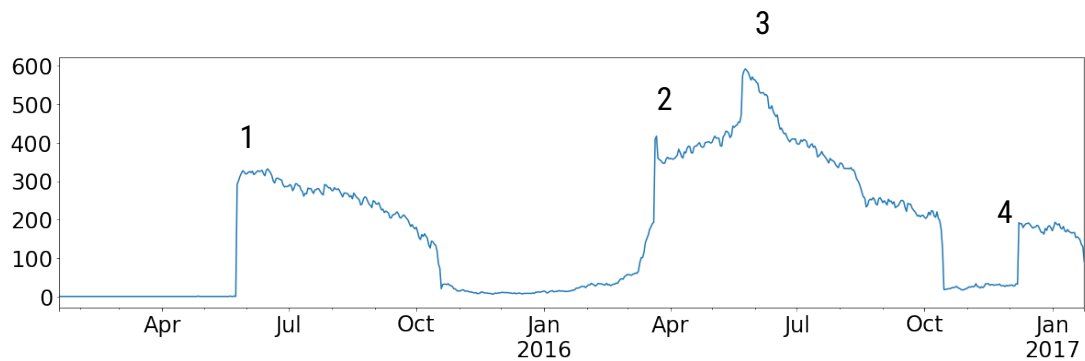


**(a) "My Day" - The overview page of the mobile application.**

**(b) The menu of the mobile application, offering access to the different features of the platform.**

**(c) The food diary feature of the mobile application. It allows to manage entries for the different meals of the day.**

**Figure 5.3:** Different screens of the Self eHealth mobile application for Android and iOS.



**Figure 5.4: Active users on the Self eHealth platform between 2015 to 2017.**

Between April 2016 and January 2017 around 330 practitioner sessions were booked over the platform, although it has to be noted that around 50 per cent of those bookings were made as part of a research project<sup>3</sup>. The food diary was rarely used: From March to December 2016 less than 100 entries were made by all registered users. Compared to the almost 600 active participants in May 2016, this feature was used by hardly any of the participants.

### 5.2.3 Technology

The platform was build in a server-client architecture. The server application was developed using the Python programming language and the Django<sup>4</sup> framework. This web application offered a Representational State Transfer (REST) interface for accessing different functions. The web interface and the mobile applications were using this REST interface to send and retrieve data to/from the server. This architecture allowed to reuse functionality independent of the UI implementation. The web application was hosted on an external Amazon Web Services (AWS)<sup>5</sup> web server, using a Linux operating system.

Figure 5.5 shows a basic representation of the architecture, as it could be analyzed from the existing source code. The different components of the web application are implemented as modules in the Django framework. Example modules are the *webapp* module, that is responsible for the web browser version of the platform, the *mobile\_api* module, handling the mobile version of the platform, and the *diary* module, containing the code responsible for the food diary. Each module contains different models; these are data objects that are mapped to the database using the object-relational mapping (ORM) layer of Django. The different view components of a module are invoked by the Uniform Resource Locator (URL) dispatcher: When a specific URL of the REST interface is contacted, the URL dispatcher forwards the

<sup>3</sup>In this research project, the participants were checked by a practitioner on several instances during the study to observe changes in health values. The participants were able to book these appointments using the Self eHealth platform.

<sup>4</sup><https://www.djangoproject.com/>, Retrieved 28/3/2017

<sup>5</sup><https://aws.amazon.com/>, Retrieved 28/3/2017

request to the responsible view. This view then uses different program logic to generate a response.

Since the Django framework uses Python as a programming language, it is compatible with all available Python libraries. The Self eHealth web application utilizes various external libraries. The Django framework centralizes its configuration in a global configuration file. In this file – among others – the modules are registered, database access is configured, and also custom configuration parameters can be included.

The web version uses a combination HyperText Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript for providing the UI. Calls to the server were made using asynchronous JavaScript and XML (AJAX). For the mobile applications the same technologies were used. Mobile applications were available for Android and iOS. Both applications used a native code wrapper to include the HTML, CSS and JavaScript interface. The implementation of the web and the mobile apps were different, but most of the code was shared between the mobile apps.

### 5.3 General Requirements

After the analysis of the platform, general requirements were extracted. These requirements were valid, independent from the actual behavior of the CA. More detailed behavioral requirements are discussed in section 5.4.

The prototype was required to offer a way for the user to enter text input in natural language. This input should then be analyzed to understand its meaning. According to the meaning, an output in natural language should be generated that can be seen as a suitable response to the input. Suitable describes the syntactical (it should be grammatically correct), as well as the semantic (the output should make sense for the corresponding input) level. The prototype should enable the user to interact with the system using this described procedure. For the implementation of the message understanding and generation, one of the available frameworks should be used.

To limit the scope of the project, the focus should be on the mobile application of the platform. It was assumed that the user behavior would differ between the web and the mobile application, and thus would not be comparable in a study. For a prototype, one UI was seen as sufficient.

The technology as described in the previous section should be reused where possible. This means that for the functionality of the prototype should be implemented using Python and the Django framework, except for what can be handled by the selected CA framework. For the UI that means the use of HTML, CSS, and JavaScript. For a seamless integration, also the design of the UI should be done in a consistent manner. That especially includes the reuse of existing CSS styles and HTML classes.

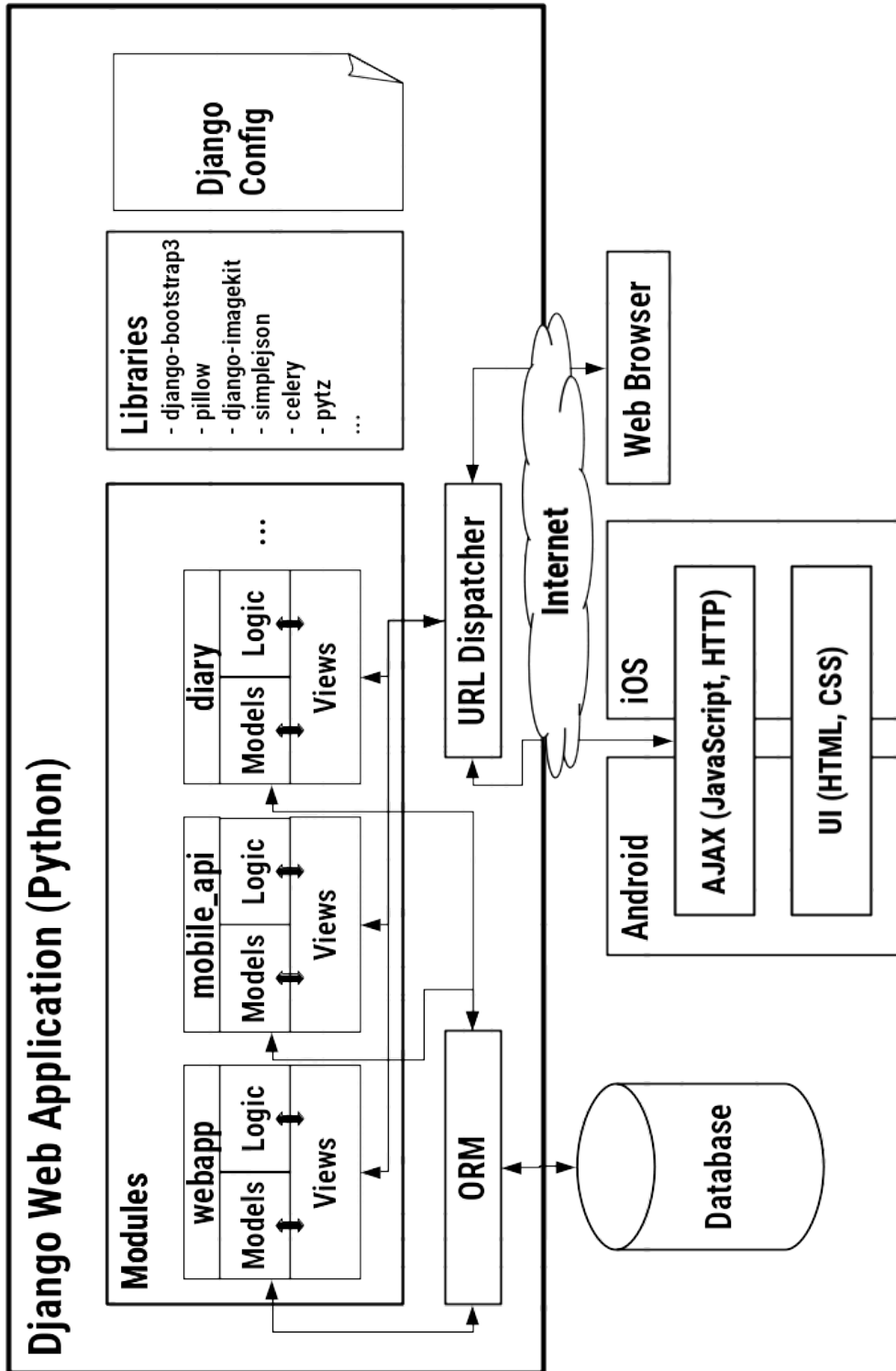


Figure 5.5: The structure of the Self eHealth platform from a technology perspective, as derived from the available source code.

## 5.4 Proposal of Alternative Concepts

With respect to the analysis of the platform, the extracted requirements, and the different types of CAs discussed in previous chapters, three alternative concepts were elaborated. These concepts are proposed and discussed below; one concept was then selected to serve as the foundation for the prototype.

### 5.4.1 Concept Proposal 1: Health Assistant

To develop a health assistant, an IPA for the Self eHealth platform, was the first idea evaluated after the analysis. This CA was thought to support a user while using the platform and offer an additional way to access the features. Furthermore, the health assistant should be able to answer user questions associated to their health and well-being.

The idea for the health assistant (see section 5.4.1) was dismissed, after estimating the needed effort. Although the platform has a limited domain, the number of features (see section 5.2.1) would require a large set user input that needed to be understood, and thus exceed the scope of this thesis. The discussion of this idea led to the insight that for a prototype, the scope of the CA needed to be strongly limited. This raised the question, how the domain could be limited without restricting the possible input by the user.

One idea was to concentrate on the most common questions asked by the users. The problem with this approach was that this would have needed to perform an extensive user study before further planning. This was seen as too time consuming and thus was not further pursued. As a solution for this problem, it was decided to create a proactive CA that limits the possible user input by asking concrete questions.

### 5.4.2 Concept Proposal 2: Motivation Diagnosis

The next proposed concept was that of a CA for motivation diagnosis. Motivation plays an important role in following a healthy lifestyle. Thus, monitoring the motivational level of the users of the Self eHealth platform would allow the system to react when changes appear. A CA was seen as a possible tool to approach the users and evaluate their motivational state. To evaluate this idea, ways to measure motivation of people were researched. A common approach is the use of self assessment questionnaires (e.g. Tremblay, Blanchard, Taylor, Pelletier, & Villeneuve, 2009; Guay, Vallerand, & Blanchard, 2000).

The self assessment questionnaires had the disadvantage that they commonly consist of more than 15 questions and expect answers on a Likert scale. To include them into a regularly occurring conversation would lead to conversations that were considered as too long and monotonous. Nevertheless, the idea of self evaluation was seen as valuable information that could be retrieved from the user. This led to the next concept proposal.

### 5.4.3 Concept Proposal 3: Motivational Conversational Agent

Instead of just retrieving simple self evaluation, a CA could be used to motivate the user to increase the number of health data added to the system. Additionally, this CA could use the collected data to provide motivational feedback for the user and so extend the motivational features already included in the platform. This concept was seen as suitable to be implemented as a prototype, regarding content as well as scope.

## 5.5 Prototype Concept

Based on the idea of a motivational CA, a concrete behavioral model was designed. The CA should motivate the user by giving tailored feedback based on user data. To ensure that enough data was available, the collection of data was added as an additional behavioral requirement for the prototype. In other words, the CA should not only motivate the user to participate in a healthy lifestyle, but also to regularly add health information to the system.

As described in section 5.2.2, especially the food diary offered on the platform was hardly used. To increase the number of diary entries made by the user, the data acquisition should be built around this platform feature: The CA should regularly interview the users about their eating behavior. With perspective on the existing categories in the food diary, data for *breakfast*, *lunch*, and *dinner* should be collected. Additionally, data that is *available on a regular basis* and *does not allow automatic collection* should be acquired by the CA. Table 5.1 shows the different types of data relevant for collection. This completes collected information with the hours and quality of sleep every night, number of steps (in case they could not be synchronized with the platform automatically), and regular questions about how the user feels.

<b>diet</b>	breakfast, lunch, dinner
<b>sleep</b>	hours, quality
<b>activities</b>	number of steps
<b>self-evaluation</b>	feeling of the user

**Table 5.1: Types of data to be acquired by the CA**

When interviewing the user, the CA should proactively approach the user. To limit the number of questions asked by the CA every time the user is contacted, every type of data is assigned to one of three topics: breakfast, lunch, and dinner. Depending on the time of the day, the CA should choose a different topic to talk about. In figure 5.6 an example configuration is shown. Each topic is active for a specific time span, and contains a set of data types that should be collected in this conversation. After the CA has collected all the needed data, it should close the conversation. If the user approaches the CA when no topic is active, it was decided that for the prototype the CA should ask the user to wait for further instructions.

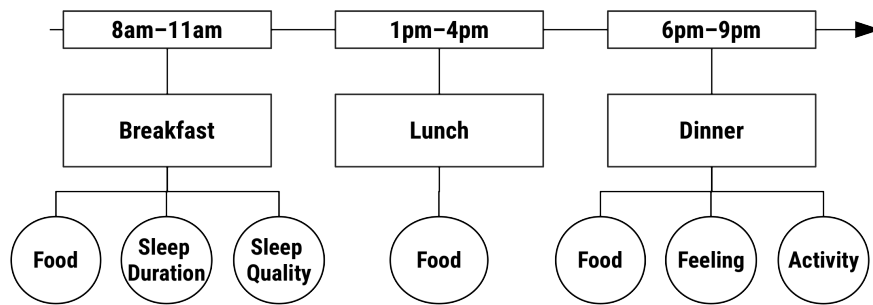


Figure 5.6: Example for the acquisition timeline.

As a second facet of the behavior, it was planned to regularly present motivational feedback to the user, generated from the available data. Within a conversation, the CA should make statements that are intended to increase the motivation of the user. Table 5.2 shows the types of feedback that should be supported. Depending on the type, the feedback should be given on a specific day only (e.g. end of the week summary), or every time the event occurs (n-day streak). The feedback messages should be designed in a way to praise the users achievement ("keep up the good work"), rather than just presenting the facts.

<b>n-day streak</b>	achieving 10,000 steps on several days in a row
<b>end of the week summary</b>	better performance than the week before
<b>better than n%</b>	more steps than most of the users
<b>healthy diet</b>	eating more than 1 portion of fruit and 4 portions of vegetables
<b>increased # of fruit</b>	eating more fruit than before
<b>increased # of vegetables</b>	eating more vegetables than before

Table 5.2: Types of feedback that should be given by the CA

To fit to the concept of a messaging application, the language of the CA should be adjusted accordingly. That means that the language used should be rather colloquial than formal. Emoticons should be used to underline the colloquial style of conversation. Furthermore, it was decided that the interaction between the user and the CA should solely rely on natural language. This opposes applications where the user input is limited by offering predefined answers that can be selected by clicking on buttons. When the user first interacts with the CA, a short introductory message should be sent.

In conclusion, the requirements in section 5.3 were extended by three additional requirements: (1) acquiring data, (2) giving feedback, and (3) doing that in a colloquial style.

## 5.6 Technology Decisions

To implement the designated behavior discussed in the previous section, technological decisions needed to be made. Especially, a CA framework that was suitable to support the required



behavior had to be selected. Other decisions that are discussed in this section are those about the logic and communication components of the implementation, and the technology used for the UI.

### 5.6.1 Conversational Agent Frameworks

For the creation of CAs, several frameworks were available. As a first step, four possible candidates were selected: (1) AIML, (2) ChatScript, (3) Wit.ai, and (4) IBM Watson Conversation. To find the most appropriate one, it was necessary to identify requirements for the prototype and evaluate the frameworks accordingly.

As a primary requirement, it was specified that the framework needed to support the behavior as described in section 5.5. Second, the framework should require as few additional dependencies as possible, and the integration to the existing system should be kept in mind during the selection process. Furthermore, the availability of comprehensive documentation was seen as a positive influence factor for framework evaluation. Since there were no financial resources available for this project, the cost factor was considered as a crucial point. Open Source Software was considered a plus, but not as absolute requirement. Another factor was the availability and accessibility of the framework; if hosted on an external server, it needed to be considered how the service could be accessed, what would happen in case of unavailability, and what kind of privacy standards were applied to the user data. The requirements are summarized in table 5.3.

- |                    |                 |                |           |
|--------------------|-----------------|----------------|-----------|
| ■ behavior support | ■ documentation | ■ hosting      | ■ privacy |
| ■ integration      | ■ cost          | ■ availability |           |

**Table 5.3: Evaluation factors for selecting a CA framework**

#### AIML

AIML (see also section 3.5.2) is a language for creating XML-based input-output rules for CAs. Different interpreters are available for free in various programming languages<sup>6</sup>; among others in Python which could be used to directly integrate the message generation into the Django web application. Additionally, AIML can also be hosted on Pandorabots<sup>7</sup>, which is, however, not available for free.

Although its simple functionality, the rules allow a very wide range of configuration as shown with the implementation of Mitsuku, two times Loebner Prize winner. Nevertheless, the

<sup>6</sup><http://www.alicebot.org/downloads/programs.html>, Retrieved 30/3/2017

<sup>7</sup><https://www.pandorabots.com/>, Retrieved 30/3/2017

number of rules needed even for a simple conversation are relatively high. AIML provides a comprehensive documentation that is available online<sup>8</sup>.

## ChatScript

As mentioned in section 3.5.3, ChatScript was thought to improve on the functionality of AIML. It uses a more advanced approach of pattern matching, and also allows to have more influence on the logic of selecting the right response. Furthermore, it already implements the functionality of proactively approaching the user with a certain topic. The advanced functionality increases the complexity of single rules, but decreases the number of needed configurations when compared to AIML.

Chatscript is available for free and as Open Source Software<sup>9</sup>. A very detailed documentation is included in the project. There is no external hosting available; the only way to run ChatScript is locally as a separate server process.

## Wit.ai

Wit.ai<sup>10</sup> is an online service that allows developers to craft CA conversations. It uses advanced machine learning algorithms that are trained with data available from all CAs that are hosted on their servers. This could also include data from user conversations and might be a privacy issue. The service allows a developer to set up basic rules using a web interface (see figure 5.7). Alternating, *user says* and *bot sends* blocks can be added to craft a conversation. The blocks can also include abstract entities such as the intent; these entities are used by the underlying ML algorithm to classify the rules. Although the web interface is relatively intuitive to use, it limits the configurability of the rules; so it is not possible let the CA start a conversation.

Wit.ai is available for free and offers several types of development kits for multiple languages (including Python) to access the service. How the logic for selecting the right response work is not visible to the developer. This might lead to difficulties when debugging a conversation. The framework is well documented<sup>11</sup>.

## IBM Watson Conversation

IBM Watson Conversation<sup>12</sup> allows developers to setup up a CA over a web interface. Similar to Wit.ai it provides highly advanced natural language processing (NLP) algorithms for processing the utterances of the user. The use of a web interface also limits the functionality

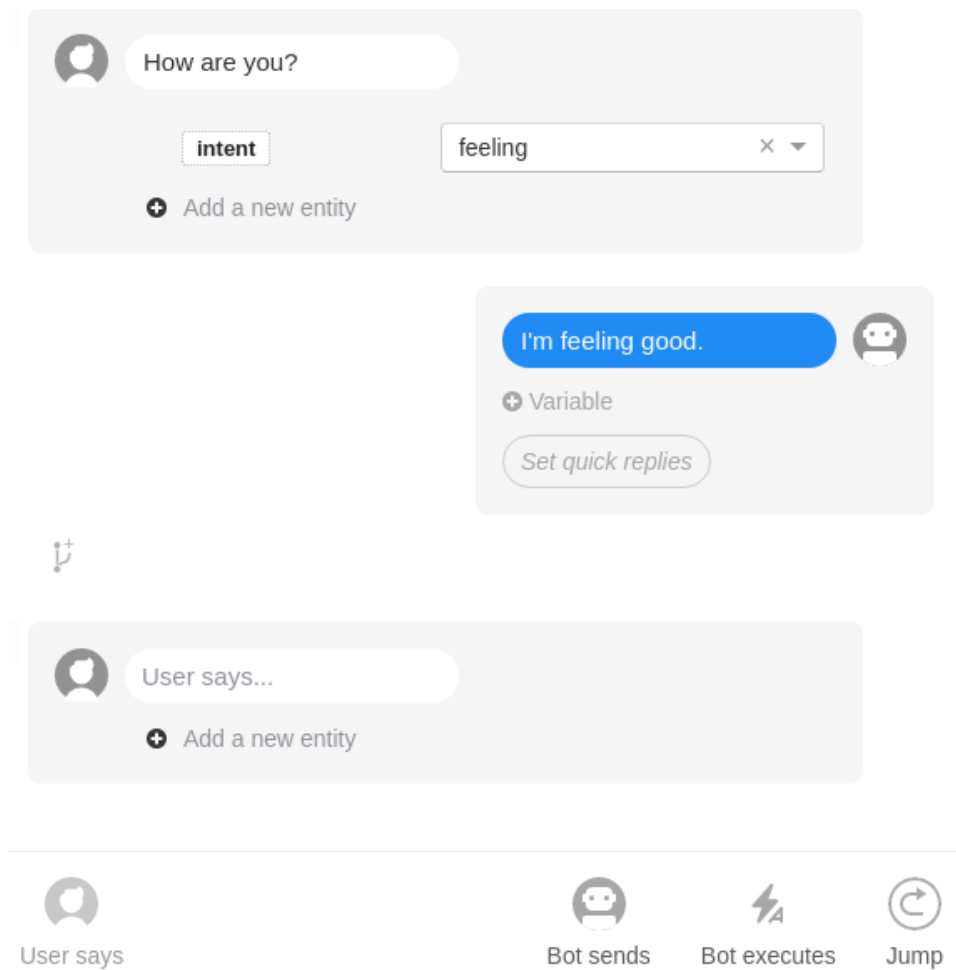
<sup>8</sup><http://www.alicebot.org/documentation/>, Retrieved 30/3/2017

<sup>9</sup><https://github.com/bwilcox-1234/ChatScript>, Retrieved 30/3/2017

<sup>10</sup><https://wit.ai/>, Retrieved 30/3/2017

<sup>11</sup><https://wit.ai/docs>, Retrieved 20/06/2017

<sup>12</sup><https://www.ibm.com/watson/developercloud/conversation.html>, Retrieved 30/3/2017



**Figure 5.7:** The user interface offered by wit.ai to configure the CA. Conversations are crafted by adding expected user input (left - "How are you?"), and required CA output (right - "I'm feeling good.") alternately.

of the framework, and does not allow the CA to proactively begin a conversation. A version with limited functionality is available for free, but there are also paid plans available. A development kit for Python is available, as well as a comprehensive documentation<sup>13</sup>.

## Discussion and Decision

All four described frameworks have their strengths and weaknesses. Table 5.4 offers a comparison of the different arguments. The limitation to a web-based editor for configuration was the main argument for excluding Wit.ai and IBM Watson Conversation. Both frameworks were easy to use for their purpose of building a passive IPA, but did not allow further configuration. Both, AIML and ChatScript allow to be used as a standalone service, and were also available for free. The arguments for using *ChatScript* instead of AIML as the preferred framework were that ChatScript allows the CA to act proactively out of the box, as well as the advanced pattern matching. The fact that ChatScript needed to be hosted locally showed to be no problem as the framework was delivered with a binary file that was able to run on the production as well as on the development machines without further configuration.

Framework	Pro	Con
AIML	free, open source, easy to use, interpreters for many languages, good documentation	relatively weak pattern matching, requires a lot of rules
ChatScript	free, open source, very flexible, allows proactive topics, advanced pattern matching, good documentation	needs separate server process, no direct Python integration, can become complex
Wit.ai	free, advanced NLP algorithms, good documentation, development kit for Python	closed source, only web based rule crafting, "black box"
IBM Watson Conversation	advanced NLP algorithms, good documentation, development kit for Python	closed source, only free for very limited version, only web based rule crafting, "black box"

**Table 5.4: Comparison of different CA frameworks.**

## 5.6.2 Backend and Logic

To keep the integration of the developed CA as simple as possible, it was decided that the technology used in the Self eHealth platform should be reused where possible. Therefore, the server-side logic of the CA should be implemented using Python as a programming language,

<sup>13</sup><https://www.ibm.com/watson/developercloud/doc/conversation/index.html>, Retrieved 20/06/2017

and Django as the underlying server framework. The network communication of the Self eHealth platform was made over HyperText Transfer Protocol (HTTP) connections using a REST interface.

### Django Web App

The Django web app was responsible for the logical operations that could be made on the server side. Therefore, also operations that cannot be computed within the ChatScript framework should be performed there. Django applications are organized in modules; for the integration of a CA, a new module should be implemented containing all the necessary functionality. This module should contain the logic for selecting the right topic, extracting feedback information, and offering access to the database. Necessary database operations are abstracted by Django; ORM is used to persist data objects to the database. Objects containing information relevant to the CA should also be stored using this integrated feature.

For additional functionality of the server-side application, various libraries are available for the Python programming language. For the implementation of the prototype the natural language toolkit (NLTK)<sup>14</sup> and the pandas<sup>15</sup> libraries should be added to the web application.

### HTTP and REST

To communicate with the web application, Django offers a REST interface. That means that different resources of the web server are linked to a certain URL and can then be modified by operations like GET, to retrieve information, or POST, to push information to the server. As an example, a GET request to the URL `home/` on the Self eHealth server returned an HTML page containing the start screen of the web application.

For the communication between the Django web application and the ChatScript service, two types of communication channels needed to be specified. First, a REST interface was defined that allows the Chatscript server to retrieve and push information to the web application. For security purposes, this interface should not be accessible to any other process than the ChatScript server. Second, for retrieving a reply for a user input, ChatScript offers a custom socket protocol. It accepts a input containing a message, and returns a reply to the received message. This socket communication should be implemented within the web application. To allow access to the CA from a mobile device, the REST interface for mobile devices should be extended by the necessary functionality.

---

<sup>14</sup><http://www.nltk.org/>, Retrieved 18/07/2017

<sup>15</sup><http://pandas.pydata.org/>, Retrieved 23/06/2017

Component	Technologies
Message Generation	ChatScript
Server Application	Django (Python)
Communication	HTTP, REST, AJAX, Sockets
User Interface	HTML, CSS, JavaScript
Mobile Native Apps	Android (Java), iOS (Objective C / Swift)

**Table 5.5: Used technologies as planned for the prototype.**

### 5.6.3 Frontend

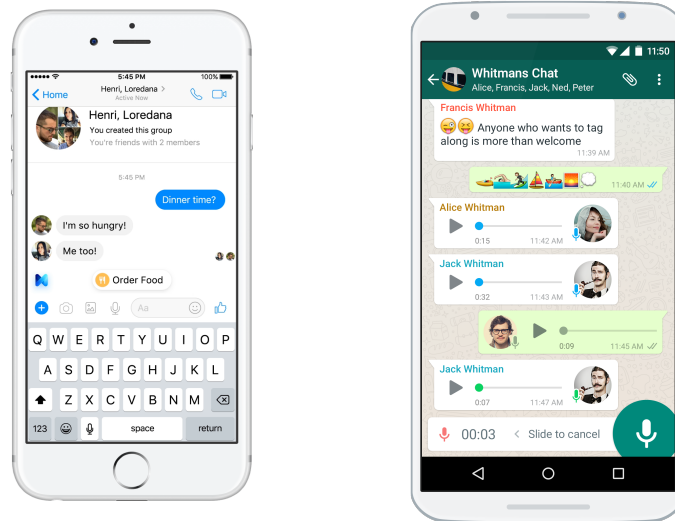
The CA should be offered to the user as part of the mobile application. This mobile application was based on HTML, CSS, and JavaScript, and presented in a native application container for Android and iOS. Therefore, the implementation of the UI should also be done using this technologies. HTML and CSS should be used to define the look of the interface that should resemble a modern chat application. JavaScript should be used for communication with the server to retrieve and send data. This communication should be done asynchronously using AJAX to offer a responsive UI, also while retrieving or sending data.

### 5.6.4 Technology Used

Different technologies were selected for the prototype. A summary is presented in table 5.5. For the message generation, ChatScript should be used as an CA framework. Technologies around server application, communication and UI should be reused from the existing implementation where possible. That meant the use of the Django framework and Python for the application backend, communication over HTTP and REST interfaces, and the implementation of the UI with HTML, CSS, and JavaScript. For the mobile applications, Java (Android), Swift and Objective C (both iOS) were used as programming languages.

## 5.7 Integration and Usability

Besides the behavioral and technological decisions, it was also important to decide how the CA was presented to the user. Since, the user should frequently use the CA it was crucial to provide an appealing UI and to motivate the user to access the application. Furthermore, the selection of embodiment and the design of the conversation style had to be planned.



(a) Facebook Messenger

(b) WhatsApp

Figure 5.8: Interfaces of common messenger applications.

### 5.7.1 User Interface Design

In order to keep the time to familiarize with the UI as short as possible, it was decided to use a classic messaging interface that resembles popular applications like Facebook Messenger (see figure 5.8a) or WhatsApp (see figure 5.8b). It was assumed, that this kind of UI is known to many of the possible users and thus feels comfortable from the first use. Also, the notifications should be similar to common messaging applications. When a message from the CA is received, the user should be informed by a push-notification and should be able to directly interact with the CA. Additionally, the CA should also be accessible from within the mobile application. Therefore, the CA UI should be linked from a prominent position in the *My Day* screen of the app. To fit well into the existing app, a look and feel should be used, that matches the design of the Self eHealth mobile application.

### 5.7.2 Embodiment

As explained in the section above, the UI should resemble a classic chat application. In those applications, users usually upload an image that is displayed next to their messages. For this reason, an avatar for the application should be developed. Like in a chat application, a static image was decided to be sufficient for this purpose. This image should globally represent the CA and should not allow configuration by the user. The image representing the user should be reused from the Self eHealth platform, where users can upload a custom image. Speech synthesization is typically not part of a chat application, and thus should not be implemented.

## 5.8 Architecture

The updated architecture of the Self eHealth platform – including the CA – is represented in figure 5.9. ChatScript was introduced as an additional server process, communicating with the Django web application in twofold manner. For retrieving responses from ChatScript, the web application needs to establish a socket connection. Communication regarding the retrieval and storage of data is done via a specifically tailored REST interface.

Within a Django application, different parts are organized as modules. For the prototype a module should be implemented, containing all the functionality that is necessary for offering the desired behavior. The new module needs to be added to the Django configuration file. New URLs should be specified that are linked to functionality of the the new module. Additionally, the *mobile\_api* module should be extended to offer the CA communication to the mobile application. For the CA functionality the NLTK and pandas libraries should be included.

The mobile application needed to extended by a chat interface, including additional communication functionality using AJAX.

## 5.9 Summary

This chapter described the planning process to include a CA into the Self eHealth platform. A motivational CA was planned that should proactively approach the user in order to retrieve information, and should regularly send motivational feedback messages.

It was decided that the prototype should reuse the technology of the platform where possible. This means, that for the web application the Django framework should be used. The UI should be implemented using HTML, CSS, and JavaScript, and should resemble a common messenger application. For the message generation, ChatScript was selected as the underlying framework. In the application, the CA should be embodied by a simple graphic, and communicate in colloquial style and solely in natural language.

The next chapter describes the implementation of the prototype planned above.



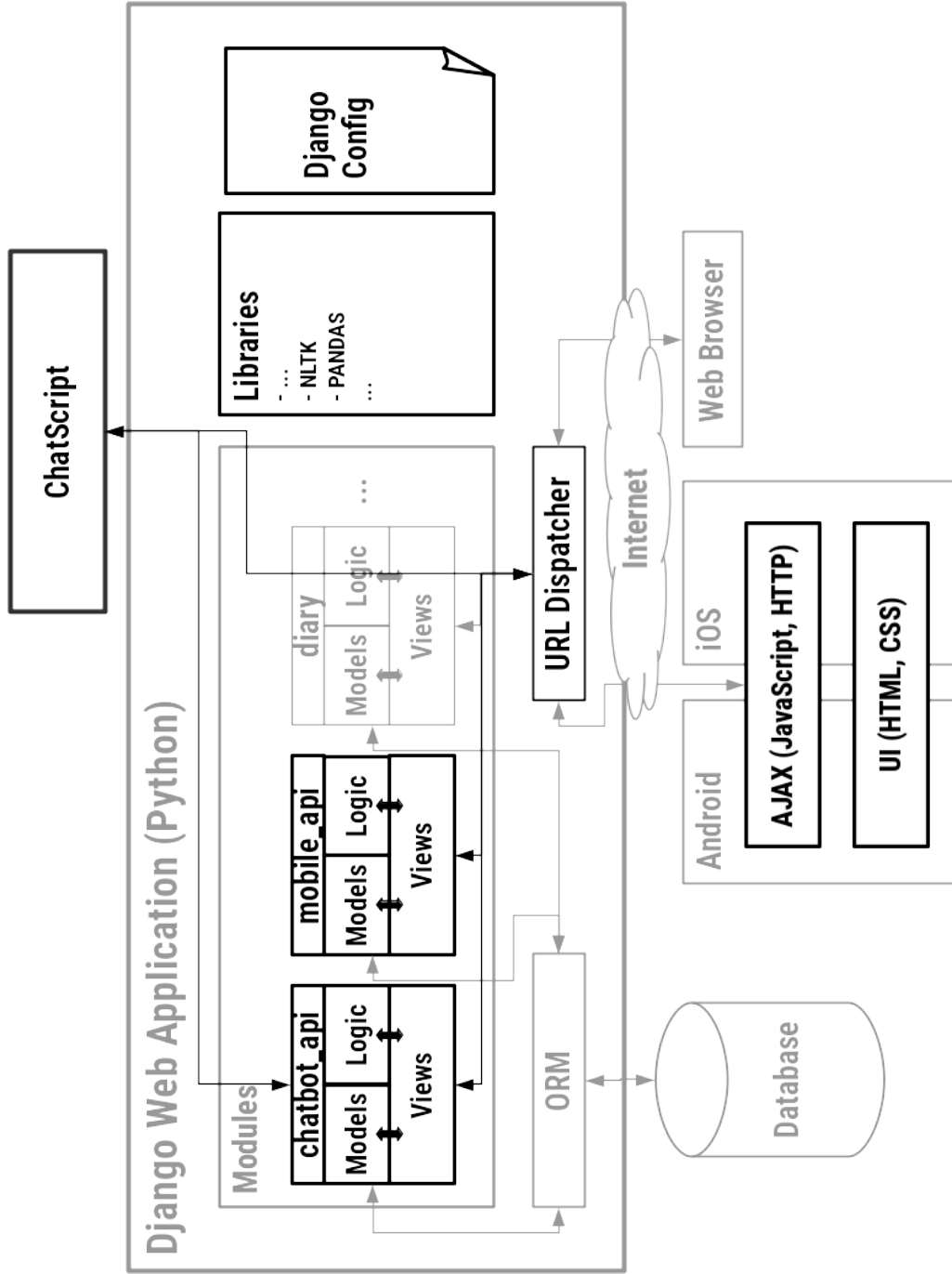


Figure 5.9: The planned architecture of Self eHealth with integrated CA. The components that need to be added, extended or configured are highlighted.



## 6 | Prototype Implementation

*The concept that was designed in the previous chapter was then implemented as a concrete prototype: the Self eHealth Bot (Sebo). This chapter provides implementation details grouped by the different components of the system.*

### 6.1 Technology Introduction

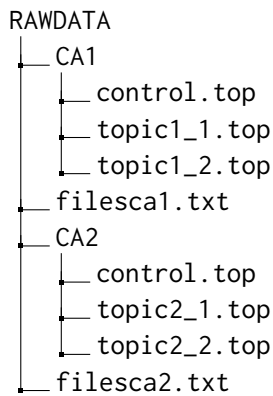
Before having a closer look on the implementation details, this section provides an introduction to the technologies used. The different approaches that were used to work with the selected frameworks and libraries are described here.

#### 6.1.1 ChatScript

The ChatScript framework allows its user to develop multiple CAs at once. Typically all files associated with a specific CA are kept in one folder. In the same directory of this folder, a `.txt` file needs to exist, enlisting the paths to all the files of a CA. This text file is of special importance when compiling the rules of the CA (see also section 6.3.1).

Rules for the CA are kept in `.top` files. It is a good practice to separate different conversations or conversation parts into different files. Every CA needs a control script; this script is also stored as a `.top` file. The control script marks the entry point for the CA: here configuration is done, and the logic of how the different replies are generated can be adjusted to the developers needs. Figure 6.1 shows a sample directory structure, including two CAs. The top directory containing all these configuration files is named RAWDATA in ChatScript.

On a development level, the different parts of a conversation (or different conversations alike) are organized by ChatScript topics. A topic is a collection of multiple expected inputs and their associated replies. For a CA in ChatScript only one topic can be active at a time. Which one is selected depends on the implementation of the control script. In the case of this prototype, the control script contacted the Django web application to retrieve a suitable topic (see also section 6.2).



**Figure 6.1:** A typical directory structure in ChatScript, organizing two bots and its associated files.

```

topic: ~topic

  u: (pattern1)
    output1

  u: (pattern2)
    output2

topic: ~another_topic

  u: (another_pattern)
    another_output

```

**Listing 6.1:** The structure of a typical topic file. It can contain more than one topic.

The topics start with the topic keyword and end either when the next topic begins, or when the file ends. Below the topic declaration, rules are listed that are used when the topic is active. It has to be noted, that this means, all the rules in other topics are getting ignored, although they would match the given input. Listing 6.1 shows the typical content of a topic file. Two topics are available, but only one can be active. Basic rules were already discussed in section 3.5.3; topics and rules for this prototype are described in section 6.3.4.

ChatScript also offers the possibility to create reusable pieces of code. In this framework, that concept is called a *macro*. They were utilized in this prototype to define the communication channel between the ChatScript service and the web application (see also section 6.3.3). Additional to the self-specified macros, ChatScript offers different build-in functions. One that was regularly used in the implementation of the prototype was `^jsonopen`, which allows to contact an external service via HTTP.

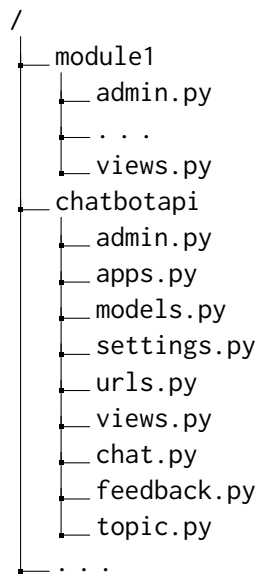


Figure 6.2: The directory structure for the different modules in a Django web application

### 6.1.2 Django

As mentioned in section 5.8, a new module in the Django web application needed to be made. Here the files that were necessary to be included in this module are described.

The `admin.py` file is responsible for configuring the administration pages (see also section 6.10), here the Django models can be selected that should be available in the administration view. An `apps.py` file is required by Django to assign a name to the module, which is necessary to enable to module in the Django configuration file later on. The `models.py` file contains all the models that should associated with this module (see also section 6.6). Objects of these models are mapped to the database using ORM. A `settings.py` can optionally be included in the module to allow configuration of certain parameters of the module to be configured in the global configuration file. The `urls.py` file specifies all the URL that can be used to access the views of the module. The views are specified in a `views.py` file.

In this prototype also the files `chat.py`, `feedback.py`, and `topic.py` were added, to provide the program logic for communicating with the ChatScript process, generating feedback, and selecting the topic, respectively.

Figure 6.2 shows how the different modules are organized within the Django web application.

### 6.1.3 NLTK

NLTK is a library for the Python programming language offering a collection of tools to work with natural language data. It comes with different corpora and lexicons, and offers functionality for "*classification, tokenization, stemming, tagging, parsing, and semantic reasoning*"

(NLTK Project, 2017). Since Django allows to include different Python libraries, also the NLTK can be used within the web application. Especially the sentiment classification functionality is of interest for the implementation of the prototype.

### 6.1.4 pandas

pandas is a Python library that provides a set of data analysis functions that are implemented for high-performance calculations. The functions offered by this library include handling of missing data, pivoting different data sets, aggregating and grouping data, and indexing of data (Pandas Project, 2017). This library was included to the web application to allow fast and simple calculations on the collected and available data in order to generate feedback (see also section 6.5).

### 6.1.5 User Interface

The technologies used for the UI were HTML, CSS, and JavaScript. These languages were interpreted by a web-browser that was included the mobile application of the platform. In other words, the mobile application consisted of a single browser view, that displayed the attached web pages.

HTML was used to define the document structure of the displayed content, where CSS was used to configure the style of the document. Since these two languages do not allow to add functionality to a document, JavaScript was used for this purpose. JavaScript allows to communicate with an external server in an asynchronous way (AJAX), and can be used to alter HTML and CSS while the page is displayed. These are also the two functions of JavaScript that were utilized in this prototype. Since it offers several shortcuts and additional functionality when programming with JavaScript, the jQuery<sup>1</sup> library was used.

A sample AJAX request, using JavaScript is described in listing 6.2. jQuery offers a convenience function, that allows for creating a simple call by specifying the URL of the server, type of the request, data to be sent, and function that should be called when the request was successful and data was received.

Listing 6.3 shows how JavaScript can be utilized to add new elements to the page programmatically and while the page is displayed. An existing element is retrieved using jQuery, a new element is created, and finally appended to the existing container. This changes take effect immediately; the user does not need to refresh the page manually.

---

<sup>1</sup><https://jquery.com/>, Retrieved 20/07/2017

```

1 | $.ajax({
2 |   url: 'some.server/url/to/function', // the URL to connect to
3 |   type: 'POST',                      // type of the HTTP request
4 |   data: someDataVariable,            // a variable containing data
5 |                                       // to send to the server
6 |   success: function(data){          // a function to be called if
7 |     // do something with             // the request was successful
8 |     // the received data
9 |   }
10| });

```

**Listing 6.2: An asynchronous HTTP call to a specific URL using jQuery.**

```

1 | var content = $('.some-class-name'); // retrieve container
2 | var par = document.createElement('p'); // create new paragraph
3 | content.append(par);                // append paragraph to container

```

**Listing 6.3: Altering a displayed HTML page using JavaScript by adding a new paragraph to the existing container.**

## 6.2 Topic Selection

As described in section 5.5, the conversation topics of Sebo should be selected according to the time of the day. For this task, a `getTopic` function was implemented in the system that returns the current topic for a specific user. Table 6.1 enumerates the available topics.

<b>introduction</b>	introductory topic for new users
<b>general</b>	default topic in case no other topic is relevant
<b>breakfast</b>	questions around breakfast
<b>lunch</b>	questions around lunch
<b>dinner</b>	questions around dinner

**Table 6.1: Available topics for selection.**

The topic selection can be described by the pseudo code in listing 6.4. When the user account is newly created, an introduction topic is selected. The topics breakfast, lunch, and dinner are selected if the current time at the location of the user is between a defined time range and this topic has not already been selected on the same day. The time ranges were implemented as settings variables; that means that they can be configured using the standard Django settings file without having to change the source code. In case none of the above topics was selected, a general topic is chosen as a default value.

```
function getTopic(user)
  if user.isNew
    return introduction

  else if BREAKFAST_START_TIME < currentTime < BREAKFAST_END_TIME
    ↪ and not entryExists(breakfast)
    return breakfast

  else if LUNCH_START_TIME < currentTime < LUNCH_END_TIME
    ↪ and not entryExists(lunch)
    return lunch

  else if DINNER_START_TIME < currentTime < DINNER_END_TIME
    ↪ and not entryExists(dinner)
    return dinner

  else
    return general
```

**Listing 6.4: Pseudo code for topic selection**

This function was made available for the ChatScript service in the REST interface via the URL `/chatbotapi/topic/<userid>/`, where `<userid>` is replaced by the unique identifier of a user in the Self eHealth system. More information about the interface can also be found in section 6.4.

## 6.3 Message Generation

For the response generation, ChatScript was used. The ChatScript server process was configured using topic files, containing the rules for creating replies. Short examples of the ChatScript syntax can also be found in section 3.5.3. The following sections describe how ChatScript was utilized in order to realize the planned behavior.

### 6.3.1 Compilation

Before the ChatScript can interpret the topic files (`.top`), they need to be compiled using the commands in listing 6.5). The first command is responsible for compiling the system files that were included in the framework. The `files0.txt` file is a text file containing a list with all files that are relevant for this compilation process. System files include predefined concepts, such as lists of food, common greetings, or activities. The second command is responsible for the compilation of files that are part of a certain CA. In this case, the `filessebo.txt` contained all file names that belonged to Sebo. For this prototype, every topic, as listed in table



```
./LinuxChatScript64 local build0=files0.txt  
./LinuxChatScript64 local build1=filessebo.txt
```

**Listing 6.5: Compilation commands for Linux systems**

6.1, was implemented in a separate file. Additionally, a file responsible for the coordination of the message generation (control script, see section 6.3.2) and a file containing different macros (see section 6.3.3) were present.

### 6.3.2 Control Script

ChatScript allows the developer to configure how the message generation should be done. A topic file (\*.top) can be marked as the control file and is then used for this purpose. For the implementation of the prototype, this file was used to define global variables, perform initialization operations and to configure the sentence processing. It was specified, that at the beginning of each conversation the ChatScript service should contact the web application to retrieve the name of the user and the topic that should be discussed with the user. The rest of the message selection process was left as provided by the framework.

### 6.3.3 Macros

Macros were specified to bundle common procedures and to make them reusable within the configuration scripts. Especially, the calls to the web application was encapsulated using macros. One example for a macro used to retrieve the topic of conversation is shown in listing 6.6.

The first line specifies the name of the macro and the parameter. In the second line, the address under which the requested data could be retrieved is build from the Internet Protocol (IP) address of the server (defined as a global variable in the control script) and the URL of the function. Variable names starting with \$ are global variables, whereas \$\$ is used for local variables. The ^-sign is used for functions and parameters. Line 3 utilizes the build in function ^jsonopen to retrieve a JavaScript Object Notation (JSON) formatted string from a URL. GET specifies the type of request, \$\$url the URL from which the data should be retrieved. The other two parameters are not used in this case and are just included as placeholders. In line 4 the retrieved topic is returned to the caller of the macro.

### 6.3.4 Topic Scripts

The scripts specifying the topics breakfast, lunch, and dinner were structured in a similar way. They were divided into two parts: one that was responsible for defining the questions

```
1 | outputmacro: ^getPreferedTopic(^userid)
2 |   $$url = ^join ($base_url ^'/chatbotapi/topic/^userid/')
3 |   $$json = ^jsonopen(GET $$url $$data $$header)
4 |   $$json.topic
```

**Listing 6.6: ChatScript macro for retrieving the topic from the web application.**

that should be asked, and one that was responsible for handling the expected answers. In the example presented in listing 6.7, the topic responsible for asking questions starts on line 26, and the topic responsible for handling the user input starts at line 2.

When a topic is started, the CA can directly approach the user. The code responsible for this behavior is defined in the blog defined with the `t: keyword` (see line 5 of the example). Normal text (line 6) is replied to the user; the `^respond` function (line 7) is called to append additional text to the response that is produced by the specified topic. In this case, the topic `~breakfast_questions` provides questions that should be asked.

Square brackets are used to define multiple responses from which one is selected randomly. This was used to avoid repeating responses by the CA.

A reply by the user is then again handled by the `~breakfast` topic. There are two blocks that are able to process the input (lines 10 and 20). The files are handled from top to bottom; that means that blocks further up in the file have higher priority. Once the pattern specified in parentheses matches to the sentence of a user, the connected block is executed. This pattern can also include variables, that are checked on whether or not a value is assigned to them. In this case the variable `$expect_hours` is used to limit the number of blocks that are responsible for handling the input. The blocks in lines 10 and 20 only match if the variable is set; a value was assigned to this variable in line 32.

The term `~number` combines two features of ChatScript. The underscore means that the matched word in the input sentence should be stored temporarily to be accessed in the block below. `~number` makes use of one of the concepts that is delivered with ChatScript; it means, that the user input is searched for any number. There are also other concepts available, such as `~insects`, `~locationnames`, or `~food`, each matching with words that are defined to belong to one of these categories.

In this example, that means that there are two possibilities. Either a number is found in the input (then the block at line 10 is used) or no number is included (than the block starting at line 20 defines how to proceed). In the first case, the variable `$sleep_time` is used to store the matched number. The variable `'_0` holds the first word that matched with a pattern starting with an underscore. ChatScript also allows the use of `if` and `else` keywords to add conditional behavior. In this case, it is used to respond differently, depending on the number of hours the user entered. The macro `^addSleep` is used to push the gained information to the web application. Finally, the `~breakfast_questions` topic is queried for more questions. If no more questions can be found, the topic falls back to `~general` where the conversation

```
1 # topic for responding to the user
2 topic: ~breakfast keep repeat []
3
4 # gets executed at topic beinning
5 t: ()
6   good morning, $name!
7   ^respond(~breakfast_questions)
8
9 # handle user input that includes a number if hours are expected
10 u: ($expect_hours _~number)
11   $sleep_time = '_0
12   if ($sleep_time < 6) {Thats not much.}
13   else {$sleep_time hours? Fine!}
14   $expect_hours = null
15   ^addSleep($userid $sleep_time)
16   ^respond(~breakfast_questions)
17
18 # handle user input that does not include a number
19 # when hours are expected
20 u: ($expect_hours)
21   [Hm, I don't understand that. How many hours?]
22   [Can you tell me how many hours?]
23   [Please tell me the number of hours.]
24
25 #topic for asking question
26 topic: ~breakfast_questions keep repeat nostay []
27
28 #if sleep time has not been set yet, ask these questions
29 u: (!$sleep_time)
30   [How long did you sleep last night?]
31   [How many hours did you sleep?]
32   $expect_hours = 1
```

**Listing 6.7: Simplified version of the breakfast topic file.**

is closed. If there was no number entered by the user, the block at line 20 defines a set of replies that ask the user input a correct sentence. This is repeated until a correct answer was entered. The different scripts used for the implementation of the CA were build in a similar style as the given example.

## 6.4 Data Interface

For the communication between the web application and the ChatScript service, a REST interface was developed. It provides different functions to retrieve or add data from/to the Django application. In total, there were four functions that allowed the CA to retrieve information

Name	URL	Description
username	/chatscriptapi/users/<userid>/	retrieve the username of a specific user
topic	/chatscriptapi/topic/<userid>/	retrieve the topic that should be discussed with a specific user
activity	/chatscriptapi/activity/<userid>/	retrieve whether an activity is present for a specific user for the present day.
sentiment	/chatscriptapi/sentiment/?data=<sentence>	retrieve a sentiment score for a certain sentence

**Table 6.2: URLs providing functionality for retrieving information using GET requests**

```

1 | @api_view(['GET'])
2 | def getUsername(request, pk):
3 |     user = get_object_or_404(User, id=pk)
4 |     return Response({'username': user.first_name})

```

**Listing 6.8: Handling a request for a username in Django.**

using HTTP GET and four functions for adding information to the database using HTTP POST requests.

The URLs that allowed the ChatScript service to *retrieve* information are listed in table 6.2. The requests for username, topic and activity all require a unique user ID to be specified, in order to retrieve the data. Since the Django framework by default assigns a unique user id to every account, these ids were also used for this purpose. The URL /chatscriptapi/users/1/ therefore could be used to retrieve the user name for the user with ID 1.

Listing 6.8 shows how a request for retrieving the username for a given user ID is handled in the Django framework. All the database operations are abstracted and handled automatically. In case no user object can be associated with the given id, the HTTP error code 404 is returned. The request of whether or not a activity is present for a given day was done in a similar style. Topic selection was already discussed in section 6.2.

The sentiment extraction of a sentence was done with the help of the NLTK library. More precisely, the VADER sentiment classifier (Hutto & Gilbert, 2014) was used. Depending on the sentiment of the input, the classifier returns values between -2 and 2, where a smaller number stands for negative, and a larger number for positive sentiment. These values were used to react to the response of a user when asked about feelings.

The classifier utilizes a lexicon approach; a list of words is associated with corresponding sentiment values per word. Depending on the occurrence in the sentence, and after taking into account negations, an overall result is calculated. A disadvantage of this approach was

Name	URL	Description
food	/chatscriptapi/food/	add food to the diary
sleep	/chatscriptapi/sleep/	add sleep (time and quality) to the database
activity	/chatscriptapi/activity/	add steps for the present day
assessment	/chatscriptapi/assessment/	add sentiment result for how the user feels

**Table 6.3: URLs providing functionality for inserting information using POST requests**

that adverbs were hardly recognized as instance of their basic form. The performance was improved by a simple trick: adverbs that were not known by the classifier, were transformed to their stem by removing the *-ly* post-fix. For the most important irregular adverbs, a list was created to simply translate them back to their basic form. This naive approach successfully improved the outcome of the detection.

To *insert* the data retrieved from the user to the database, URLs as presented in table 6.3 were provided. The parameters to insert into the database were send as the content of a HTTP POST request. In the web application, this data was extracted and inserted to the database.

## 6.5 Feedback Generation

To provide feedback to the user, it was necessary to analyze the collected data, and retrieve information that could be used as feedback. This process can be described by the six steps enumerated in figure 6.3).

The *collection* of user data is done by the CA, by synchronizing with a fitness device, or by manual data entry by the user. All the data is stored in a database and can be retrieved and altered by the ORM functionality of the Django framework. This means that the stored data is available as objects of the Python programming language. For improved data manipulation functionality, pandas was included as an external library. This library provides functions for querying, aggregating and preprocessing data.

*Preprocessing* was necessary to prepare the data for easier access. This included the limitation to relevant data, and the organization in a simple format for quicker execution of the next steps. In order to make the data comparable, values were *aggregated* to represent certain time ranges. For example, the number of activities were counted for a certain week in order to compare it to previous weeks or to the week of another user. In other words, a pivot table is created, giving the number of activities per user and week. Depending on which kind of feedback should be generated, values that should be compared needed to be *selected*.

After comparing the values, it was *decided*, whether the result is worth for creating a motivational feedback. In general, that meant the comparison had to show a positive progress (e.g.

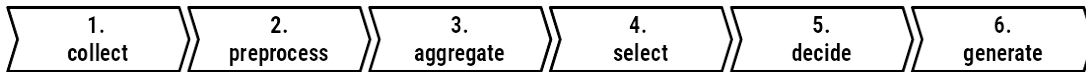


Figure 6.3: Steps for feedback generation.

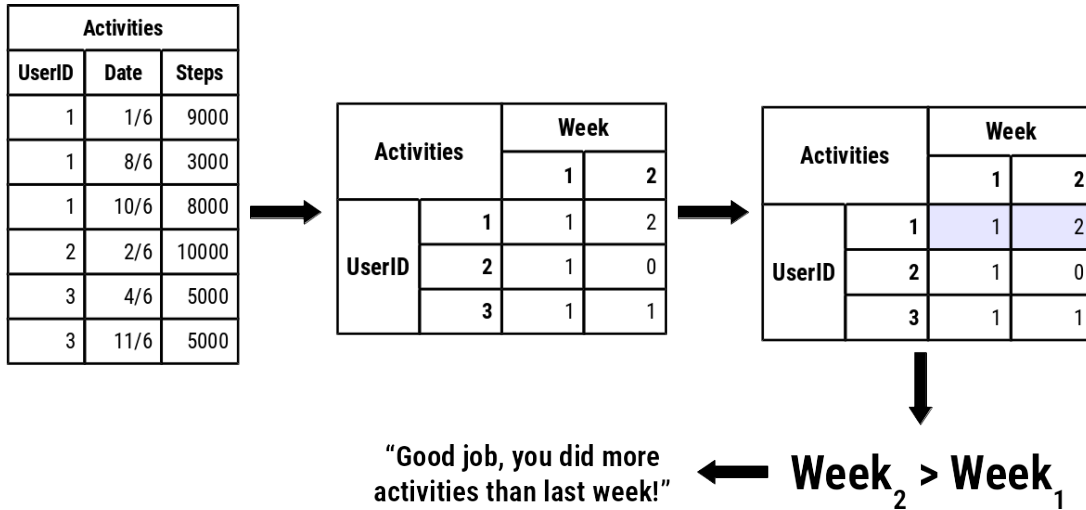


Figure 6.4: Simple transformation example for feedback generation.

the user increased the number of steps in comparison to the previous week). In case positive feedback could be generated, user specific values are inserted into a output template, which was then provided as a feedback sentence for the user.

In figure 6.4, a simple example shows how the transformation of data is done throughout the steps of the generation process.

## 6.6 Data Models

In order to coordinate the user interactions and analyze their behavior with the system, it was necessary to store certain information. For this purpose models were developed, which objects were directly mapped to the database using ORM. First, a status object was kept for every user. This included the current topic, the next message the user is going to receive, information whether the CA is activated for the user, or whether the user should receive notifications.

A second data object was associated to every user that is responsible to store the types of feedback every user received on a particular day. This was necessary to prevent the CA to sent a particular feedback several times in a given period.

To analyze the interaction between the users and the system, a log was kept for every interaction. All conversations could be retrieved from the database to document the communications.

## 6.7 Sebo Heartbeat

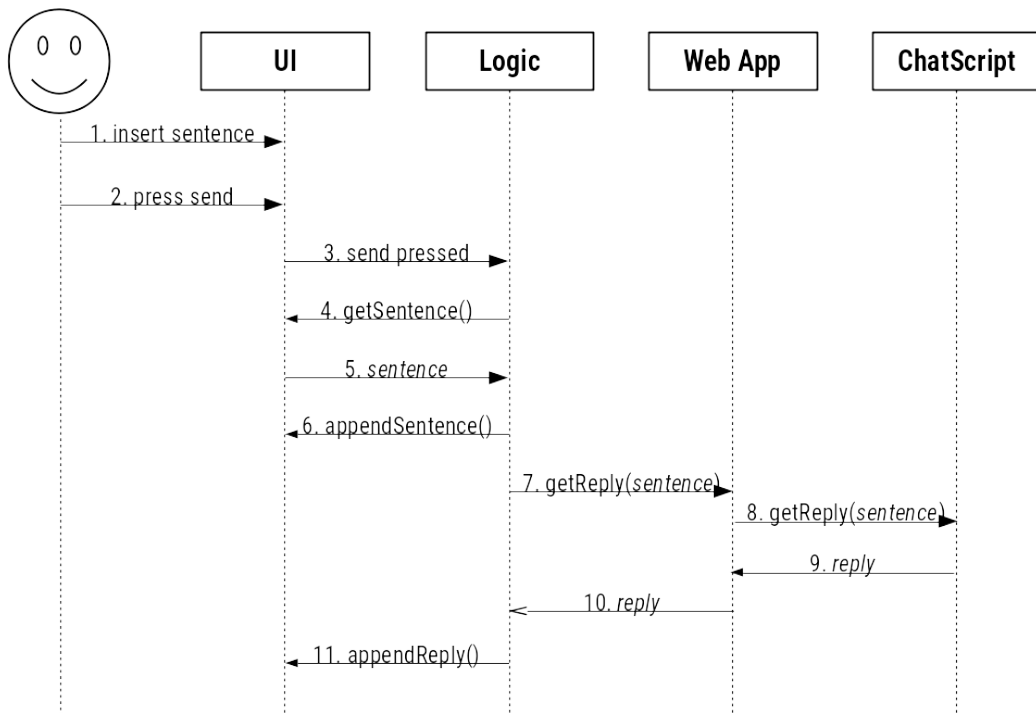
The system was designed to execute a routine in defined time intervals. This routine was responsible for updating the status object for every user. In particular, two important sub-tasks were executed.

The first task was to update the topic for each user. By utilizing the functionality described in section 6.2, a topic was retrieved matching to the user's current time. Given the topic has changed from the last time the routine was called, the ChatScript service was contacted to retrieve the next message for the user. This message was then stored in the status object of the user, where it could be retrieved via the mobile application. Additionally, the routine was started to generate feedback (see section 6.5). In case feedback could be retrieved, it is also stored in status object. When the topic changed to either of the breakfast, lunch, or dinner topic, or a feedback was generated, a notification was sent to the user.

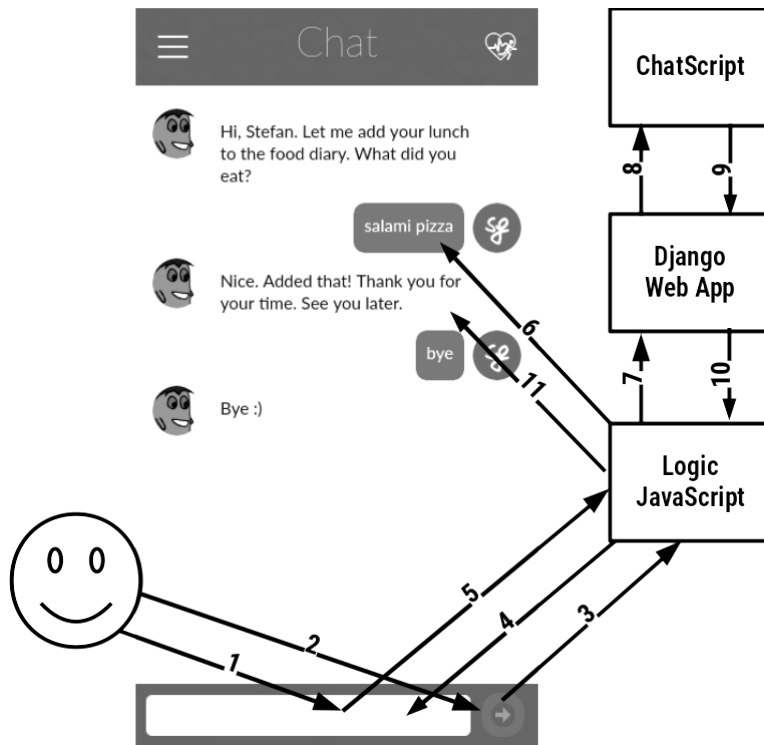
## 6.8 User Interface

One file of each type – .html, .css, and .js (JavaScript) – were sufficient for implementing the UI and its functionality. The HTML file inherited the structure from the preexisting HTML files of the project. Thus, only the bottom bar, including a text input field and a button, had to be added (see figure 6.6c). The responsibility of the JavaScript logic was to retrieve data and add it to the UI as HTML elements. Once the user entered text to the text field and pressed the send button, the JavaScript program retrieved this text, and appended it as HTML code to the HTML page (similar to as described in listing 6.3). Additionally, this text was also sent to the Django web application as a parameter of an AJAX call. As a result of this call, the JavaScript logic received the response of the CA. Once the data has been received, it was also formatted as HTML code and appended to the HTML page.

A detailed description of the interactions between the different layers of the system in case of an entered sentence is displayed in figure 6.5: as a sequence diagram (figure 6.5a), and as a graphical representation of the UI (figure 6.5b). When the user enters a sentence into the text field (1) and presses send (2), the logic layer (JavaScript) is notified (3). This layer then retrieves the sentence from the HTML page (4 & 5). The sentence is used to append HTML code including the sentence to the UI (6). Additionally, the sentence is sent to the web app as a request for a reply (7). This request is asynchronous; that means that the JavaScript code continues executing while waiting for the reply. The sentence is forwarded to the ChatScript service (8), where a reply is generated. This reply is then returned to web application (9) where it is sent back to the mobile application (10). Once the reply is received, it is appended to the HTML page (11), similar to the sentence previously.



(a) Sequence Diagram showing the data during a typical interaction between user and system.



(b) Graphical representation of a standard interaction between user and system.

Figure 6.5: Two representations of the standard use case. The numbers show the equivalent steps in the process.



For styling the HTML document, the CSS file specified the style attributes. To display an image next to the messages of the CA and provide the user with a "physical" counterpart, an avatar was created (see figure 6.6b). The avatar was named *Sebo*. An image of the user could be retrieved from the web application, where images already have been stored.

There were two ways created to launch the CA interface: (1) by clicking on the notification displayed when receiving a new message (see figure 6.6a, or (2) by pressing on the "Talk to Sebo" button that was integrated into the "My Day" screen of the mobile app (see figure 6.6d).

## 6.9 Conversation Design

The CA was designed to contact the user at least three times a day, every time one of topics breakfast, lunch, or dinner was selected, or when feedback was generated. For this prototype the breakfast topic was active from 8 a.m. to 1 p.m., the lunch topic was active from 1 p.m. to 6 p.m. and the dinner topic from 6 p.m. to 10 p.m. This means that the users were notified about topic changes at 8 a.m., 1 p.m. and 6 p.m. If feedback was present at the same time, it was appended to the messages sent as part of the topics. Otherwise, a separate message with the feedback message was sent.

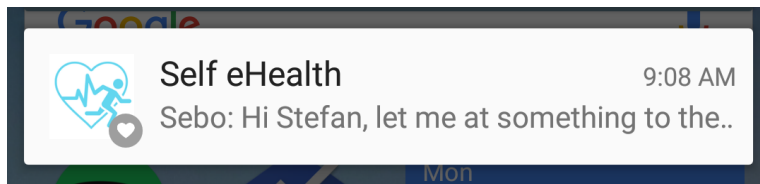
An example of a conversation between a user and the CA throughout a day is presented in listing 6.9. First, when the user never talked to the CA before, the CA gives a short and friendly introduction. In the breakfast topic, Sebo starts with giving feedback by endorsing the user for being on an 18 day streak of achieving 10.000 steps per day. After that, Sebo asks the user about the breakfast, sleep duration and sleep quality. After all the information has been collected, the CA closes the conversation.

The lunch topic is designed shorter; only the food consumed as lunch is added to the system. At the dinner topic, Sebo starts with asking the user about how the day was. As in all other topics, the user is questioned about food. The CA also accepts the case that the user did not eat anything. In the case of this sample conversation, the user already added steps to the Self eHealth platform. Therefore, Sebo compliments the user on this action, and asked about how the activity was experienced. After having completed all the questions, the CA says goodbye.

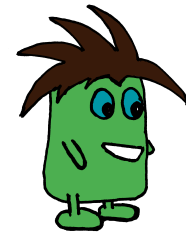
## 6.10 Maintainability and Administration

For administrating the data managed by the web application, the Django framework allows to create web pages for displaying and editing of the stored data. Figure 6.7 shows an example, displaying the admin page of the status data objects.

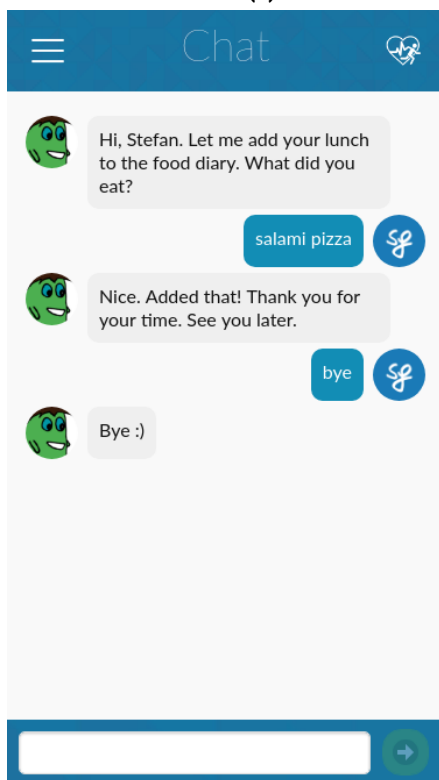
Furthermore, the Django framework allows to create commands that enables certain functionality of the web application to being executed as shell commands on the operating system.



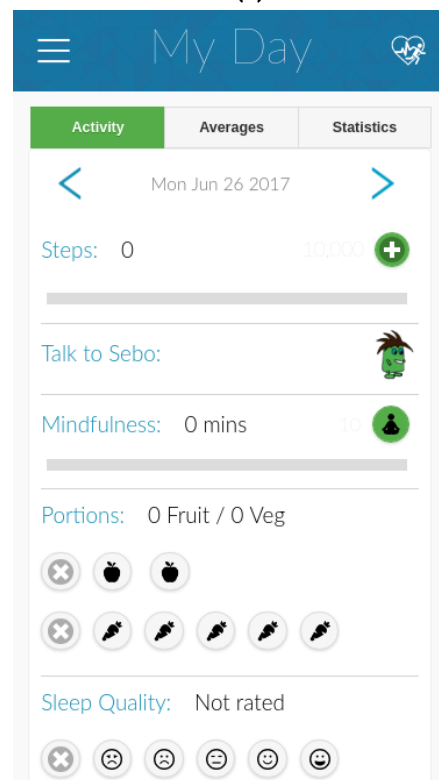
(a) Notification on Android



(b) The Sebo avatar



(c) User Interface



(d) "Talk to Sebo" on the MyDay screen.

Figure 6.6: Different components of the user interface.

Two of these commands were created for this prototype. The first command allowed easy compilation of changed ChatScript (see also 6.3.1) files via a Secure Shell (SSH) connection, without having to execute several commands manually. Another command was designed to execute the heartbeat tasks described in section 6.7 timed by the Linux *cron* program. This program can be configured to execute specified tasks in regular time intervals. Since the topics of the CA only change every few hours, one hour was specified as the interval for this implementation.

## 6.11 Summary

This chapter described the components of the prototype. For selecting the right topic for a user, a routine was described that handles this task with respect on time and user data. Depending on the selected topic, ChatScript is utilized to generate messages which are sent to the mobile application. ChatScript was configured by several files: one control script for the overall setup of a conversation, five topic scripts (one for each topic), and a utility file containing macros for communicating with the web application. Furthermore, feedback is generated and delivered to the users; this generation is done in regular time intervals. For displaying the messages of the CA a UI was implemented, including the development of an avatar that serves as a counterpart to the user. For administrative purposes, web pages for reading and configuring the produced meta data (chat status of the user, log of used feedback messages, and a copy of the conversations), were added. Commands were implemented for compiling the ChatScript files and for triggering the heartbeat of the system. The next chapter describes the study that was planned for testing the acceptance of the implemented prototype.

```
# introduction when the user first gets contacted by Sebo
Sebo: "Hi, User. My name is Sebo, I'll be your buddy here.
      Sometimes, I'll ask you for some information about yourself.
      This helps me to make your experience as awesome as possible :)

# breakfast topic, notification sent at 8 a.m.
Sebo: Good morning, User!
      You are on an 18 day streak. Keep up the good work!
      Let me add something to the food diary for you.
      What did you eat?
User: blueberries and yogurt and coffee
S: Yummy. I like that you eat fruit. It's in your diary now.
   How many hours did you sleep?
U: 7
S: 7 hours? Fine! How well did you sleep?
U: Very well
S: Nice. Thank you, I'll come back later.

# lunch topic, notification sent at 1 p.m.
Sebo: Hi, User. Let me add your lunch to the food diary.
      What did you eat?
User: bread with peanut butter
S: Nice. Added that! Thank you for your time. See you later.
U: bye
S: Bye

# dinner topic, notification sent at 6 p.m.
Sebo: Hi, User! How did you do today?
User: Good
S: That sounds nice. What did you eat for dinner?
U: Nothing
S: Well, maybe next time. Oh, you've already added steps.
   Good Job! How did you like it?
U: Good
S: Sounds good. Thank you, I'll come back later.
```

**Listing 6.9: Sample conversation between the CA and a user during one day.**

WELCOME STEFAN | [VIEW SITE](#) / [CHANGE PASSWORD](#) / [LOG OUT](#)

## Django administration

Home > [Chatbotapi](#) > [Chat status](#)

ADD CHAT STATUS +

Select chat status to change

Action: -----  0 of 14 selected

<input type="checkbox"/>	ID	TOPIC	MESSAGE	FEEDBACK	ISACTIVE	ISNEW	SENDNOTIFICATIONS
<input type="checkbox"/>	2418	~lunch	Hi, Manon. <feedback> Let me add your lunch to the food diary. What did you eat?	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2417	~breakfast	-	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2416	~breakfast	Good morning, Andreasi <feedback> Let me add something to the food diary for you. What did you eat?	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2415	~breakfast	Good morning, Davidl <feedback> What did you have for breakfast?	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2414	~breakfast	Good morning, Sandra! <feedback> What did you have for breakfast?	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2413	~breakfast	Good morning, Janine ! <feedback> What did you have for breakfast?	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2412	~general	-	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2411	~lunch	Hi, Reece. <feedback> Let me add your lunch to the food diary. What did you eat?	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2410	~introduction	-	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2409	~introduction	Good morning, Lorraine! Can I add some breakfast to the food diary for you?	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2408	~introduction	Do you have any questions?	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2407	~lunch	Hi, David. <feedback> Let me add your lunch to the food diary. What did you eat?	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2406	~general	Thank you, I'll come back later.	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<input type="checkbox"/>	2405	~lunch	Hi, John. <feedback> Let me add your lunch to the food diary. What did you eat?	-	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

14 chat status

Figure 6.7: The admin interface provided by the Django framework.



## 7 | User Study

*The implemented prototype should be tested regarding its influence on the users. This chapter describes the original idea for the study and the adapted version that was used in this thesis.*

### 7.1 Originally Planned Study

Originally, it was planned to conduct a study comparing three implementations of the Self eHealth system to measure the influence of a CA. The study application was submitted for review by the Human Research Ethics Committee at the University of Western Australia (UWA), but it has not been approved before the submission of this thesis.

The plan for this study was to find out whether an included CA would increase (1) the number of entries in the food diary, and (2) the number of activities added to the fitness tracker. For this reason, 60 people should be recruited that agree to use the mobile application for two weeks. To compare the impact of a CA the participants should be randomly assigned to one of three groups. The first group would receive the default Self eHealth mobile application and would serve as a control group for this trial. Group two and three would use a version where Sebo was included, but in group two, the CA would only behave passively, and no notifications would be sent. This separation was planned to measure the effect of the proactive approach.

### 7.2 Study Setup

Since it was not possible to implement the study within the time boundaries of the project, an alternative study was planned. This study was not organized as a research project at the UWA, but at Graz University of Technology. Therefore, the ethics standards of Graz University of Technology applied, and approval of the UWA Human Research Ethics Committee was not needed. As a consequence, the recruitment of study participants could not be done over the communication channels of the UWA, and the originally planned number of participants was not able to be reached.

Opposing to the originally planned study, it was decided to implement an exploratory study on the acceptance of the prototype with six to eight participants. These participants were asked to use the mobile version of the Self eHealth platform with the included CA for two weeks. After that they filled out a questionnaire that was designed to measure the acceptance of the CA by the users. As demographic information, the age and gender of the participants were collected. Logs for the conversations were kept to derive information about problems and usage behavior.

## 7.3 Questionnaire

For measuring how well the prototype is accepted by the participants, two standardized scales – the Computer Emotion Scale (CES) (Kay & Loverock, 2008) and the System Usability Scale (SUS) (Brooke, 1996) – were used in combination with additional system specific questions. The complete questionnaire can be found in appendix B; part 1 implements the CES, part 2 the SUS, and part 3 includes the system specific questions.

### 7.3.1 Computer Emotion Scale

The CES is intended to measure the emotions that occur when people learn to use new software. In particular, four emotions were selected for this purpose: anger, anxiety, happiness, and sadness. The study participant is asked to rate the degree of 12 specific feelings on a Likert scale from 0 to 3. Each of these feelings is part of one of the emotion constructs listed above. Table 7.1 shows their assignments. To interpret the result, the ratings are summed up for each construct and normalized by dividing with the number of feelings that are associated to the construct (Kay & Loverock, 2008).

This scale was selected to understand the emotions involved when working with the prototype. Opposing to open ended questions regarding the feelings of the participants, the CES produces comparable results. This allows to find out which emotions prevailed during the study, and enables to compare these results to with possible future studies.

Emotion Construct	Feelings
<b>Happiness</b>	Satisfied, Excited, Curious
<b>Sadness</b>	Disheartened, Dispirited
<b>Anxiety</b>	Anxious, Insecure, Helpless, Nervous
<b>Anger</b>	Irritable, Frustrated, Angry

**Table 7.1: Assignment of feelings to the emotion constructs.**



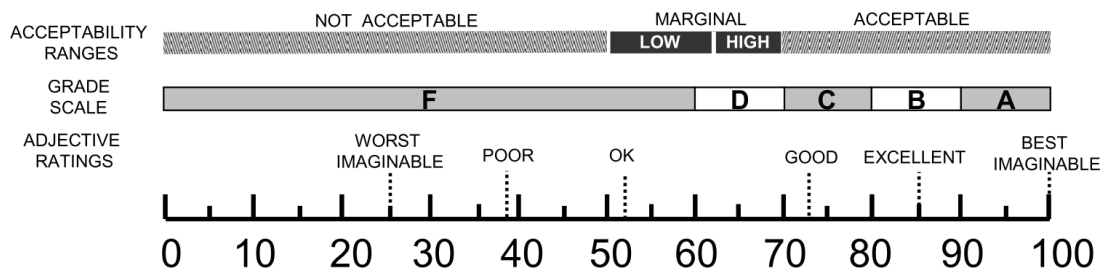


Figure 7.1: The interpretation of the SUS according to Bangor et al. (2009)

### 7.3.2 System Usability Scale

The SUS is a simple scale formed around ten items on a Likert scale. Every element includes a statement about the system, which can be rated between 0 (strong disagreement) to 4 (strong agreement). The scale is designed to be used after a study participant has used the system under evaluation. For calculating the score, the ratings for questions 1,3,5,7, and 9 are subtracted by one and the result is summed up. For the questions 2,4,6,8, and 10 the element rating is subtracted from 5 and is also added to the sum. This sum is then multiplied by 2.5, which leads to a result between 0 and 100 (Brooke, 1996). Bangor et al. (2009) propose the scoring system in figure 7.1 to interpret the outcome of the SUS. They see a score of 70 as the lower limit for a system to pass a usability test.

The SUS was included into the questionnaire to compare how the subjective usability of the prototype was experienced by each participant. Beside the score, that allows to compare potential future enhancements, also concrete answers can give information about possible weaknesses of the implementation.

### 7.3.3 System Specific Questions

Additional to the general scales, specific questions were used in the questionnaire. Four statements about the CA were listed, allowing the user to rate the degree of agreement between 0 (strongly disagree) and 4 (strongly agree). These questions were selected to measure the subjective influence of Sebo on the whole Self eHealth platform.

Furthermore, two open questions asked the study participant to specify reasons she did or did not like the CA. The third open questions was intended to evaluate the motive for continuing or discontinuing to use the platform after finishing the study. These questions reach out to retrieve information on how the system could be improved in the future.

## 7.4 Implementation

For the study, ten people were contacted within a period of two months. Since the participants did not interact with each other, it was decided that it was possible to run the study period of two weeks independently for each participant; that means that each participant could start at an individual point in time.

Once a person agreed to participate in the study, further instructions were given on how to install the app and register on the platform. The participants were instructed to use the mobile application to their convenience. The study period was started as soon as the participant successfully received a message from the CA. Once the study period was over, the participant was notified. The questionnaire, as described above, was made digitally available using Google Forms<sup>1</sup>, and the link was sent to participants that completed the study.

## 7.5 Summary

To evaluate the acceptance of the implemented prototype, a study was designed. The SUS and the CES were used as standardized scales to measure the usability of the system and the emotions of the user when interacting with it. Together with system specific questions, these scales were combined to a questionnaire. Ten users were asked to use the mobile application for a period of two weeks and afterwards fill out the questionnaire. The results are discussed in the next chapter.

---

<sup>1</sup><https://docs.google.com/forms/u/0/>, Retrieved 13/07/2017

## 8 | Results and Discussion

*The results of the study described in the previous chapter are discussed here. This includes the different scales, open questions, and additional information that was provided by the participants. The discussion includes suggested improvements for the system that were derived from the user feedback.*

### 8.1 Results

After using the mobile application with the included CA for two weeks, each participant filled out the questionnaire described in the previous chapter. The results are discussed below and are split according to the parts in the questionnaire.

#### 8.1.1 Participants

Originally, ten people agreed to participate in the study. For three of the ten people it was not possible to receive messages from the CA. The reason for this problem was investigated, but the cause could not be located<sup>1</sup>. After excluding these three people, a total of seven people (three male and four female) participated in the study. The average age of the participants was 32 with a standard deviation (SD) of 12.08. The native language of two participants was English; for the other five people, German was the native language.

#### 8.1.2 Computer Emotion Scale

Table 8.1 contains the collected results of the CES. It displays the number and per cent of users that selected a certain answer, grouped by the four basic emotions. None of the users selected the answer *all of the time* for any of the emotions.

According to figure 8.1, *anxiety* got the lowest score; an average rating of 0.04 (SD 0.09) on a scale between 0 (none of the time) and 3 (all of the time) was measured. Only one study

---

<sup>1</sup>It was assumed that some versions of the Android operating system were not compatible with the application.

	none of the time	some of the time	most of the time	all of the time
<i>satisfied</i>	0 (00.0%)	5 (71.4%)	2 (28.6%)	0 (00.0%)
<i>excited</i>	2 (28.6%)	2 (28.6%)	3 (42.9%)	0 (00.0%)
<i>curious</i>	1 (14.3%)	4 (57.1%)	2 (28.6%)	0 (00.0%)
<b>Happiness</b>	<b>3</b> (14.3%)	<b>11</b> (52.4%)	<b>7</b> (33.3%)	<b>0</b> (00.0%)
<i>disheartened</i>	5 (71.4%)	2 (28.6%)	0 (00.0%)	0 (00.0%)
<i>dispirited</i>	7 (100.0%)	0 (00.0%)	0 (00.0%)	0 (00.0%)
<b>Sadness</b>	<b>12</b> (85.7%)	<b>2</b> (14.3%)	<b>0</b> (00.0%)	<b>0</b> (00.0%)
<i>anxious</i>	7 (100.0%)	0 (00.0%)	0 (00.0%)	0 (00.0%)
<i>insecure</i>	7 (100.0%)	0 (00.0%)	0 (00.0%)	0 (00.0%)
<i>helpless</i>	6 (85.7%)	1 (14.3%)	0 (00.0%)	0 (00.0%)
<i>nervous</i>	7 (100.0%)	0 (00.0%)	0 (00.0%)	0 (00.0%)
<b>Anxiety</b>	<b>27</b> (96.4%)	<b>1</b> (03.6%)	<b>0</b> (00.0%)	<b>0</b> (00.0%)
<i>irritable</i>	3 (42.9%)	3 (42.9%)	1 (14.3%)	0 (00.0%)
<i>frustrated</i>	2 (28.6%)	4 (57.1%)	1 (14.3%)	0 (00.0%)
<i>angry</i>	3 (42.9%)	4 (57.1%)	0 (00.0%)	0 (00.0%)
<b>Anger</b>	<b>8</b> (38.1%)	<b>11</b> (42.4%)	<b>2</b> (09.5%)	<b>0</b> (00.0%)

**Table 8.1: Number of times an answer on the Computer Emotion Scale has been selected.**

participant selected an answer associated with this emotion. Also *sadness* reached a low score:  $0.14$  ( $SD\ 0.24$ ); here, two participants selected feelings that are connected with sadness. *Anger* and associated feelings were felt by six out of seven participants. This led to an average rating of  $0.71$  ( $SD\ 0.40$ ). That means that most of the users experienced anger at least some of the time. The strongest emotion was *happiness*: an average of  $1.33$  ( $SD\ 0.47$ ) points were put in answers associated to this emotion. This is also the only emotion all participants experienced at least to some amount. Table 8.2 summarizes the average ratings and SDs.

Figure 8.2 gives an additional view on the data. It shows the distribution of the possible answers grouped by the four emotions. For *anxiety* (96.4%) and *sadness* (85.7%) a great majority of the answers, *none of the time* was selected. The feelings associated to *anger* were felt *some of the time* in the majority of answers. This can be compared to the *happiness* emotion, where the same number of answers were selected for this choice. Nevertheless, for this emotion the *most of the time* option was selected a third of the time (compared to under 10% at the *anger* emotion).

### 8.1.3 System Usability Scale

In figure 8.3 the answers on the SUS are collected. Five out of seven people stated that they would not continue using the system after the study. All participants strongly disagreed or

Emotion	Average Rating	SD
happiness	1.33	0.47
sadness	0.14	0.24
anxiety	0.04	0.09
anger	0.71	0.40

Table 8.2: The average rating of all users per emotion and its standard deviation.

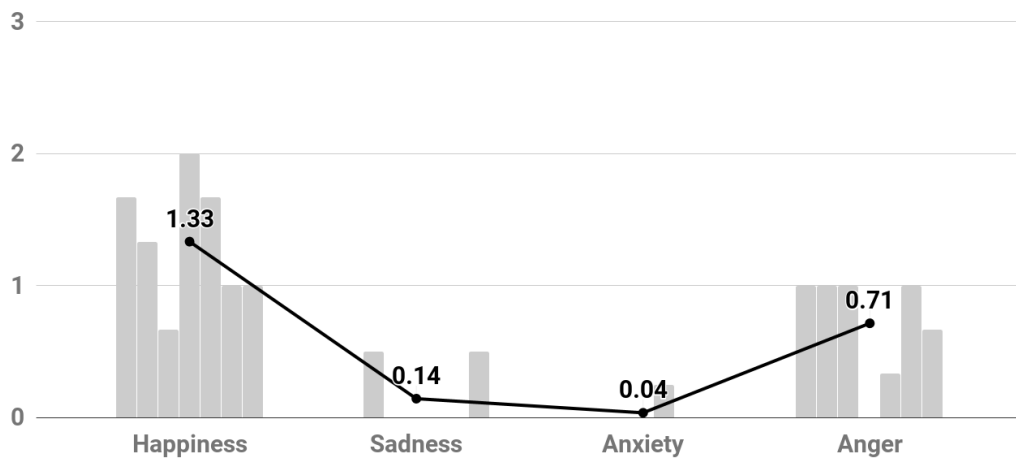


Figure 8.1: Average points per user and emotion on a scale between 0 (none of the time) and 3 (all of the time). The black lines shows the average rating including all participants.

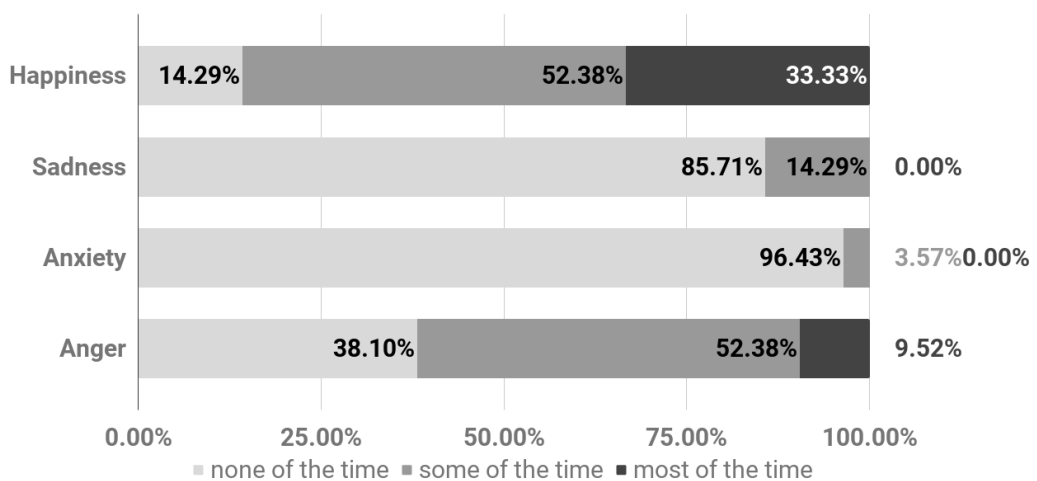
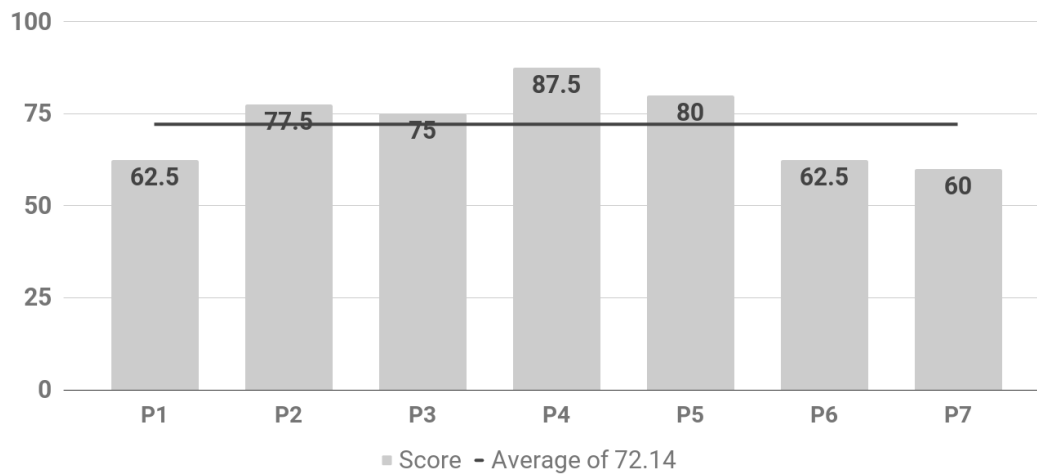


Figure 8.2: Distribution of answers grouped by the basic emotions on the Computer Emotion Scale. Choice "all of the time" was never selected.



**Figure 8.3: Results of the System Usability Score per participant P1 – P7.**

disagreed to the statement that the prototype was too complex and that they would need help from a technical person while using this system. Most participants found the prototype easy to use, although one user strongly disagreed on this statement. Two users felt that functions of the system were not well integrated, while five agreed or strongly agreed that integration was done well. Inconsistency in the functionality was criticized by one person, three agreed that the system was cumbersome to use. All participants agreed that people would not need a lot of time to learn to use the system, and most felt confident while using the prototype.

From these answers scores were calculated as explained in section 7.3.2. The results are shown in figure 8.3. The average rating was  $72.14$  ( $SD\ 10.55$ ). This is slightly over the limit of 70 that was set by Bangor et al. (2009) as the minimum result for a usable system. It should be noted that the results vary strongly between the different participants. Between the lowest (60) and the highest score (87.5) lie 27.5 points. From another perspective, four users rated the prototype over the 70-point threshold, three saw it below this number.

### 8.1.4 System Specific questions

Five out of seven participants stated that the CA motivated them to add more entries to the food diary; three users felt motivated to add more steps that they would have otherwise. The participants generally saw the CA as an improvement to the overall experience: one person strongly agreed to this statement, five agreed, and one person was undecided. The general tendency was that there were too many notifications sent; 4 participants agreed or strongly agreed to this statement, where two were undecided. One person did not agree that too many messages were sent, in this case a technical error prevented this participant to receive any notifications at all.

	strongly disagree	disagree	undecided	agree	strongly agree
I think that I would like to use this system frequently.	1 (14.3%)	4 (57.1%)	0 (00.0%)	2 (28.6%)	0 (00.0%)
I found the system unnecessarily complex.	4 (57.1%)	3 (42.9%)	0 (00.0%)	0 (00.0%)	0 (00.0%)
I thought the system was easy to use.	1 (14.3%)	0 (00.0%)	1 (14.3%)	4 (57.1%)	1 (14.3%)
I think that I would need the support of a technical person to be able to use this system.	6 (85.7%)	1 (14.3%)	0 (00.0%)	0 (00.0%)	0 (00.0%)
I found the various functions in this system were well integrated.	0 (00.0%)	2 (28.6%)	0 (00.0%)	4 (57.1%)	1 (14.3%)
I thought there was too much inconsistency in this system.	2 (28.6%)	4 (57.1%)	0 (00.0%)	1 (14.3%)	0 (00.0%)
I would imagine that most people would learn to use this system very quickly.	0 (00.0%)	0 (00.0%)	0 (00.0%)	6 (85.7%)	1 (14.3%)
I found the system very cumbersome to use.	2 (28.6%)	1 (14.3%)	1 (14.3%)	3 (42.0%)	0 (00.0%)
I felt very confident using the system.	0 (00.0%)	1 (14.3%)	0 (00.0%)	5 (71.4%)	1 (14.3%)
I needed to learn a lot of things before I could get going with this system.	4 (57.1%)	3 (42.9%)	0 (00.0%)	0 (00.0%)	0 (00.0%)

**Table 8.3: Number of times an answer on the System Usability Scale has been selected.**

	strongly disagree	disagree	undecided	agree	strongly agree
"Sebo" motivated me to add more food to the diary.	0 (00.0%)	2 (28.6%)	0 (00.0%)	4 (57.1%)	1 (14.3%)
"Sebo" motivated me to add more activities/steps.	2 (28.6%)	2 (28.6%)	0 (00.0%)	3 (42.9%)	0 (00.0%)
"Sebo" improved the experience of the system.	0 (00.0%)	0 (00.0%)	1 (14.3%)	5 (71.4%)	1 (14.3%)
"Sebo" contacted me too often.	1 (14.3%)	0 (00.0%)	2 (28.6%)	2 (28.6%)	2 (28.6%)

**Table 8.4: Number of times an answer was selected at the system specific questions.**

### 8.1.5 Other Feedback By Participants

This section combines the feedback from the questions "What did you like when using 'Sebo'?", "What did you dislike when using 'Sebo'?", and additional comments received in conversations.

Many positive comments were given in regard to the appearance of the CA. Participants commented positively on the polite way it asked for data, the use of emoticons, and its appearance overall. The motivational feedback was also endorsed, as well as the "personalized experience".

Several points were criticized. Multiple participants stated that the notifications were either too many, or came at inconvenient times. Especially the fact that the CA was configured to send messages at a fixed time every day led to criticism: often people were asked about what they ate before they had the corresponding meal. Adding the meals at a later time was not supported, which led to confusion ("Why can't I add my breakfast later on?"). Listing 8.1 shows how one of these conversations looked like.

Sometimes, incorrect or inaccurate input understanding by the CA led to inappropriate replies or actions. When food was added, the whole input was stored, not only the food names: "I had an apple" would be stored as it is, instead of just storing "1 apple". Similarly, the input "more than 100 steps" was added as exactly 100 steps. Another example of inaccurate understanding was the detection of concepts like "tomato soup" as two separate words. The answer "I like that you eat vegetables" that was triggered by the word "tomato" was seen as inappropriate for this input. When detecting how a user felt, the CA sometimes did not correctly understand the user. Listing 8.2 shows such a conversation, leading to an unsuitable answer.

Whether or not people like tracking their lifestyle played a big role for further using the app after completing the study. Not liking to track their activities was the main argument for stopping, where having a positive attitude towards fitness tracking convinced the participants to continue to use the mobile application. In conclusion, 5 participants decided to stop and 2 wanted to keep using the platform.

### 8.1.6 Behavior Analysis

Additionally to the study questionnaire, also the behavior of the participants was analyzed. Table 8.5 shows how often a participant interacted with the CA grouped by week and topic. In week 1 the breakfast topic was interacted with 4.9 times (SD 1.8) on average. This was more often as with the lunch topic (3.6 times, SD 2.6) and the dinner topic (3.3 times, SD 2.6). During the second week of the study, these number dropped to 2.4 interactions (SD 2.4) for the breakfast topic, 2.3 interactions (SD 2.2) on average for the lunch topic and 2.1 interactions (SD 2.5) for the dinner topic.



```
Participant: Hi sebo
Sebo: I don't have anything new for you right now. Come back later.
P: Can I tell you about my lunch?
S: Can't answer that, sorry.
P: I had lunch
S: I'll contact you later.
P: But... i am having it now
S: I will let you know when I have more to say.
P: Hum ok
S: Come back a little later.
P: Ok
S: I don't have anything new for you right now. Come back later.
P: Ok
S: I will let you know when I have more to say.
...
```

**Listing 8.1: Sample conversation showing the problem when a user tried to add food at a later point in time.**

```
Sebo: ... How long did you sleep last night?
Participant: Less than 6 hours
S: 6 hours? Fine! How well did you sleep?
P: Not so good
S: That sounds awesome! Thank you, I'll come back later.
...
```

**Listing 8.2: Sample conversation showing a miss-classification of a feeling.**

This decrease in interactions is also displayed in figure 8.4. The average number of interactions in week 1 (11.7, SD 6.1) decreased to 6.9 (SD 6.9) in week 2. During the study, a participant communicated with the CA 18.6 times (SD 12.5) on average. This is more than once a day, during the 14-day period. The number of interaction varied strongly between the participants; one achieved a total of 35 interactions (around 2.5 per day), where two participants only conversed with the CA 7 times (average of 0.5 interactions per day). Figure 8.5 show the number of interactions grouped by users and the different topics.

## 8.2 Discussion and Suggested Improvements

The first improvement that could be derived from the user feedback, is to implement a more user oriented timing of messages. This means to either let the user decide on when they want to receive messages, and how many they want to receive in a certain time period, or automatically learn the preferred settings from the user behavior. People often have a different daily time schedule, thus a general setting for all users does not allow to satisfy every lifestyle. The easiest way to implement this improvement would be to add a preference setting for

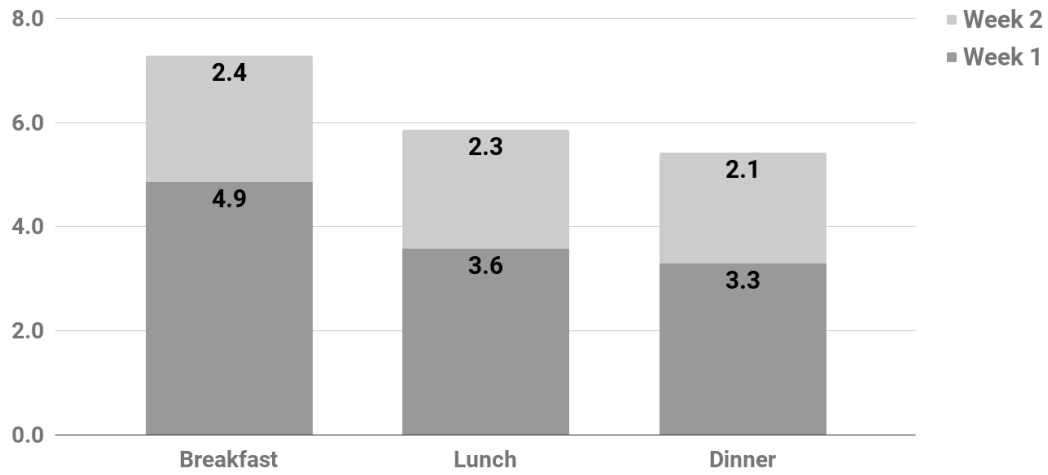


Figure 8.4: Average number of interactions with the CA by week and topic.

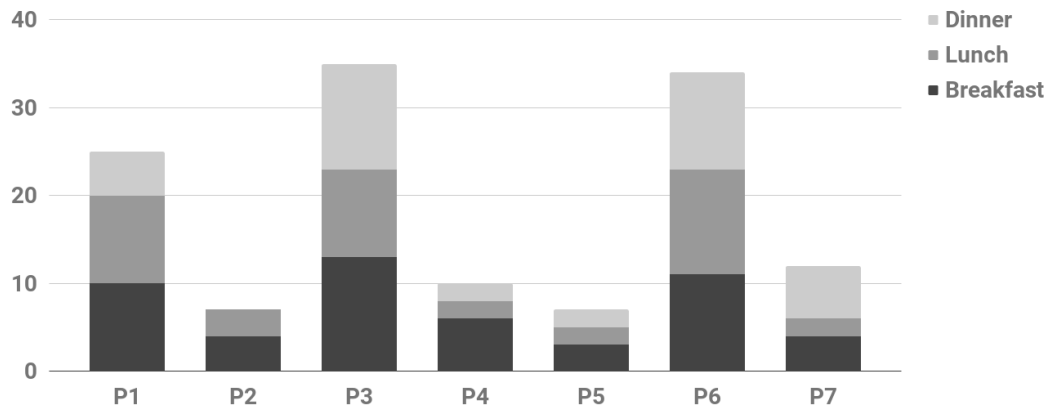


Figure 8.5: Number of interactions by user and topic during the two weeks of the study. The labels of this table do not correlate with those in figure 8.3.

	Week 1			Week 2			Total			
	Breakfast	Lunch	Dinner	Breakfast	Lunch	Dinner	Breakfast	Lunch	Dinner	All
	P1	7	7	2	3	3	3	10	10	5
P2	3	1	0	1	2	0	4	3	0	7
P3	7	6	6	6	4	6	13	10	12	35
P4	4	2	1	2	0	1	6	2	2	10
P5	3	1	2	0	1	0	3	2	2	7
P6	6	6	6	5	6	5	11	12	11	34
P7	4	2	6	0	0	0	4	2	6	12
Average	4.9	3.6	3.3	2.4	2.3	2.1	7.3	5.9	5.4	18.6
SD	1.8	2.6	2.6	2.4	2.2	2.5	4.0	4.6	4.6	12.5
Median	4	2	2	2	2	1	6	3	5	12

**Table 8.5: Number of times the study participants (P1-P7) interacted with the CA by week. The labels of this table do not correlate with those in figure 8.3.**

the CA to the mobile application. Alternatively, the system could interpret the answers and response times of the user to automatically adjust to this behavior. For example: if the user only replies to the breakfast topic, and does that every day at around 10 in the morning, the system should only send out this notification and change the time for the message to 10am. This improvement would solve the problem where study participants repeatedly had to tell the CA that they did not eat yet.

Additionally, the accuracy of the input understanding should be improved. Some rules were developed relatively general, in order to accept many different user inputs. Especially, when participants used more complex inputs (e.g. "more than 100 steps", "tomato soup"), the CA did not handle them very well. Here, either more specific rules need to be crafted to improve this functionality, or the user input needs to be limited to exclude too complex sentences. A mixture between these two options could also be discussed.

Some participants tried to take the initiative in the conversation. Since the CA was implemented to proactively approach the user, but not to answer questions, that led to several odd conversations (e.g. the CA stubbornly told the user to come back later). For this problem it is suggested to communicate the available features more clearly, and also explicitly address the limitations of the system.

Furthermore, the system should be increasingly improved upon the problems that can be observed in the chat logs. For example, during the study it was common for participants trying to add food on their incentive (not when the CA asked them about it). Since this functionality was not implemented, the users were rejected. In this case, this feature would need to be implemented in future versions, or otherwise, it should be clearly communicated that this feature is not available.

The ChatScript concept that was used for detecting food did not include all the names that were used by the participants. Although it was still possible to add these meals, the CA would tell the users repeatedly to confirm their input. Ideally, the system should automatically learn new types of food when they are entered. Alternatively, unknown food names could be added manually by developer, after scanning the chat log for rejected names.

### 8.3 Limitations

The number of seven study participants do not allow draw conclusions of statistical relevance. The results should be used to detect problems that appeared during the study, and to serve as a foundation for further research.

No control group was used in this study. This means that the results do not show whether the CA improved the experience compared to the original platform.

A part of the participants stated that they did not usually use software to track their lifestyle. This could have influenced the outcome, since they probably could not identify with a typical user of such a system.

The majority of the participants did not have English as their native language. The CA, nevertheless, was programmed to communicate in English. This might have had influence on the results, in case the participants had problems understanding the CA or expressing their wants.

## 8.4 Summary

The results of the CES show that *happiness* was the prevailing emotion when using the application. Nevertheless, also *anger* was felt regularly. The SUS showed that four out of seven participants rated the usability above the usable threshold of 70 points. The average score also exceeded this value. Five out of seven felt positively influenced by the CA to add more meals to the food diary; three of the participants felt that they added more steps than they would have otherwise.

The appearance of the prototype was perceived positively. Several points were part of criticism. The notifications sent by the CA were seen as sent too frequently, and received at inconvenient times. Incorrect or inaccurate input understanding occasionally led to inappropriate replies from the CA.

The study revealed potential for improvement. For further development, the notifications should be more personalized to each user. Furthermore, the accuracy of language understanding needs to be increased. In the next chapter, the lessons learned from this project are discussed.



## 9 | Lessons Learned

*This chapter summarizes the points that were learned during working on this thesis. This includes points about literature, implementation of the prototype, and results of the study.*

### 9.1 Literature

Compared to the area of eHealth, literature for CAs was relatively hard to find. Where in eHealth almost all of the literature seemed to agree on this term, for CAs, a variety of names were used. Often, research was carried out in parallel, using different terms, but working on similar topics. It appears that no real effort to combining these streams has been made so far.

A great part of the research concerning CAs is done by private people or companies. Most of the time, those do not care about sharing their knowledge with the public. Where some people involved with CAs just consider themselves as hobbyists (e.g. Steve Worswick, the inventor of Mitsukui) and do not want to spend time on documentation, companies – such as Apple Inc. or Google – keep their research secret to achieve competitive advantages. This leads to the problem that most state-of-the-art research is not available publicly.

The development of the CA technology has been slower than it was expected. After the development of ELIZA in 1966, researches expected CAs soon to be applicable. But still, CAs work in a similar way they did decades ago. Only the availability of huge data sources seems to allow a new way of development. This steers the source of knowledge used to empower the CA away from static to dynamic, self-learned rules. The development of this technology is put in the hands of those companies and institutions who have access to this data.

### 9.2 Implementation

It is hard to predict how the users are going to interact with the system. Users of the system often have different expectations than assumed when planning the system. Investigating the

needs and wants of the future users during the planning phase would have made it easier to prevent errors, and communicate the scope of the system.

Freedom of user input leads to great complexity. Deciding to accept all kinds of natural language input turned out to introduce multiple potential sources for errors. Especially, using a sentiment classifier to interpret the feelings of a user, regularly led to wrong results. Letting the user choose from multiple choices, would probably make the system easier to use.

Giving the initiative to the CA allowed to reduce the number of rules by a big amount. Since the CA did introduce the topics, the domains to talk about were strictly limited. Leaving the initiative to the user would require extensive research prior to the implementation, in order to craft replies to the most common topics and questions.

### 9.3 Outcome

The attitude of the study participants seemed to have an influence on the study outcome. For further studies it would probably make sense to select only participants that like to collect their data, or have to do this, because it was prescribed by a practitioner. This would also make the results easier to compare.

Unforeseen errors happen. Although the prototype was regularly tested on different devices, it turned out that for three out of ten people the implementation of the CA could not be accessed in the mobile application. For this reason, it turned out to be positive to ask more people to participate in the study than the number originally planned.

Always having a running prototype was very positive for the development process. Instead of developing several big components of the system independently and putting them together in the end, the prototype was continuously extended with new functionality. This helped to detect errors as soon as they appeared. Having a running prototype throughout development would also allow studies to be implemented in earlier stages.



## 10 | Conclusion and Future Work

*In this chapter the research questions of the thesis are summed up and outlook to future research is provided.*

### 10.1 Conclusion

The first question this thesis reached out to answer was which role a CA could take within an eHealth system. With respect to the knowledge gathered in the literature review, a concept for a CA that could be integrated into the existing Self eHealth platform was created. Several different roles of a CA were analyzed. The broad application area of CAs led to multiple potential candidates that could be implemented within the platform.

The concept of a CA as information source was dismissed due to time constraints; ideas from the concept of a CA for motivation diagnosis were partly transferred to the concept that was selected later on: a motivational CA. This concept involved a CA that proactively approaches a user in order to retrieve health information. Using the acquired data, motivational messages are generated and send to the user. For the interaction between the CA and the user, a UI was designed that resembles common messaging applications. This concept was implemented as a prototype and integrated into the Self eHealth platform, as part of the mobile application.

As a second question it was investigated how this prototype was perceived by users of the Self eHealth platform. For this reason, a study was implemented. Seven users were asked to use the prototype over the period of two weeks. During this duration, logs were kept for every interaction between the users and the system. After the two weeks, a questionnaire was sent out to the participants to investigate their emotions and the perceived usability of the system.

The results of the study concluded that users were mostly happy when interacting with the system, but the prototype also caused anger some of the time. The perceived usability varies strongly, where four participants see the system as good, and three rate it below the usable threshold of 70 points on the SUS. Also the usage statistics indicate that the system received mixed appraisal.

## 10.2 Future Work

Further research could focus on increased personalization of the CA. Learning from user behavior would make it possible to adjust the topics and timing of the messages according to the needs of the user.

Automatically improving the system could also be researched in future projects. Especially names of food cannot be stored in a complete list, and new terms often appear. A way to add unknown terms to the vocabulary of the system could further increase the usability of the system.

The actual impact of the implemented prototype needs to be tested in a larger scale study to retrieve more significant information. The study in this thesis suggest that the CA had positive effects on at least some of the study participants. A comparison to a control group, using the platform without a CA could provide further information.





## Bibliography

- Abdul-Kader, S. A., & Woods, J. (2015). Survey on Chatbot Design Techniques in Speech Conversation Systems. *IJACSA) International Journal of Advanced Computer Science and Applications*, 6(7), 72–80. Retrieved from [www.ijacsa.thesai.org](http://www.ijacsa.thesai.org)
- Ajami, S., & Bagheri-Tadi, T. (2013). Barriers for adopting electronic health records (EHRs) by physicians. *Acta Informatica Medica*, 21(2), 129–134. doi: 10.5455/aim.2013.21.129-134
- Ajzen, I. (1991, dec). The theory of planned behavior. *Organizational Behavior and Human Decision Processes*, 50(2), 179–211. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/074959789190020T> doi: 10.1016/0749-5978(91)90020-T
- Allen, J., Guinn, C., & Horvitz, E. (1999, sep). Mixed-initiative interaction. *IEEE Intelligent Systems*, 14(5), 14–23. Retrieved from <http://ieeexplore.ieee.org/document/796083/> doi: 10.1109/5254.796083
- Al-Zubaide, H., & Issa, A. A. (2011). OntBot: Ontology based ChatBot. *2011 4th International Symposium on Innovation in Information and Communication Technology, ISIICT'2011*, 7–12. doi: 10.1109/ISIICT.2011.6149594
- Anabuki, M., Kakuta, H., Yamamoto, H., & Tamura, H. (2000). Welbo. In *Chi '00 extended abstracts on human factors in computing systems - chi '00* (p. 10). New York, New York, USA: ACM Press. Retrieved from <http://portal.acm.org/citation.cfm?doid=633292.633299> doi: 10.1145/633292.633299
- Andersson, G., & Titov, N. (2014). Advantages and limitations of Internet-based interventions for common mental disorders. *World Psychiatry*, 13(1), 4–11. doi: 10.1002/wps.20083
- Andrews, P., Boni, M. D., Manandhar, S., & De, M. (2006). Persuasive Argumentation in Human Computer Dialogue. *AAAI Spring Symposium: Argumentation for Consumers of Healthcare*, 8–13. Retrieved from <http://www.aaai.org/Papers/Symposia/Spring/2006/SS-06-01/SS06-01-002.pdf>
- Apple Inc. (2011). *Apple Launches iPhone 4S, iOS 5 & iCloud*. Retrieved from <https://www.apple.com/pr/library/2011/10/04Apple-Launches-iPhone-4S-iOS-5-iCloud.html>

- Augello, A., Gentile, M., Weideveld, L., & Dignum, F. (2016). A Model of a Social Chatbot. In G. A. Tsihrintzis, E. Damiani, M. Virvou, R. J. Howlett, & L. C. Jain (Eds.), *Smart innovation, systems and technologies* (Vol. 6, pp. 637–647). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from [http://link.springer.com/10.1007/978-3-319-39345-2\\_57](http://link.springer.com/10.1007/978-3-319-39345-2_57) doi: 10.1007/978-3-319-39345-2\_57
- August, S. E. (2006). Personal Healthcare Assistant / Companion in Virtual World. *Communications of the ACM*, 41–42.
- Bandura, A. (1989). Human agency in social cognitive theory. *American Psychologist*, 44(9), 1175–1184. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/2782727> doi: 10.1037/0003-066X.44.9.1175
- Bandura, A. (2001, feb). Social Cognitive Theory: An Agentic Perspective. *Annual Review of Psychology*, 52(1), 1–26. Retrieved from <http://www.annualreviews.org/doi/10.1146/annurev.psych.52.1.1> doi: 10.1146/annurev.psych.52.1.1
- Bangor, A., Staff, T., Kortum, P., Miller, J., & Staff, T. (2009). Determining What Individual SUS Scores Mean : Adding an Adjective Rating Scale. *Journal of Usability Studies*, 4(3), 114–123.
- Barak, A., Hen, L., Boniel-Nissim, M., & Shapira, N. (2008). A Comprehensive Review and a Meta-Analysis of the Effectiveness of Internet-Based Psychotherapeutic Interventions. *Journal of Technology in Human Services*, 26(2-4), 109–160. Retrieved from <http://www.tandfonline.com/doi/abs/10.1080/15228830802094429> doi: 10.1080/15228830802094429
- Beal, V. (n.d.). *IPA - Intelligent Personal Assistant*. Retrieved 2017-05-08, from <http://www.webopedia.com/TERM/I/intelligent-personal-assistant.html>
- Becker, C., Kopp, S., & Wachsmuth, I. (2004). Simulating the Emotion Dynamics of a Multimodal Conversational Agent. In *Affective dialogue systems* (pp. 154–165). Retrieved from [http://link.springer.com/10.1007/978-3-540-24842-2\\_15](http://link.springer.com/10.1007/978-3-540-24842-2_15) doi: 10.1007/978-3-540-24842-2\_15
- Berner, E. S., & La Lande, T. J. (2007). Overview of Clinical Decision Support Systems. In *Decision support systems* (Vol. 6, pp. 3–22). Retrieved from [http://link.springer.com/10.1007/978-0-387-38319-4\\_1](http://link.springer.com/10.1007/978-0-387-38319-4_1) doi: 10.1007/978-0-387-38319-4\_1
- Bickmore, T., & Giorgino, T. (2006). Health dialog systems for patients and consumers. *Journal of Biomedical Informatics*, 39(5), 556–571. doi: 10.1016/j.jbi.2005.12.004
- Bickmore, T., Gruber, A., & Picard, R. (2005). Establishing the computer-patient working alliance in automated health behavior change interventions. *Patient Education and Counseling*, 59(1), 21–30. doi: 10.1016/j.pec.2004.09.008
- Bickmore, T. W., Caruso, L., Clough-Gorr, K., & Heeren, T. (2005). 'It's just like you talk to a friend' relational agents for older adults. *Interacting with Computers*, 17(6), 711–735. doi: 10.1016/j.intcom.2005.09.002

- Bickmore, T. W., Puskar, K., Schlenk, E. A., Pfeifer, L. M., & Sereika, S. M. (2010). Maintaining reality: Relational agents for antipsychotic medication adherence. *Interacting with Computers*, 22(4), 276–288. Retrieved from <http://dx.doi.org/10.1016/j.intcom.2010.02.001> doi: 10.1016/j.intcom.2010.02.001
- Bickmore, T. W., Utami, D., Matsuyama, R., & Paasche-Orlow, M. K. (2016, jan). Improving Access to Online Health Information With Conversational Agents: A Randomized Controlled Experiment. *Journal of Medical Internet Research*, 18(1), e1. Retrieved from <http://www.jmir.org/2016/1/e1/> doi: 10.2196/jmir.5239
- Bii, P. (2013). Chatbot technology: A possible means of unlocking student potential to learn how to learn. *Educational Research (ISSN: 2141-5161)*, 4(February), 218–221. Retrieved from <http://interesjournals.com/ER/pdf/2013/February/Bii.pdf>
- Bobrow, D. G., Kaplan, R. M., Kay, M., Norman, D. A., Thompson, H., & Winograd, T. (1977, apr). GUS, a frame-driven dialog system. *Artificial Intelligence*, 8(2), 155–173. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/0004370277900182> doi: 10.1016/0004-3702(77)90018-2
- Bohus, D., & Rudnicky, A. I. (2003). RavenClaw : Dialog Management Using Hierarchical Task Decomposition and an Expectation Agenda..
- Boogerd, E. A., Arts, T., Engelen, L. J., & van de Belt, T. H. (2015, mar). “What Is eHealth”: Time for An Update? *JMIR Research Protocols*, 4(1), e29. Retrieved from <http://www.ncbi.nlm.nih.gov/pubmed/25768939> doi: 10.2196/resprot.4065
- Bradeško, L., & Mladenčić, D. (2012). A Survey of Chabot Systems through a Loebner Prize Competition. *Researchgate.Net*, 2, 2–5. Retrieved from [http://www.researchgate.net/publication/235664166\\_A\\_Survey\\_of\\_Chatbot\\_Systems\\_through\\_a\\_Loebner\\_Prize\\_Competition/file/79e41512679b504a17.pdf](http://www.researchgate.net/publication/235664166_A_Survey_of_Chatbot_Systems_through_a_Loebner_Prize_Competition/file/79e41512679b504a17.pdf)
- Brooke, J. (1996). SUS - A quick and dirty usability scale. *Usability evaluation in industry*, 189(194), 4–7. Retrieved from <http://hell.meiert.org/core/pdf/sus.pdf> doi: 10.1002/hbm.20701
- Bull, S. (2012). *eHealth Applications*. Routledge. Retrieved from <http://www.tandfebooks.com/isbn/9780203149096> doi: 10.4324/9780203149096
- Bull, S., Pratte, K., Whitesell, N., Rietmeijer, C., & McFarlane, M. (2009). Effects of an internet-based intervention for HIV prevention: the youthnet trials. *AIDS and Behavior*, 13(3), 474–487. doi: 10.1007/s10461-008-9487-9
- Buller, D. B., & Floyd, A. H. L. (2012). Internet-based interventions for health behavior change. *eHealth Applications: Promising Strategies for Health Behavior Change*. New York: Taylor & Francis, 59–78.
- Busse, R., Blümel, M., Scheller-Kreinsen, D., & Zentner, A. (2010). *Tackling chronic disease in Europe: strategies, interventions, and challenges*. World Health Organization.

- Cassell, M. M., Jackson, C., & Cheuvront, B. (1998, jan). Health Communication on the Internet: An Effective Channel for Health Behavior Change? *Journal of Health Communication, 3*(1), 71–79. Retrieved from <https://www.ncbi.nlm.nih.gov/pubmed/10947375> doi: 10.1080/108107398127517
- Charras, F., Duplessis, G. D., Letard, V., Ligozat, A.-I., & Rosset, S. (2016). Comparing System-response Retrieval Models for Open-domain and Casual Conversational Agent. *Proceedings of the Second Workshop on Chatbots and Conversational Agent Technologies*.
- Choi, J., Noh, G.-Y., & Park, D.-J. (2014, feb). Smoking Cessation Apps for Smartphones: Content Analysis With the Self-Determination Theory. *Journal of Medical Internet Research, 16*(2), e44. Retrieved from <http://www.jmir.org/2014/2/e44/> doi: 10.2196/jmir.3061
- Chou, W. Y. S., Prestin, A., Lyons, C., & Wen, K. Y. (2013). Web 2.0 for health promotion: Reviewing the current evidence. *American Journal of Public Health, 103*(1), 9–18. doi: 10.2105/AJPH.2012.301071
- Chung, G. (2004). Developing a flexible spoken dialog system using simulation. *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, 63–71. Retrieved from <http://portal.acm.org/citation.cfm?id=1218964> doi: 10.3115/1218955.1218964
- Cisco. (2017). *Cisco Visual Networking Index (VNI) Update Global Mobile Data Traffic Forecast 2016–2021* (Tech. Rep.). Retrieved from <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html>
- Colby, K. M., Hilf, F. D., Weber, S., & Kraemer, H. C. (1972). Turing-Like Indistinguishability Tests for the Validation of a Computer Simulation of Paranoid Processes. *Artificial Intelligence, 3*, 199–221. Retrieved from <http://www.sciencedirect.com/science/article/pii/0004370272900495%5C> doi: 10.1016/0004-3702(72)90049-5
- Colby, K. M., Watt, J. B., & Gilbert, J. P. (1966). A computer method of psychotherapy: preliminary communication. *Journal of Nervous and Mental Disease, 42*(2), 148–152.
- Copple, K. L. (2009). Bringing AI to Life. In *Parsing the turing test* (pp. 359–376). Dordrecht: Springer Netherlands. Retrieved from [http://link.springer.com/10.1007/978-1-4020-6710-5\\_22](http://link.springer.com/10.1007/978-1-4020-6710-5_22) doi: 10.1007/978-1-4020-6710-5\_22
- Corti, K., & Gillespie, A. (2015). Offscreen and in the Chair Next to Your: Conversational Agents Speaking Through Actual Human Bodies. In H. Prendinger, J. Lester, & M. Ishizuka (Eds.), *Intelligent virtual agents* (Vol. 5208, pp. 405–417). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from [http://link.springer.com/10.1007/978-3-319-21996-7\\_44](http://link.springer.com/10.1007/978-3-319-21996-7_44) doi: 10.1007/978-3-319-21996-7\_44
- Crutzen, R., Peters, G. J. Y., Portugal, S. D., Fisser, E. M., & Grolleman, J. J. (2011). An artificially intelligent chat agent that answers adolescents' questions related to sex,



- drugs, and alcohol: An exploratory study. *Journal of Adolescent Health*, 48(5), 514–519. Retrieved from <http://dx.doi.org/10.1016/j.jadohealth.2010.09.002> doi: 10.1016/j.jadohealth.2010.09.002
- Cunningham, J. A., Wild, T. C., Cordingley, J., Van Mierlo, T., & Humphreys, K. (2009). A randomized controlled trial of an internet-based intervention for alcohol abusers. *Addiction*, 104(12), 2023–2032. doi: 10.1111/j.1360-0443.2009.02726.x
- de Gasperis, G. (2010). Building an AIML chatter bot knowledge-base starting from a FAQ and a glossary. *Journal of E-Learning and Knowledge Society*, 6(2), 75–83.
- DesRoches, C. M., Campbell, E. G., Rao, S. R., Donelan, K., Ferris, T. G., Jha, A., ... Blumenthal, D. (2008, jul). Electronic Health Records in Ambulatory Care — A National Survey of Physicians. *New England Journal of Medicine*, 359(1), 50–60. Retrieved from <http://www.nejm.org/doi/abs/10.1056/NEJMsa0802005> doi: 10.1056/NEJMsa0802005
- Edwards, B. I., Muniru, I. O., & Cheek, A. D. (2016). Robots to the Rescue: A Review of Studies on Differential Medical Diagnosis Employing Ontology-Based Chat Bot Technology. *Preprints*(December), 2–9. Retrieved from <http://www.preprints.org/manuscript/201612.0027/v1> doi: 10.20944/preprints201612.0027.v1
- Epstein, R. (1992). The Quest for the Thinking Computer. *AI Magazine*, 13(2), 80–95.
- Etherington, D. (2014). *Amazon Echo Is A \$199 Connected Speaker Packing An Always-On Siri-Style Assistant*. Retrieved 2017-05-09, from <https://techcrunch.com/2014/11/06/amazon-echo/>
- Etter, J.-F. (2005). *Comparing the Efficacy of Two Internet-Based, Computer-Tailored Smoking Cessation Programs: A Randomized Trial* (Vol. 7) (No. 1). doi: 10.2196/jmir.7.1.e2
- Eurostat. (2016). *European Health Interview Survey: Almost 1 adult in 6 in the EU is considered obese* (No. October). eurostat. Retrieved from <http://ec.europa.eu/eurostat/documents/2995521/7700898/3-20102016-BP-EN.pdf/c26b037b-d5f3-4c05-89c1-00bf0b98d646>
- Eurostat. (2017). *Overweight and obesity - BMI statistics*. Retrieved 2017-07-24, from [http://ec.europa.eu/eurostat/statistics-explained/index.php/Overweight\\_and\\_obesity\\_-\\_BMI\\_statistics](http://ec.europa.eu/eurostat/statistics-explained/index.php/Overweight_and_obesity_-_BMI_statistics)
- Eysenbach, G. (2001, jun). What is e-health? *Journal of Medical Internet Research*, 3(2), e20. Retrieved from <http://www.jmir.org/2001/2/e20/> doi: 10.2196/jmir.3.2.e20
- Galvao, A., Barros, F., Neves, A., & Ramalho, G. (2004). Persona-AIML: an architecture for developing chatterbots with personality. *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004*.(January), 1264–1265. doi: 10.1109/AAMAS.2004.242897
- Gardiner, P. M., McCue, K. D., Negash, L. M., Cheng, T., White, L. F., Yinusa-Nyahkoon, L., ... Bickmore, T. W. (2017). Engaging Women with an Embodied Conversational Agent to Deliver Mindfulness and Lifestyle Recommendations: A Feasibility Randomized Control Trial. *Patient Education and Counseling*. Retrieved from

<http://linkinghub.elsevier.com/retrieve/pii/S0738399117302495> doi:  
10.1016/j.pec.2017.04.015

- Glance, D. G., Ooi, E., Berman, Y., Glance, C. F., & Barrett, H. R. (2016). Impact of a Digital Activity Tracker-Based Workplace Activity Program on Health and Wellbeing. In *Proceedings of the 6th international conference on digital health conference - dh '16* (pp. 37–41). New York, New York, USA: ACM Press. Retrieved from <http://dl.acm.org/citation.cfm?id=2896338.2896345> doi: 10.1145/2896338.2896345
- Glasgow, R. E. (2007). eHealth Evaluation and Dissemination Research. *American Journal of Preventive Medicine*, 32(5 SUPPL.), 119–126. doi: 10.1016/j.amepre.2007.01.023
- Graesser, A. C., Moreno, K., Marineau, J., Adcock, A., Olney, A., & Person, N. (2003). AutoTutor improves deep learning of computer literacy: Is it the dialog or the talking head. *Proceedings of Artificial Intelligence in Education*, 47–54. Retrieved from [https://www.researchgate.net/publication/228760959\\_Auto-Tutor\\_Improves\\_Deep\\_Learning\\_of\\_Computer\\_Literacy\\_Is\\_it\\_the\\_Dialog\\_or\\_the\\_Talking\\_Head](https://www.researchgate.net/publication/228760959_Auto-Tutor_Improves_Deep_Learning_of_Computer_Literacy_Is_it_the_Dialog_or_the_Talking_Head)
- Graesser, A. C., VanLehn, K., Rosé, C. P., Jordan, P. W., & Harter, D. (2001). Systems with Conversational Dialogue with Conversational Dialogue. *AI Magazine*, 22(4), 39–52. Retrieved from <http://www.aaai.org/ojs/index.php/aimagazine/article/download/1591/1490> doi: <http://dx.doi.org/10.1609/aimag.v22i4.1591>
- Grolleman, J., van Dijk, B., Nijholt, A., & van Emst, A. (2006). Break the Habit! Designing an e-Therapy Intervention Using a Virtual Coach in Aid of Smoking Cessation. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 3962 LNCS, pp. 133–141). Retrieved from [http://link.springer.com/10.1007/11755494\\_19](http://link.springer.com/10.1007/11755494_19) doi: 10.1007/11755494\_19
- Guay, F., Vallerand, R. J., & Blanchard, C. (2000). On the Assessment of Situational Intrinsic and Extrinsic Motivation: The Situational Motivation Scale (SIMS). *Motivation and Emotion*, 24(3), 175–213. doi: 10.1023/A:1005614228250
- Gulenko, I. (2014). Chatbot for IT security training: Using motivational interviewing to improve security behaviour. In *Ceur workshop proceedings* (Vol. 1197, pp. 7–16).
- Hoerbst, A., & Ammenwerth, E. (2010, jul). Electronic Health Records. *Methods of Information in Medicine*, 49(4), 320–336. Retrieved from <http://www.schattauer.de/index.php?id=1214&doi=10.3414/ME10-01-0038> doi: 10.3414/ME10-01-0038
- Huang, J., Zhou, M., & Yang, D. (2007). Extracting chatbot knowledge from online discussion forums. *IJCAI International Joint Conference on Artificial Intelligence*, 423–428.
- Hutchens, J. (1997). *How to Pass the Turing Test by Cheating* (Tech. Rep.). University of Western Australia. Retrieved from [csee.umbc.edu](http://csee.umbc.edu)
- Hutto, C. J., & Gilbert, E. (2014). Vader: A parsimonious rule-based model for sentiment analysis of social media text. *Eighth international AAI conference on weblogs and*

- social media*, 216–225. Retrieved from  
<http://www.aaai.org/ocs/index.php/ICWSM/ICWSM14/paper/view/8109>
- Jha, A. K., DesRoches, C. M., Campbell, E. G., Donelan, K., Rao, S. R., Ferris, T. G., ... Blumenthal, D. (2009, apr). Use of Electronic Health Records in U.S. Hospitals. *New England Journal of Medicine*, 360(16), 1628–1638. Retrieved from  
<http://www.nejm.org/doi/abs/10.1056/NEJMsa0900592> doi:  
10.1056/NEJMsa0900592
- Jurafsky, D. (2017). *Conversational agents (slides)*. Retrieved from  
<http://web.stanford.edu/class/cs124/lec/chatbot17.pdf>
- Kawamoto, K., Houlihan, C. a., Balas, E. A., & Lobach, D. F. (2005). Improving clinical practice using clinical decision support systems: a systematic review of trials to identify features critical to success. *BMJ (Clinical research ed.)*, 330(March), 765. doi:  
10.1136/bmj.38398.500764.8F
- Kay, R. H., & Loverock, S. (2008). Assessing emotions related to learning new software: The computer emotion scale. *Computers in Human Behavior*, 24(4), 1605–1623. doi:  
10.1016/j.chb.2007.06.002
- Kennedy, C. M., Powell, J., Payne, T. H., Ainsworth, J., Boyd, A., & Buchan, I. (2012). Active assistance technology for health-related behavior change: An interdisciplinary review. *Journal of Medical Internet Research*, 14(3), 1–20. doi: 10.2196/jmir.1893
- Kenny, G. (2015). *I Know Everything About You! The Rise of the Intelligent Personal Assistant*. Retrieved 2017-05-08, from <https://securityintelligence.com/i-know-everything-about-you-the-rise-of-the-intelligent-personal-assistant/>
- Kerly, A., Ellis, R., & Bull, S. (2008). CALMsystem: A Conversational Agent for Learner Modelling. *Knowledge-Based Systems*, 21(3), 238–246. doi: 10.1016/j.knosys.2007.11.015
- Kerly, A., Hall, P., & Bull, S. (2007). Bringing chatbots into education: Towards natural language negotiation of open learner models. *Knowledge-Based Systems*, 20(2), 177–185. doi: 10.1016/j.knosys.2006.11.014
- Kerr, D. A., Pollard, C. M., Howat, P., Delp, E. J., Pickering, M., Kerr, K. R., ... Boushey, C. J. (2012, dec). Connecting Health and Technology (CHAT): protocol of a randomized controlled trial to improve nutrition behaviours using mobile devices and tailored text messaging in young adults. *BMC Public Health*, 12(1), 477. Retrieved from  
<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3438105/> doi:  
10.1186/1471-2458-12-477
- Kerry, A., Ellis, R., & Bull, S. (2009). Conversational Agents in E-Learning. In *Applications and innovations in intelligent systems xvi* (pp. 169–182). London: Springer London. Retrieved from [http://www.academia.edu/download/45475284/Conversational\\_Agents\\_in\\_E-Learning20160509-9244-1edaa0f.pdf](http://www.academia.edu/download/45475284/Conversational_Agents_in_E-Learning20160509-9244-1edaa0f.pdf) doi:  
10.1007/978-1-84882-215-3\_13
- Kiousis, S. (2002, sep). Interactivity: a concept explication. *New Media & Society*, 4(3), 355–383. Retrieved from

<http://nms.sagepub.com/cgi/doi/10.1177/146144402320564392> doi:  
10.1177/146144480200400303

- Kopp, S., Gesellensetter, L., Krämer, N. C., & Wachsmuth, I. (2005). A Conversational Agent as Museum Guide – Design and Evaluation of a Real-World Application. In *Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics)* (Vol. 3661 LNAI, pp. 329–343). Retrieved from [http://link.springer.com/10.1007/11550617\\_28](http://link.springer.com/10.1007/11550617_28) doi: 10.1007/11550617\_28
- Kreps, G. L., & Neuhauser, L. (2010). New directions in eHealth communication: Opportunities and challenges. *Patient Education and Counseling*, 78(3), 329–336. Retrieved from <http://dx.doi.org/10.1016/j.pec.2010.01.013> doi: 10.1016/j.pec.2010.01.013
- Kreuter, M. W., Strecher, V. J., & Glassman, B. (1999). One size does not fit all: the case for tailoring print materials. *Annals of behavioral medicine : a publication of the Society of Behavioral Medicine*, 21(4), 276–283. doi: 10.1007/BF02895958
- Kumar, R., & Rosé, C. P. (2011). Architecture for building conversational agents that support collaborative learning. *IEEE Transactions on Learning Technologies*, 4(1), 21–34. doi: 10.1109/TLT.2010.41
- Kumar, V., & Keerthana, A. (2016). Sanative Chatbot For Health Seekers. *International Journal Of Engineering And Computer Science*, 05(16022), 16022–16025. Retrieved from <http://ijecs.in/issue/v5-i3/28ijecs.pdf> doi: 10.18535/ijecs/v5i3.28
- Lau, C. (2014). *Why Cortana Assistant Can Help Microsoft in the Smartphone Market*. Retrieved 2017-05-09, from <https://www.thestreet.com/story/12534433/1/why-cortana-assistant-can-help-microsoft-in-the-smartphone-market.html>
- Lee, C., Jung, S., Kim, S., & Lee, G. G. (2009). Example-based dialog modeling for practical multi-domain dialog system. *Speech Communication*, 51(5), 466–484. Retrieved from <http://dx.doi.org/10.1016/j.specom.2009.01.008> doi: 10.1016/j.specom.2009.01.008
- Lee, S., & Choi, J. (2017). Enhancing user experience with conversational agent for movie recommendation: Effects of self-disclosure and reciprocity. *International Journal of Human-Computer Studies*, 103(May 2016), 95–105. Retrieved from <http://linkinghub.elsevier.com/retrieve/pii/S1071581917300198> doi: 10.1016/j.ijhcs.2017.02.005
- Lehto, T., & Oinas-Kukkonen, H. (2015). Examining the persuasive potential of web-based health behavior change support systems. *Transactions on Human-Computer Interaction*, 7(3), 126–140.
- Lester, J., & Mott, B. (2004). Conversational Agents. *The Practical Handbook of Internet Computing*, 220–240.
- Li, J., Monroe, W., Ritter, A., & Jurafsky, D. (2016). Deep Reinforcement Learning for Dialogue Generation. *arXiv*, 2(2), 1192–1202. Retrieved from <http://arxiv.org/abs/1606.01541> doi: 10.1103/PhysRevB.94.020410

- Lin, L., D'Haro, L. F., & Banchs, R. (2016). A Web-based Platform for Collection of Human-Chatbot Interactions. In *Proceedings of the fourth international conference on human agent interaction - hai '16* (pp. 363–366). New York, New York, USA: ACM Press. Retrieved from <http://dl.acm.org/citation.cfm?doid=2974804.2980500> doi: 10.1145/2974804.2980500
- Lokman, A. S., & Zain, J. M. (2009). An architectural design of virtual dietitian (ViDi) for diabetic patients. *Proceedings - 2009 2nd IEEE International Conference on Computer Science and Information Technology, ICCSIT 2009*(ViDi), 408–411. doi: 10.1109/ICCSIT.2009.5234671
- Lokman, A. S., & Zain, J. M. (2010). Extension and prerequisite: An algorithm to enable relations between responses in chatbot technology. *Journal of Computer Science*, 6(10), 1212–1218. doi: 10.3844/jcssp.2010.1212.1218
- López-Cózar, R., Callejas, Z., Griol, D., & Quesada, J. F. (2014). Review of spoken dialogue systems ; Sistemas de diálogo: una revisión. *Loquens*, 1(12), 1–15. Retrieved from <http://dx.doi.org/10.3989/loquens.2014.012> doi: 10.3989/loquens.2014.012
- Luckerson, V. (2016). *How Siri Got Left Behind*. Retrieved 2017-05-08, from <https://theringer.com/apple-siri-lost-assistant-race-2ffc86fe5dfa>
- Luperfoy, S. (2004). Retrofitting synthetic dialog agents to game characters for lifestyle risk training. *AAAI Fall Symposium - Technical Report, FS-04-04*, 78–87. Retrieved from <http://www.scopus.com/inward/record.url?eid=2-s2.0-32844467386&partnerID=40&md5=b12afd3ff9248aedee2406a9a9351eb1>
- Lynley, M. (2016). *Google unveils Google Assistant, a virtual assistant that's a big upgrade to Google Now*. Retrieved 2017-05-08, from <https://techcrunch.com/2016/05/18/google-unveils-google-assistant-a-big-upgrade-to-google-now/>
- Mikic, F. A., Burguillo, J. C., Llamas, M., Rodríguez, D. A., & Rodríguez, E. (2009). CHARLIE: An AIML-based chatterbot which works as an interface among INES and humans. *20th EAEEIE Annual Conference, EAEEIE 2009 - Formal Proceedings*(September). doi: 10.1109/EAEEIE.2009.5335493
- Mikolov, T., Karafiat, M., Burget, L., Cernocky, J., & Khudanpur, S. (2010). Recurrent Neural Network based Language Model. *Interspeech*(September), 1045–1048.
- Musen, M. A., Shahar, Y., & Shortliffe, E. H. (2006). Clinical Decision-Support Systems. In E. H. Shortliffe & J. J. Cimino (Eds.), *Springer* (pp. 698–736). New York, NY: Springer New York. Retrieved from [http://link.springer.com/10.1007/0-387-36278-9\\_20](http://link.springer.com/10.1007/0-387-36278-9_20) doi: 10.1007/0-387-36278-9\_20
- NLTK Project. (2017). *nlk.org*. Retrieved 2017-07-19, from <http://www.nltk.org/>
- Noar, S. M., & Harrington, N. G. (2012a). Computer-tailored interventions for improving health behaviors. *eHealth Applications: Promising Strategies for Behavior Change*, 128–46.

- Noar, S. M., & Harrington, N. G. (2012b). eHealth Applications: An Introduction and Overview. In *ehealth applications: Promising strategies for behavior change* (pp. 3–16). New York: Routledge.
- Norman, G. J., Zabinski, M. F., Adams, M. A., Rosenberg, D. E., Yaroch, A. L., & Atienza, A. A. (2007). A Review of eHealth Interventions for Physical Activity and Dietary Behavior Change. *American Journal of Preventive Medicine*, 33(4). doi: 10.1016/j.amepre.2007.05.007
- OECD/EU. (2016). *Health at a Glance: Europe 2016: State of Health in the EU Cycle*. Paris: OECD Publishing. Retrieved from [http://www.oecd-ilibrary.org/social-issues-migration-health/health-at-a-glance-europe-2016\\_9789264265592-en](http://www.oecd-ilibrary.org/social-issues-migration-health/health-at-a-glance-europe-2016_9789264265592-en) doi: 10.1787/9789264265592-en
- Office of Disease Prevention and Health Promotion. (2000). *Healthy People 2010* (No. 2). Washington, DC.
- Oh, H., Rizo, C., Enkin, M., Jadad, A., Powell, J., & Pagliari, C. (2005, feb). What Is eHealth (3): A Systematic Review of Published Definitions. *Journal of Medical Internet Research*, 7(1), 1–12. Retrieved from <http://www.jmir.org/2005/1/e1/> doi: 10.2196/jmir.7.1.e1
- Oinas-Kukkonen, H., & Harjumaa, M. (2008). Towards deeper understanding of persuasion in software and information systems. *Proceedings of the 1st International Conference on Advances in Computer-Human Interaction, ACHI 2008*(March), 200–205. doi: 10.1109/ACHI.2008.31
- Oinas-Kukkonen, H., & Harjumaa, M. (2009). Persuasive Systems Design : Key Issues , Process Model , and System Features. *Communications of the Association for Information Systems*, 24(1).
- op den Akker, H., Jones, V. M., & Hermens, H. J. (2014). Tailoring real-time physical activity coaching systems: a literature survey and model. *User Modeling and User-Adapted Interaction*, 24(5), 351–392. doi: 10.1007/s11257-014-9146-y
- Pajares, F., Prestin, A., & Nabi, R. L. (2009). Social cognitive theory and mass media effects. *The SAGE Handbook of Media Processes and Effects*, 283–297.
- Pandas Project. (2017). *pandas.pydata.org*. Retrieved 2017-07-19, from <http://pandas.pydata.org/>
- Papaioannou, I., & Lemon, O. (2017). Combining Chat and Task-Based Multimodal Dialogue for More Engaging HRI. *Proceedings of the Companion of the 2017 ACM/IEEE International Conference on Human-Robot Interaction - HRI '17*, 365–366. Retrieved from <http://dl.acm.org/citation.cfm?doid=3029798.3034820> doi: 10.1145/3029798.3034820
- Poggi, I., Pelachaud, C., de Rosis, F., Carofiglio, V., & De Carolis, B. (2005). Greta. A Believable Embodied Conversational Agent. In (pp. 3–25). Retrieved from [http://link.springer.com/10.1007/1-4020-3051-7\\_1](http://link.springer.com/10.1007/1-4020-3051-7_1) doi: 10.1007/1-4020-3051-7\_1

- Prochaska, J. O., & Velicer, W. F. (1997, sep). The Transtheoretical Model of Health Behavior Change. *American Journal of Health Promotion*, 12(1), 38–48. Retrieved from <http://ajhpcontents.org/doi/abs/10.4278/0890-1171-12.1.38> doi: 10.4278/0890-1171-12.1.38
- Radlinski, F., & Craswell, N. (2017). A Theoretical Framework for Conversational Search. In *Proceedings of the 2017 conference on conference human information interaction and retrieval - chiir '17* (pp. 117–126). New York, New York, USA: ACM Press. Retrieved from <http://dl.acm.org/citation.cfm?doid=3020165.3020183> doi: 10.1145/3020165.3020183
- Radziwill, N. M., & Benton, M. C. (2017, apr). Evaluating Quality of Chatbots and Intelligent Conversational Agents. *ArXiv e-prints*. Retrieved from <https://arxiv.org/abs/1704.04579><http://arxiv.org/abs/1704.04579>
- Raux, A., & Eskenazi, M. (2009). A finite-state turn-taking model for spoken dialog systems. *Proceedings of Human Language Technologies The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics on NAACL 09*(June), 629–637. Retrieved from <http://portal.acm.org/citation.cfm?doid=1620754.1620846> doi: 10.3115/1620754.1620846
- Raux, A., Langner, B., & Bohus, D. (2005). Let's go public! taking a spoken dialog system to the real world. In *Proc. of interspeech 2005* (pp. 885–888). Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.71.5597>
- Riedman, P., Venable, J. R., Chang, V., Reiners, T., & Gütl, C. (2013). RIVALE: A Prototype Realistic Immersive Virtual Agent-based Learning Environment Case Study for Learning Requirements Elicitation Skills. In (pp. 1–15).
- Rigg, J. (2017). *UK health service to trial chatbot that gives medical advice*. Retrieved 2017-05-11, from <https://www.engadget.com/2017/01/05/nhs-chatbot/>
- Ritter, A., Cherry, C., & Dolan, W. B. (2011). Data-driven response generation in social media. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'11)*, 583–593. Retrieved from <http://dl.acm.org/citation.cfm?id=2145500>
- Ritterband, L. M., Thorndike, F. P., Gonder-Frederick, L. A., Magee, J. C., Bailey, E. T., Saylor, D. K., & Morin, C. M. (2009). Efficacy of an Internet-based behavioral intervention for adults with insomnia. *Archives of general psychiatry*, 66(7), 692–8. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3723339&tool=pmcentrez&rendertype=abstract> doi: 10.1001/archgenpsychiatry.2009.66
- Rojas-Barahona, L. M., & Giorgino, T. (2009). Adaptable dialog architecture and runtime engine (AdaRTE): A framework for rapid prototyping of health dialog systems. *International Journal of Medical Informatics*, 78(SUPPL. 1), 56–68. doi: 10.1016/j.ijmedinf.2008.07.017

- Rosenstock, I. M. (1974, dec). Historical Origins of the Health Belief Model. *Health Education & Behavior*, 2(4), 328–335. Retrieved from <http://heb.sagepub.com/content/2/4/328.extract> doi: 10.1177/109019817400200403
- Santangelo, a., Augello, A., Gentile, A., Pilato, G., & Gaglio, S. (2006). A Chat-bot based Multimodal Virtual Guide for Cultural Heritage Tours. *Psc*, 1–4. Retrieved from [http://www.researchgate.net/publication/221228353\\_A\\_Chat-bot\\_based\\_Multimodal\\_Virtual\\_Guide\\_for\\_Cultural\\_Heritage\\_Tours/file/9fcfd514087cca7ce2.pdf](http://www.researchgate.net/publication/221228353_A_Chat-bot_based_Multimodal_Virtual_Guide_for_Cultural_Heritage_Tours/file/9fcfd514087cca7ce2.pdf)
- Saygin, P., Cicekli, I., & Akman, V. (2000). Turing test: 50 years later. *Minds and Machines*, 10(4), 463–518. Retrieved from <http://cogprints.org/1925/2/tt50.ps>
- Schumaker, R. P., & Chen, H. (2010). Interaction analysis of the ALICE chatterbot: A two-study investigation of dialog and domain questioning. *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, 40(1), 40–51. doi: 10.1109/TSMCA.2009.2029603
- Serban, I. V., Sordoni, A., Bengio, Y., Courville, A., & Pineau, J. (2016). Building End-To-End Dialogue Systems Using Generative Hierarchical Neural Network Models. *AAAI*, 8. Retrieved from <http://arxiv.org/abs/1507.04808>
- Shawar, B., & Atwell, E. (2002). A comparison between ALICE and Elizabeth chatbot systems. *Computer*(December). Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.3.4703&rep=rep1&type=pdf>
- Shawar, B., & Atwell, E. (2003). Using dialogue corpora to train a chatbot. *Proceedings of the Corpus Linguistics 2003*, 681–690. Retrieved from <http://ucrel.lancs.ac.uk/publications/CL2003/papers/shawar.pdf> doi: 10.13140/2.1.1455.7122
- Shawar, B. A., & Atwell, E. (2005a). *A Corpus Based Approach to Generalising a Chatbot System*.
- Shawar, B. A., & Atwell, E. S. (2005b). Using corpora in machine-learning chatbot systems. *International Journal of Corpus Linguistics*, 10(4), 489–516. doi: 10.1075/ijcl.10.4.06sha
- Showell, C., & Nøhr, C. (2012). How should we define eHealth, and does the definition matter? *Studies in Health Technology and Informatics*, 180(April 2017), 881–884. doi: 10.3233/978-1-61499-101-4-881
- Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., ... Dolan, W. B. (2015). A Neural Network Approach to Context-Sensitive Generation of Conversational Responses. *Naacl-2015*, 196–205.
- Spek, V., Cuijpers, P., Nyklíček, I., Riper, H., Keyzer, J., & Pop, V. (2007). Internet-based cognitive behaviour therapy for symptoms of depression and anxiety: a meta-analysis. *Psychological medicine*, 37(3), 319–328. doi: 10.1017/S0033291706008944
- Spittaels, H., De Bourdeaudhuij, I., Brug, J., & Vandelanotte, C. (2006, aug). Effectiveness of an online computer-tailored physical activity intervention in a real-life setting. *Health Education Research*, 22(3), 385–396. Retrieved from



- <https://academic.oup.com/her/article-lookup/doi/10.1093/her/cyl096>  
doi: 10.1093/her/cyl096
- Stajkovic, A. D., & Luthans, F. (2003). Social cognitive theory and self-efficacy: Implications for motivation theory and practice. *Motivation and Work Behavior*(March), 126–140. doi: 10.1016/0024-6301(93)90245-B
- Stekl, J. (2012). *Android 4.1 Jelly Bean Now Official: Project Butter, Better Notifications, Google Now, And More*. Retrieved 2017-05-08, from <http://www.androidpolice.com/2012/06/27/android-4-1-jelly-bean-now-official-project-butter-android-bean-google-now-and-more/>
- Suendermann-Oeft, D., Ramanarayanan, V., Teckenbrock, M., Neutatz, F., & Schmidt, D. (2015). HALEF: An Open-Source Standard-Compliant Telephony-Based Modular Spoken Dialog System: A Review and An Outlook. In *Natural language dialog systems and intelligent assistants* (pp. 53–61). Cham: Springer International Publishing. Retrieved from [http://link.springer.com/10.1007/978-3-319-19291-8\\_5](http://link.springer.com/10.1007/978-3-319-19291-8_5) doi: 10.1007/978-3-319-19291-8\_5
- Tan, J. T. C., Duan, F., & Inamura, T. (2012). Multimodal human-robot interaction with Chatterbot system: Extending AIML towards supporting embodied interactions. *2012 IEEE International Conference on Robotics and Biomimetics, ROBIO 2012 - Conference Digest*, 1727–1732. doi: 10.1109/ROBIO.2012.6491217
- Tegos, S., Demetriadis, S., & Karakostas, A. (2015). Promoting academically productive talk with conversational agent interventions in collaborative learning settings. *Computers and Education*, 87, 309–325. Retrieved from <http://dx.doi.org/10.1016/j.compedu.2015.07.014> doi: 10.1016/j.compedu.2015.07.014
- Tremblay, M. A., Blanchard, C. M., Taylor, S., Pelletier, L. G., & Villeneuve, M. (2009). Work Extrinsic and Intrinsic Motivation scale: its value for organizational psychology research. *Canadian Journal of Behavioural Science*, 41(4), 213–226. doi: 10.1037/a0015167
- Turing, A. M. (1950). *Computing machinery and intelligence* (Vol. 59) (No. 236).
- Utami, D., Barry, B., Bickmore, T., Paasche-orlow, M., Ave, M., & Ave, H. (2013). *A Conversational Agent-based Clinical Trial Search Engine*.
- Van Eeuwen, M., & Van Der Kaap, H. (2017). *Mobile conversational commerce: messenger chatbots as the next interface between businesses and consumers*. Retrieved from <http://essay.utwente.nl/71706/>
- Vincent, J. (2016). *Baidu launches medical chatbot to help Chinese doctors diagnose patients*. Retrieved 2017-05-11, from <https://www.theverge.com/2016/10/11/13240434/baidu-medical-chatbot-china-melody>
- Vincent, J. (2017). *The UK's National Health Service is testing out a medical chatbot as a non-emergency helpline*. Retrieved from <https://www.theverge.com/2017/1/4/14168312/uk-nhs-babylon-medical-chatbot-helpline>

- Vrajitoru, D. (2003). Evolutionary Sentence Building for Chatterbots. *GECCO 2003 Late Breaking Papers*, 315–321. Retrieved from <http://citeseer.ist.psu.edu/596573.html>
- Wallace, R. S. (2009). The Anatomy of A . L . I . C . E . In *Parsing the turing test* (pp. 181–210). Dordrecht: Springer Netherlands. Retrieved from [http://link.springer.com/10.1007/978-1-4020-6710-5\\_13](http://link.springer.com/10.1007/978-1-4020-6710-5_13) doi: 10.1007/978-1-4020-6710-5\_13
- Wantland, D. J., Portillo, C. J., Holzemer, W. L., Slaughter, R., & McGhee, E. M. (2004, nov). The Effectiveness of Web-Based vs. Non-Web-Based Interventions: A Meta-Analysis of Behavioral Change Outcomes. *Journal of Medical Internet Research*, 6(4), e40. Retrieved from <http://www.jmir.org/2004/4/e40/> doi: 10.2196/jmir.6.4.e40
- Watson, A., Bickmore, T., Cange, A., Kulshreshtha, A., & Kvedar, J. (2012). An internet-based virtual coach to promote physical activity adherence in overweight adults: Randomized controlled trial. *Journal of Medical Internet Research*, 14(1), 1–12. doi: 10.2196/jmir.1629
- Watson, R. (2004). EU wants every member to develop a "roadmap" for ehealth. *BMJ*, 328(7449), 1155–1155. Retrieved from <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=411133&tool=pmcentrez&rendertype=abstract> doi: 10.1136/bmj.328.7449.1155-b
- Webb, T. L., Joseph, J., Yardley, L., & Michie, S. (2010, feb). Using the internet to promote health behavior change: a systematic review and meta-analysis of the impact of theoretical basis, use of behavior change techniques, and mode of delivery on efficacy. *Journal of medical Internet research*, 12(1), e4. Retrieved from <http://www.jmir.org/2010/1/e4/> doi: 10.2196/jmir.1376
- Weizenbaum, J. (1966). ELIZA: A computer program for the study of natural language communication between man and machine. *Communications of the Association for Computing Machinery*, 9(1), 36–45. Retrieved from <http://dl.acm.org/citation.cfm?id=357980.357991> doi: 10.1145/365153.365168
- Weizenbaum, J. (1976). *Computer Power and Human Reason: From Judgement to Calculation*. New York, NY, USA: W. H. Freeman & Co.
- Welch, C. (2014). *Google Now finally comes to desktop Chrome browser*. Retrieved 2017-05-08, from <https://www.theverge.com/apps/2014/3/24/5542994/google-now-finally-comes-to-desktop-chrome-browser>
- Whalen, T. (1994). *My Experience with the 1994 Loebner Competition*. Retrieved 2017-03-22, from [http://hps.elte.hu/~sim\\$gk/Loebner/story94.htm](http://hps.elte.hu/~sim$gk/Loebner/story94.htm)
- Whalen, T. (1995). *Thom's Participation in the Loebner Competition 1995*. Retrieved 2017-03-22, from [http://hps.elte.hu/~sim\\$gk/Loebner/story95.htm](http://hps.elte.hu/~sim$gk/Loebner/story95.htm)
- Whitney, L. (2015). *Microsoft's Cortana crosses over to iOS and Android*. Retrieved from <https://www.cnet.com/news/microsoft-to-expand-cortana-voice-assistant-to-ios-and-android/>

- Whitson, J. R. (2013). Gaming the quantified self. *Surveillance and Society*, 11(1-2), 163–176. doi: 10.1089/big.2012.0002
- Wilcox, B. (2008). *Beyond AIML: Chatbots 102*. Retrieved 2017-03-24, from [http://www.gamasutra.com/view/feature/132155/beyond\\_aiml\\_chatbots\\_102.php](http://www.gamasutra.com/view/feature/132155/beyond_aiml_chatbots_102.php)
- Wilcox, B. (2011). Beyond Façade: Pattern Matching for Natural Language Applications. *Gamasutra*, 1–5. Retrieved from [http://www.gamasutra.com/view/feature/6305/beyond\\_façade\\_pattern\\_matching\\_.php?page=1](http://www.gamasutra.com/view/feature/6305/beyond_façade_pattern_matching_.php?page=1)
- Wilcox, B. (2016). *ChatScript Basic User Manual*.
- Wilcox, B., & Wilcox, S. (2010). *Suzette, the Most Human Computer*.
- Wilcox, B., & Wilcox, S. (2013). Making it real: Loebner-winning chatbot design. *Arbor*, 189(764), a086. Retrieved from <http://arbor.revistas.csic.es/index.php/arbor/article/view/1888/2079> doi: 10.3989/arbor.2013.764n6009
- World Health Organization. (2015). *Third Global Survey on eHealth The use of eHealth in support of universal health coverage*.
- World Health Organization. (2017). *Obesity: Data and Statistics*. Retrieved 2017-07-25, from <http://www.euro.who.int/en/health-topics/noncommunicable-diseases/obesity/data-and-statistics>
- Xu, A., Liu, Z., Guo, Y., Sinha, V., & Akkiraju, R. (2017). A New Chatbot for Customer Service on Social Media. In *Proceedings of the 2017 chi conference on human factors in computing systems - chi '17* (pp. 3506–3510). New York, New York, USA: ACM Press. Retrieved from <http://dl.acm.org/citation.cfm?doid=3025453.3025496> doi: 10.1145/3025453.3025496
- Xuetao, M., Sansonnet, J.-P., & Bouchet, F. (2008). *A corpus-based NLP-chain for a web-based Assisting Conversational Agent*.
- Yan, Z., Duan, N., Bao, J., Chen, P., Zhou, M., Li, Z., & Zhou, J. (2016). DocChat: An Information Retrieval Approach for Chatbot Engines Using Unstructured Documents. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 516–525. Retrieved from <http://www.aclweb.org/anthology/P16-1049>
- Yu, Z., Papangelis, A., & Rudnicky, A. (2015). TickTock: A Non-Goal-Oriented Multimodal Dialog System with Engagement Awareness. In (pp. 108–111).
- Yu, Z., Xu, Z., Black, A. W., & Rudnicky, A. I. (2016). Chatbot Evaluation and Database Expansion via Crowdsourcing. *WOCHAT workshop*.



# Appendix

## A Terms in Literature including References

Term	Frequency	References
Chatbot	52	
- Chatbot	38	<i>Lokman and Zain (2009), Yan et al. (2016), Radlinski and Craswell (2017) Jurafsky (2017), Corti and Gillespie (2015), Wallace (2009), C. Lee, Jung, Kim, and Lee (2009), Augello et al. (2016), Lin et al. (2016), Yu, Xu, Black, and Rudnicky (2016), Xu et al. (2017), Kerly et al. (2007), Kerly et al. (2008), Radziwill and Benton (2017), Van Eeuwen and Van Der Kaap (2017), Al-Zubaide and Issa (2011), Bradeško and Mladenić (2012), Bii (2013), Edwards et al. (2016), Papaioannou and Lemon (2017), Li, Monroe, Ritter, and Jurafsky (2016), Andrews et al. (2006), Santangelo, Augello, Gentile, Pilato, and Gaglio (2006), Xuetao, Sansonnet, and Bouchet (2008), B. Shawar and Atwell (2002), Lokman and Zain (2010), Kerry, Ellis, and Bull (2009), Riedman, Venable, Chang, Reiners, and Gütl (2013), Gulenko (2014), B. A. Shawar and Atwell (2005b), Abdul-Kader and Woods (2015), B. Shawar and Atwell (2003), V. Kumar and Keerthana (2016), Huang, Zhou, and Yang (2007), B. A. Shawar and Atwell (2005a).</i>
- Chatterbot	14	<i>Ritter et al. (2011), Kopp, Gesellensetter, Krämer, and Wachsmuth (2005), Charras et al. (2016), Yu et al. (2015), Serban, Sordoni, Bengio, Courville, and Pineau (2016), Copple (2009), Vrajitoru (2003), Lokman and Zain (2010), Galvao, Barros, Neves, and Ramalho (2004), Mikic et al. (2009), de Gasperis (2010), Tan et al. (2012), Schumaker and Chen (2010)</i>

Term	Frequency	References
Conversational Agent	47	
– <i>Conversational Agent</i>	32	<i>Radlinski and Craswell (2017), Jurafsky (2017), Corti and Gillespie (2015), Raux and Eskenazi (2009), T. W. Bickmore et al. (2010), Augello et al. (2016), Lin et al. (2016), Kerly et al. (2008), Kerly et al. (2007), Tegos, Demetriadis, and Karakostas (2015), Kopp et al. (2005), Radziwill and Benton (2017), S. Lee and Choi (2017), Van Eeuwen and Van Der Kaap (2017), Charras et al. (2016), Yu et al. (2015), Al-Zubaide and Issa (2011), Bradeško and Mladenčić (2012), Edwards et al. (2016), Serban et al. (2016), Papaioannou and Lemon (2017), Graesser et al. (2003), Li et al. (2016), Lester and Mott (2004), Kerry et al. (2009), Mikic et al. (2009), de Gasperis (2010), Graesser, VanLehn, Rosé, Jordan, and Harter (2001), B. Shawar and Atwell (2003), Huang et al. (2007), B. A. Shawar and Atwell (2005a)</i>
– <i>Embodied Conversational Agent</i>	13	<i>Grolleman et al. (2006), T. Bickmore and Giorgino (2006), Kennedy et al. (2012), T. Bickmore, Gruber, and Picard (2005), Kennedy et al. (2012), T. Bickmore et al. (2005), T. W. Bickmore, Caruso, Clough-Gorr, and Heeren (2005), Anabuki, Kakuta, Yamamoto, and Tamura (2000), Poggi, Pelachaud, de Rosis, Carofiglio, and De Carolis (2005), Utami et al. (2013), Kopp et al. (2005), op den Akker, Jones, and Hermens (2014), López-Cózar, Callejas, Griol, and Quesada (2014), Xuetao et al. (2008), August (2006)</i>
– <i>Multimodal Conversational Agent</i>	1	<i>Becker et al. (2004)</i>
– <i>Assisting Conversational Agent</i>	1	<i>Xuetao et al. (2008)</i>
Dialog System	33	
– <i>Dialog System</i>	22	<i>T. Bickmore and Giorgino (2006), Jurafsky (2017), Sordoni et al. (2015), Ritter et al. (2011), Kennedy et al. (2012), C. Lee et al. (2009), T. W. Bickmore et al. (2005), T. W. Bickmore et al. (2010), Yu et al. (2016), Bobrow et al. (1977), Utami et al. (2013), op den Akker et al. (2014), Charras et al. (2016), Yu et al. (2015), Serban et al. (2016), R. Kumar and Rosé (2011), Li et al. (2016), Santangelo et al. (2006), Kerry et al. (2009), B. A. Shawar and Atwell (2005b), Abdul-Kader and Woods (2015), Tan et al. (2012)</i>
– <i>Spoken Dialog System</i>	10	<i>Radlinski and Craswell (2017), Raux, Langner, and Bohus (2005), Bohus and Rudnicky (2003), Raux and Eskenazi (2009), Rojas-Barahona and Giorgino (2009), Chung (2004), Luperfoy (2004), López-Cózar et al. (2014), Suendermann-Oeft, Ramanarayanan, Teckenbrock, Neutatz, and Schmidt (2015), Papaioannou and Lemon (2017)</i>
– <i>Natural Language Dialog System</i>	1	<i>Schumaker and Chen (2010)</i>
Total	132	

## B Questionnaire

### Part 1

While using the Self eHealth App, I felt

	None of the time	Some of the time	Most of the time	All of the time
satisfied	0	0	0	0
disheartened	0	0	0	0
anxious	0	0	0	0
irritable	0	0	0	0
excited	0	0	0	0
dispirited	0	0	0	0
insecure	0	0	0	0
frustrated	0	0	0	0
curious	0	0	0	0
helpless	0	0	0	0
nervous	0	0	0	0
angry	0	0	0	0

## Part 2

	strongly disagree	disagree	undecided	agree	strongly agree
I think that I would like to use this system frequently.	0	0	0	0	0
I found the system unnecessarily complex.	0	0	0	0	0
I thought the system was easy to use.	0	0	0	0	0
I think that I would need the support of a technical person to be able to use this system.	0	0	0	0	0
I found the various functions in this system were well integrated.	0	0	0	0	0
I thought there was too much inconsistency in this system.	0	0	0	0	0
I would imagine that most people would learn to use this system very quickly.	0	0	0	0	0
I found the system very cumbersome to use.	0	0	0	0	0
I felt very confident using the system.	0	0	0	0	0
I needed to learn a lot of things before I could get going with this system.	0	0	0	0	0



**Part 3**

Please rate how you felt when working with “Sebo” (the chatbot).

	strongly disagree	disagree	undecided	agree	strongly agree
“Sebo” motivated me to add more food to the diary.	0	0	0	0	0
“Sebo” motivated me to add more activities/steps.	0	0	0	0	0
“Sebo” improved the experience of the system.	0	0	0	0	0
“Sebo” contacted me too often.	0	0	0	0	0

What did you like when using Sebo?

---



---



---

What did you dislike when using Sebo?

---



---



---

Will you continue using the Self eHealth Platform?

- yes
- no

Why?

---



---



---

**Thank you for your participation!**