Enhancing Open-Source Content Management Systems to Semi-automatically Generate Web-based and Platform-native Mobile Applications

Master's Thesis

Submitted by:
Korab Qerkini

Institute for Information Systems and Computer Media
Graz University of Technology
A-8010 Graz, Austria

Supervisor:
Univ.Doz. Dipl.-Ing. Dr. techn. Christian Gütl

Co-supervisor:
Dipl. Ing. Dr. techn. Victor Manuel Garcia Barrios

November 2011

Erweiterung von Open-Source Content Management Systemen zur Semi-automatischen Erstellung von Web-basierenden und Plattform-nativen Mobilen Applikationen

Masterarbeit

Verfasst von:
Korab Qerkini

Institut für Informationssysteme und Neue Medien
Technische Universität Graz
A-8010 Graz, Österreich

Betreuer:
Univ.Doz. Dipl.-Ing. Dr. techn. Christian Gütl

Co-Betreuer:
Dipl. Ing. Dr. techn. Victor Manuel Garcia Barrios

November 2011

STATUTORY DECLARATION

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

…… ……..

date (signature)

EIDESSTATTLICHE ERKLÄRUNG

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

Graz, am …… .....……..

(Unterschrift)

Abstract

The penetration and the demand of mobile devices in our daily life, their constantly growing technical capabilities and the heterogeneity of mobile technologies are increasing in an unstoppable manner. Companies are facing the problem on which mobile technology to use for delivering content, covering as many mobile devices as possible. The development of mobile applications for the distinct devices of the same vendor or manufacturer (e.g. tablets and smartphones) is distinct as well. This thesis addresses the aforementioned problem moving towards a low-cost, easy-to-use and cross-platform software solution, which enables non-programmers to semi-automatically build mobile native and mobile Web applications. Further, this thesis aims at bearing the challenges of coping with the heterogeneity in the mobile devices market area by compiling template-based cross-platform mobile applications.

The target of this thesis is to present a solution for a reference design of a reusable, flexible and modular Mobile Application Publishing Platform (MAPP) system architecture. Therefore, different mobile application frameworks and Web based mobile application platforms will be analyzed. In addition, to take advantage of established Web user experience, the proposed solution should build on an existing open source Content Management System (CMS) and enhance it for the given goals. Further this thesis is aimed at designing and implementing the mobile application development process to efficiently support non-programmers in creating mobile applications. In order to provide a reusable, reliable, efficient, maintainable, portable and extensible system, the solution approach follows well-established software technologies and diverse application-specific design patterns.

To provide a solution approach to a real-world example, a customer expectation evaluation is conducted in order to capture the straightforwardness of the MAPP and identify to which extent the resulting mobile applications cope with the user's expectations.

Zusammenfassung

Der Bedarf und die Vielfalt an mobilen Geräten, als auch die Nachfrage nach mobilen Applikationen, steigen unaufhaltsam. Unternehmen sehen sich mit der Herausforderung konfrontiert, sich für eine mobile Technologie zu entscheiden um spezifische Inhalte optimiert auf den verschiedenen mobilen Geräten darstellen zu können. Hierbei ist die Entwicklung eigener mobiler Applikationen mit entsprechendem zeitlichen Aufwand, Knowhow und Kosten verbunden. In Anlehnung an den vorherrschenden Bedarf nach einer kostengünstigen Lösung, richtet sich diese Arbeit an die obenerwähnte Herausforderung, indem eine leicht handzuhabende und plattformübergreifende Softwarelösung präsentiert wird. Dieser Lösungsansatz soll es Personen ohne Programmierkenntnisse ermöglichen semi-automatisch plattformübergreifende, web-basierende und plattform-native mobile Applikationen zu erstellen.

Das Ziel dieser Arbeit ist es eine wiederverwendbare, flexible und modulare System-Architektur einer Mobile Application Publishing Platform (MAPP) vorzustellen. Im Zuge dessen, werden verschiedene Frameworks und web-basierende Plattformen zum Erstellen mobiler Applikationen analysiert. Um eine benutzerfreundliche Anwendung des Systems garantieren zu können, soll der Lösungsansatz auf bereits existierende Open-Source Content Management Systeme (CMS) aufbauen. Damit das System den Attributen Wiederverwendbarkeit, Zuverlässigkeit, Effizienz, Wartbarkeit und Erweiterbarkeit gerecht wird, kommen im Zuge des System-Designs bewährte Technologien und Entwurfsmuster zum Einsatz.

Um das Systems in der Praxis zu testen, wurde im Rahmen dieser Arbeit eine Benutzerevaluierung durchgeführt, wobei die Anwenderfreundlichkeit der MAPP und die Erwartungshaltungen der Testpersonen gegenüber den resultierenden mobilen Applikation untersucht und analysiert wurden.

Acknowledgments

I would like to express my gratitude to all those who gave me the possibility to complete this thesis.

At first I want to thank my supervisor Dr. Christian Gütl for giving me the opportunity to write this master thesis. My deepest appreciation goes to my co-supervisor Dr. Victor Manuel Garcia Barrios who assisted me with this thesis by words and deeds. I thank you for your professional advice, your patience, support, time and understanding.

I thank in particular my friends Luki, Stefan, Bernhard and Gunther for helping me staying sane through these difficult but instructiv years.

Further i would like to express my sincere gratitude to Anna, Berat, Brikena, Julia, Luki and my brother Ylli for carefully reading this thesis.

I want to thank Desiree, whos love, assistance and appreciation helped me to overcome frustrating moments.

Dhe në fund do të dëshiroja të falenderoja familjen time për mbështetjen e dhënë - prindërit e mi që më ndihmuan në shkollë dhe në universitet dhe besuan që mundem të kaloj edhe përmes kësaj. Ju gjithmonë qëndruat pranë meje, në kohë të mira dhe të këqija dhe për atë dua të ju falenderoj. Dua ta falenderoj edhe vëllaun tim Yllin për mbështatjen e tij dhe për këshillat gjithashtu dhe për pushimet dhe relaksimet nga fakulteti që e kemi kaluar së bashku.

Contents

Chapter 1: Introduction

The achievements in the field of mobile devices such as smartphones, netbooks and mobile phones have deserved great attention over the last years [Sanjay 2008]. At the same time, the development of the technical capabilities (displays, sensors, mobile networks) of these mobiles devices proceeds rapidly. In addition, the mobile market segment has been showing a high heterogeneity, regarding the technologies used for developing mobile applications for distinct mobile devices [Garcia-Barrios et al 2009]. Content providing companies face the problem on deciding how to deliver their contents on as many mobile devices as possible, focusing on the barrier of finding out which content fits which mobile technology best. Thus, there exists the need for low-cost solutions! Regarding the heterogeneity, mobile applications have to be developed for each operating system on the distinct mobile devices [DogHousemedia 2009]. Moreover, the development of applications for the distinct devices of the same vendor or manufacturer (e.g. tablets and smatphones) is distinct as well. This paper addresses the aforementioned problem moving towards a low-cost, easy-to-use and cross-platform software solution that enables non-programmers to design and develop mobile applications in a single-step process (i.e. "develop once, deploy multiple"). Therefore, in a first phase, we analyze the challenges on publishing and delivering content through a mobile Web application or mobile native application. Further, we propose a solution for a system architecture of a Mobile Application Publishing Platform (MAPP), which we call EMMA (Easy Made Mobile Applications) and enables a semi-automatically, cross-plattform and user-friendly deployment of mobile applications through Content Management Systems (CMS)[1].

Based on the statements given so far, and from a technical point of view, this thesis aims at bearing the challenges of coping with the heterogeneity in the mobile devices market area by compiling template-based cross-platform mobile applications using Domain Specific Language (DSL). Beside this solution

[1] CMS: http://php.opensourcecms.com/, Last visited: 17.08.2011

proposal, this thesis questions if there exist other solutions targeting at non-programmers, and providing an easy-to-use Web-based User Interface (UI) for the design and development of cross-platform mobile applications. In order to provide a reusable, reliable, efficient, maintainable, portable and extensible system, the solution approach investigates established technologies to keep the reusability and flexibility level as high as possible. Hence, in order to take advantage of established Web user experience, the proposed solution should build on an existing open source CMS and enhance it for the given goals.

Due to the maturation and convergence of technologies and the fact that mobile devices are equipped with large displays, Global Positioning System (GPS) sensors, and mobile networks, Location-Based Services (LBS) are getting very popular in the mobile market [Grossniklaus et al 2006]. Given this fact, this thesis narrows its application domain and focuses on the design and implementation of mobile applications by optimising the development process of non-programmers that want to create by themselves mobile applications within the domain of LBS. These mobile applications will help people (e.g. tourists) gathering information on specific locations.

For the purposes mentioned so far, a five step methodology has been chosen. Chapter 2 briefly analyzes the existing and most used mobile technologies, for the development of mobile native applications and mobile Web applications as well as shows that currently there is no Web-based solution that enables non-programmers to develop cross-platform mobile applications. Thereupon, chapter 2 deals with the subject of outlining the strengths and weaknesses of these mobile technologies as well as outlining the heterogeneity in the mobile market. Chapter 3 deals with the topic of using CMS for content delivery and outlines the challenges of choosing the right CMS, for the solution approach desribed in chapter 5. Chapter 4 presents the overall solution idea for a CMS based MAPP that efficiently supports non-programmers in the mobile application development process. Chapter 5 presents a reference design of a reuseable, flexible and modular MAPP architecture. Chapter 6 presents the evaluation of the developed prototype system, focusing on ease-of-use and briefly outlines the evaluation results Finally, chapter 7 gives a summary of the most relevant conclusions of this work as well as presents an insight into future work and open issues.

Chapter 2: Content Delivery Through Mobile Applications

The main target of this chapter is, from the user point of view, to identify the challenges of choosing whether to publish content through a mobile Web application or a mobile native application. Further, the chapter gives a comprehensible overview over the development and deployment possibilities for mobile applications that are currently available. Therefore, the first section of this chapter briefly analyses the existing and mostly used mobile technologies, for the development of mobile native applications and mobile Web applications. Section 2.2 provides a brief overview on the strengths and weaknesses of these technologies. Section 2.3 deals with the subject of developing mobile applications by using mobile application frameworks and Web based mobile application platforms. Since the application domain of the prototype implementation is Location-Based Services (described in chapter 6), the results of this chapter should help to better define the specific scope of this master thesis.

2.1 Understanding the Different Approaches

Mobile technologies and the market of mobile applications are a highly growing segments in the field of information technology and computer media. The mobile applications market is established on thousands of mobile applications, running on different mobile devices. Each device supports and performs certain tasks for the customer. Heterogeneity occurs based on the fact that different mobile devices run on different mobile operating systems. [DogHousemedia 2009]

In 2008, Internet access provided through mobile devices exceeded desktop computer-based access. Accordingly, the mobile application market is becoming a very interesting aspect for content providers and other content providers. The challenge now is to offer to these people an easy and cheap way to publish their content on mobile devices, covering as much mobile devices as possible. Regarding the heterogeneity, mobile applications have to be developed

for each mobile operating system. At the moment the three best selling smart-phone devices are Apple's iPhone / iPad, Android based mobile devices and Windows Phone 7 based devices. [DogHousemedia 2009]

The next barrier is finding out which content fits which mobile technology best; should content providers develop mobile native applications, mobile Web applications or hybrid applications? Therefore, the next part introduces the differences between the following mobile application technologies:

- Mobile Native Applications
- Mobile Web Applications & HTML 5
- Hybrid Applications

Mobile Native Applications

Mobile native applications are device specific applications, designed to run on a targeted mobile device with its appropriate mobile operating system. Currently, Apple's iOS, Google's Android and Microsoft's Windowsphone 7 are the most popular and most widely used mobile operating systems. Each of these mobile operating systems entails its own Software Development Kit (SDK), its own programming language and proprietary Application Programming Interfaces (API). These API's enable developers to use a mobile operating system, and its specific system resources like Touch Screen, WiFi, GSM Network, Camera and other resources (see figure 1 on the next page). Regarding the limited resources of a mobile device, this API guarantees flexible solutions with high performance. The wide range of mobile native applications is based on the diversity of the application-spedific functions basic telephony/messaging services, games, videos, access blogs and other rich media content. [MMA 2008]

Mobile Web Applications & HTML 5

Regarding the fragmentation in the mobile application market (discussed in section 2.2 in more details), content providers are looking for a solution of how to homogenize the heterogeneity of mobile devices, running different mobile operating systems, and design cross-platform mobile applications. The solution may be mobile Web applications. [Pimmel 2011]

> "In mature markets, the mobile Web, along with associated Web adaptation tools, will be a leading technology for B2C mobile applications through 2012, and should be part of every organization's B2C technology portfolio."[Gartner 2010]

Mobile Web applications are HTML 5[2], CSS[3] and JavaScript[4] based applications delivered over HTTP which use server-side or client-side processing. The code is executed by the browser on the mobile device and not like in mobile native applications by the operating system. Mobile Web applications provide a native like experience on a mobile device through the Web browser and can be used on all Web enabled mobile devices. [Conors and Sullivan 2010]

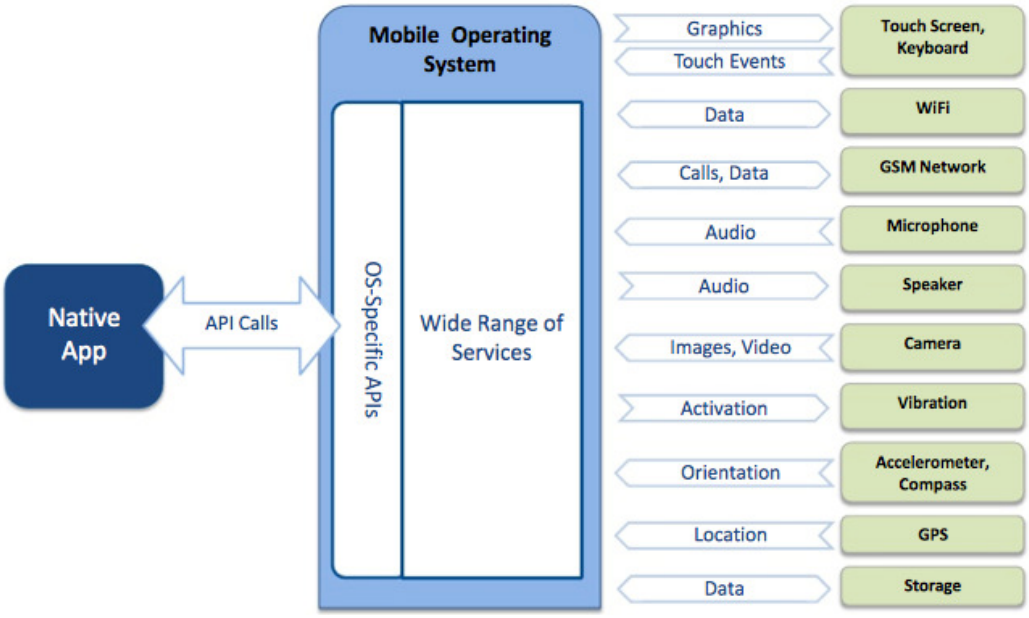

Figure 1: Architecture of Mobile Native Applications [Kaminitz 2011]

The technology making mobile Web applications so powerful is the next major revision of HTML - HTML 5. HTML 5 offers the possibility to develop features for mobile Web applications without the need of using proprietary API's. Below are listed the features in which mobile Web applications gained a quick advantage over HTML 5 [Marshall 2011]:

- Touch/Gestural Interfaces

[2] HTML 5 Overview: http://dev.w3.org/html5/spec/Overview.html, Last visited: 06.06.2011
[3] CSS overview: http://www.w3.org/Style/CSS/, Last visited: 06.06.2011
[4] JavaScript: http://www.javascript.com/, Last visit: 06.06.2011

- Video/Audio
- Graphics
- Access System Recourses

Touch/Gestural Interfaces: HTML 5 provides possibilities to improve the customer experience of a mobile Web application by simulating touch and swipe gestures, to which customers are used to in mobile native applications. Accordingly UI elements like scrolling lists, tab bars, grid views etc. (provided through CSS) achieve a native look & feel. [Marshall 2011]

Video/Audio: HTML 5 offers the possibility to embed audio and video files in a mobile Web application, apart from flash-based videos. [Marshall 2011]

Graphics: HTML5 makes use of device system resources to accelerate graphics rendering (although mobile native applications are much faster in terms of graphics-intensive operations; therefore games wont render as effectively in HTML 5). [Marshall 2011]

Access System Resources: HTML 5 provides access to system resources like camera (although not all mobile operating systems are supporting this at the moment), storage, accelerometer and Bluetooth (see figure 2). Using the HTML 5 geolocation API developers can access the mobile devices location over mobile Web applications. [Marshall 2011]

Hybrid Applications

Hybrid applications represent the combination of mobile native development and mobile Web technologies. Hybrid applications allow the development of applications on cross media platforms and accessing specific mobile device APIs. The native part of the hybrid application has access to all proprietary API's and serves as a bridge between the mobile Web browser and the mobile device API's. In this way the hybrid applications leverages all the features of a mobile device. The mobile Web application part of the hybrid application can either be on a server or local on the device as a bundle of HTML, CSS and JavaScript files (see figure 3). The Web portion of the app is responsible for rendering the UI elements on the mobile device screen. [Balaji 2011]

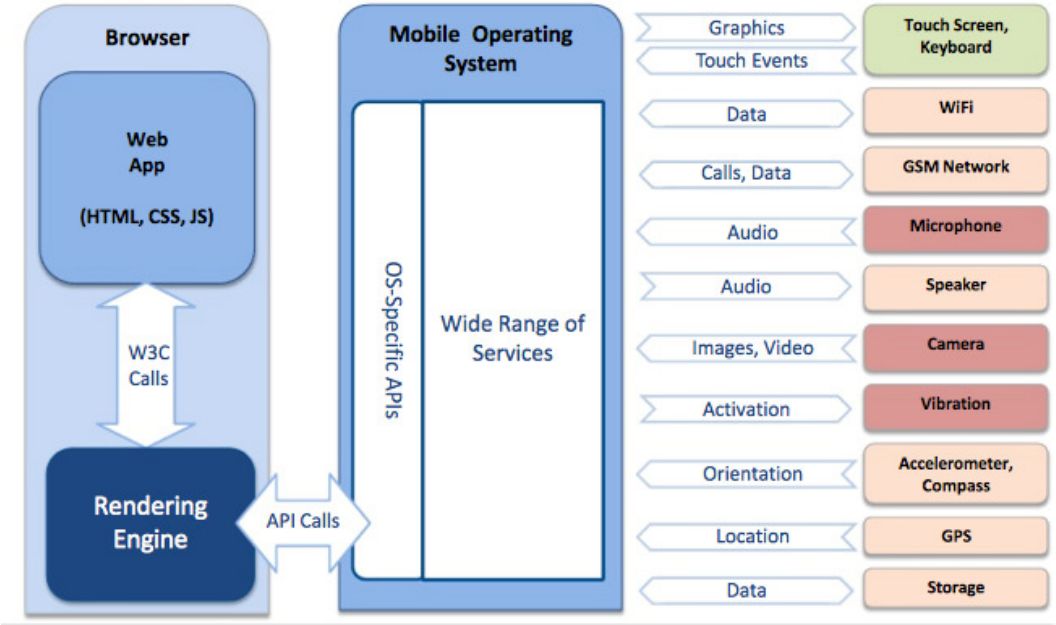

Figure 2: Architecture of Mobile Web Applications [Kaminitz 2011]

If developers choose to store HTML, CSS and JavaScript files on a server, they are able to update their content without having to submit an application update. If developers choose to store their files within the application bundle, the application will be able to run in the offline mode as well (more information about mobile Web application running in the offline mode are available in section 2.2) [Balaji 2011]. The disadvantage of hybrid applications is that developers need to know how to develop HTML 5 based mobile web applications, as well as how to develop the native part of the mobile application using native Objectiv-C or Java.

The next step is to outline the main differences between mobile native applications and mobile Web applications. The results will facilitate a response to the question of which mobile technology to choose, that targets the huge amount of available mobile devices.

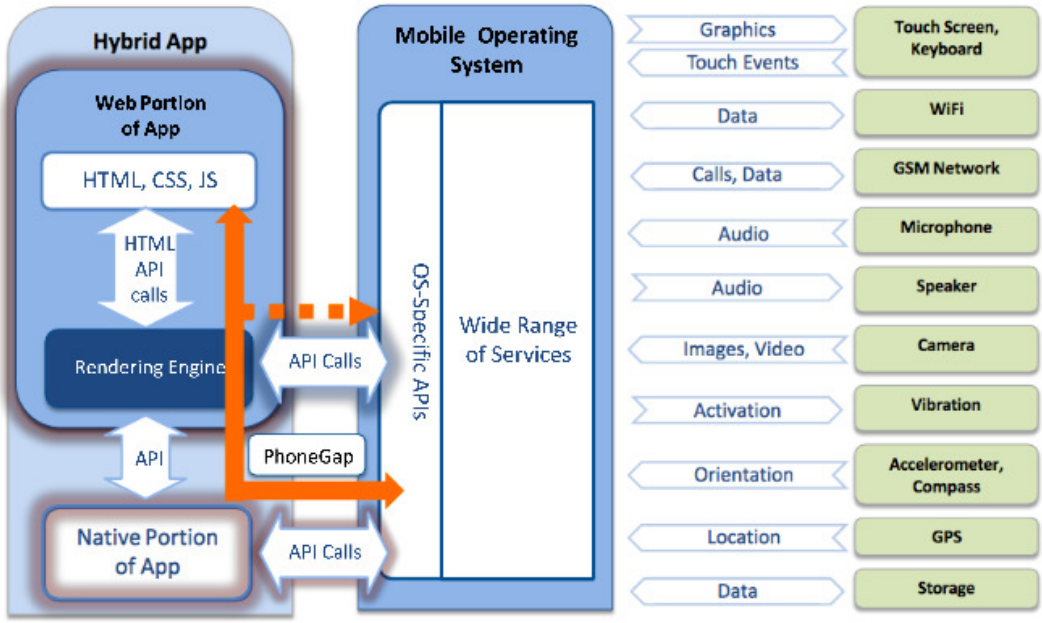

Figure 3:Architecture of Hybrid Applications [Kaminitz 2011]

2.2 Mobile Web Applications vs Mobile Native Applications

Section 2.1 introduced the main characteristics of mobile native applications and mobile Web applications. The next step is to outline the strengths and weaknesses of these two mobile technologies.

Around 400.000 mobile native applications are available on different app stores, but it is hard to know the exact number of active mobile Web applications [Caroll 2010]. Most people talk about mobile applications mean natively coded device-specific applications. Having in mind that mobile Web applications work in the offline mode, gives them the same presence as a mobile native applications [Caroll 2010]. Studies by the Global Intelligence Alliance[5] on mobile applications developed by different content providers showed that:

- 44 % of these content providers offer a mobile native application,
- 22 % of these content providers offer a mobile Web application,
- 34 % of these content providers offer both mobile native and mobile Web applications.

[5] Global Intelligence Alliance: http://www.globalintelligence.com/insights-analysis/white-papers/native-or-Web-application-how-best-to-deliver-cont, Last visited: 05.06.2011

- Many publishers saw higher user adaption's, usage volume and user engagement over mobile native applications.

- Mobile native applications deliver a higher click through rate (CTR) among the mobile advertising[6].[Caroll 2010]

Furthermore, they found out that there is content predestinated to publish as a mobile native application and other predestinated to publish as a mobile Web application.

> "Our study shows that games, social networking, lifestyle and entertainment, technology and gadgets, and travel and local category apps, tend to prefer the native approach. Web apps are dominated by news and weather publishers, who either only support a Web application or offer both." [Sæterås 2010]

From a development content provider's point of view, factors like development costs, maintenance, performance, usability and devices coverage are the most important factors deciding between either developing mobile Web applications or mobile native applications. The next step is to compare mobile native applications with mobile Web applications regarding the following points:

- Discoverability & Distribution
- Development & Maintenance
- Performance & Offline mode
- Fragmentation
- Usability

Discoverability & Distribution

Companies have different ways of publishing and distributing mobile native applications. Depending on the chosen mobile operating system, the publishing and distribution methods are different. Each mobile operating system is equipped with an online application market store. This application store offers the possibility to the customer to search and download mobile native applications, as well as developers to analyze, maintain and observe their

[6] Mobile advertising: http://www.admob.com/, Last visited: 07.06.2011

applications. The mobile applications usually are divided in different genres, rankings and categories hence represented in a more attractive way to the customers. Apple's iOS for example bounds their customer's completely to their App Store[7]. There is no other way for iPhone/iPad customers to install applications legally. This further means, if a content provider decides to develop mobile native applications for iOS, they are obliged to distribute their mobile native applications over the App Store (where Apple gets a 30% cut from all purchases). The same phenomenon appears with Microsoft's Windows Phone 7 and its App Hub[8] store. Concerning security reasons Microsoft's mobile operating system forces content providers to publish their mobile applications through App Hub. Google's Android OS otherwise offers content providers two ways of publishing and distributing their mobile applications. On the one hand the Android OS offers an application store – Android Market[9], and on the other hand the OS offers the possibility to make applications available via Web. This means that content providers are able to publish their mobile native applications on their own homepage where customers will be able to download them from. [Sæterås 2010]

Apart from Google's Android Market the other application stores have their own review guidelines. If content providers want to make their mobile applications available on one of the application stores, they first have to submit their application for a review. After the mobile application passes the review process (which can last for weeks) it will be ready for sale or it will be rejected. [Sæterås 2010]

The big advantage of distributing mobile Web applications is that they are searchable by search engines, which ensures that the mobile Web applications can be discovered by billions of consumers. Accordingly access to analytical services, traffic measurement tools, and ad server-targeting technologies is supported. Regarding the fact that content providers are able to publish their mobile Web application on their own homepages they do not need anyone's permission for distribution. [Sæterås 2010]

Development & Maintenance

As mentioned in section 2.1 mobile native applications use their own programming languages and proprietary device specific API. From a content provider's point of view this means that these programming languages has to be

[7] Mac App Store: http://www.apple.com/mac/app-store/, Last visit: 07.06.2011
[8] Windows App Hub: http://create.msdn.com/en-US/, Last visit: 07.06.2011
[9] Google Android Market: https://market.android.com/, Last visit: 07.06.2011

learned by their developers before they can start developing mobile native applications. Accordingly it takes more time developing mobile native applications in comparison with mobile Web applications, which in turn entails inescapeable costs. The next fact is that a content provider has to develop an own mobile native application for each mobile operating system. Thus, they have to build at least one for iOS, Android, and Windows Phone 7. Having 3 different application binaries means maintaining each one separately. [Sæterås 2010]

Mobile Web applications are cheaper and faster to develop based on the fact that they are HTML 5, CSS and JavaScript based applications. These programming languages are common to developers, thus it won't increase costs for further trainings. Skilled Web developers are far more readily available than those skilled in native programming languages such as the iPhone's Objective-C[10].[Caroll 2010]

Mobile Web applications run on common browsers and can be accessed via most Web-enabled mobile devices. Furthermore, hardware fragmentation decreases, which again leads to a cost advantage.[Caroll 2010] Regarding application maintenance mobile Web applications are very easy to handle, based on the fact that developers have to maintain one code base over all operating systems and devices. [Sæterås 2010]

Performance & Offline mode

Based on the fact that mobile native applications use device specific API's, this API's guarantee flexible solutions with high performances. With regards to the technological advancements, the distinction between mobile Web applications and mobile native applications in terms of speed and accessibility is decreasing. Mobile operating systems like iOS embarrass mobile Web applications from working as performant as mobile native applications. Since iOS 4.3 Apple arranged the iOS 4.3 Safari browser with its high-speed Nitro JavaScript engine, which is not used when Web apps are launched from the home screen. Accordingly, mobile Web applications run up to 2 times slower when they are launched from the home screen in „full screen" mode. [Rowehl 2010]

[10] Objectiv-C:
http://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html, Last visited: 07.06.2011

With regards to the fact that mobile native application bundles are saved on the devices storage (without memory limitations) and are executed by the mobile operating system, they are able to run completely without an Internet access (if the application logic does not need data from the internet). [Rowehl 2010]

Mobile Web applications had a cost disadvantage, particularly when roaming data across countries. This problem is eliminated with the offline mode support of mobile Web applications. Mobile Web application developers may get up to 50MB of database storage, in order to keep operating without a network connection. Using a manifest file, HTML5 lists all external resources used for running in offline mode. This manifest files mobile applications can be cashed when used offline, which in turn enables customers to use mobile Web applications while being abroad. [Rowehl 2010]

Running mobile Web applications on iOS 4.3 means that various Web caching systems, including HTML 5 Application Cashe, cannot be accessed through Web apps. According to this it is not possible to run mobile Web applications in the offline mode. [Metz 2011]

Fragmentation

Fragmentation is a serious problem in mobile native application development. Mobile operating systems like iOS are very straightforward to develop on. They provide a single API, which is used on all mobile iOS devices. Developers have to check for 3 different screen resolutions (iPhone 4, iPhone 3gs and iPad). All in all, the iPhone platform presents a clean development experience where developers write one piece of code, which runs on all iPhones and iPads. [Yared 2010]

Android in comparison has big problems with heterogeneous devices. There is a huge number of different devices with distinct display resolutions, branded Android operating systems and hardware. Thus, developers have to test their mobile native applications on as much devices as possible before publishing them on the Android Market. Google recognized that the fragmentation among Android devices is getting out of control and went about this problem with the Google Nexus One.

> "The Nexus One is the equivalent of the Java Reference Implementation or UNIX POSIX[11] and X/Open[12]: a baseline of

[11] POSIX: http://linux.about.com/cs/linux101/g/posixlparportab.htm, Last visited: 07.06.2011

what handset manufacturers would have to support in order to create a real Android handset. "[Yared 2010]

This further means that if a developer writes a mobile native application, which is supported, on the Nexus One, this application is supported on the pure Android operating system. If the same application does not run on a Motorola Milestone or HTC Legend, it is a problem induced by Motorola or HTC, not by the developer or Android. [Yared 2010]

Regarding different mobile browser characteristics (for example WebGL support), heterogenity through mobile browsers doesn't evade fragmentation problems. There fore mobile Web applications are not the solution for covering as much devices as possible.

Usability

Developing a UI for mobile native application, developers have a large set of different customizable elaborated UI elements supported by the corresponding SDK. [MacManus 2006]

Usability tests on mobile devices showed that costumers using mobile native applications had a much better user experience than customers using mobile Web applications.[MacManus 2006] Download delays and badly designed UIs are the main reasons why mobile Web applications did poorly. Since HTML 5 provides possibilities to improve the user experience of a mobile Web application by simulating touch and swipe gestures, users are used to in mobile native applications, attaching more importance on UI design could accomplish this bad come off [MacManus 2006]. Mobile application frameworks like Sencha touch, which will be discussed in section 2.4, will try to bring the native experience on the mobile Web application.

2.3 Mobile Application Platforms and Frameworks - Develop Once Deploy Multiple

Many software development companies have started offering different methods on how to build mobile applications. Since mobile Web applications are getting popular, mobile Web application frameworks entered the market as well. These frameworks provide Web developers, skilled with HTML, CSS and JavaScript knowledge, a way to build high performance mobile Web applications by using

[12] X/Open: http://groups.google.com/group/net.usenix/msg/638f079a0dad4802, Last visited: 07.06.2011

predefined stylesheets (all within this mobile Web application framework package).

On the other hand, development of Web based mobile application platforms has come through. Such mobile application platforms enable customers to automatically build mobile native applications targeted for special mobile operating systems [Balaji 2011]. Customers without special programming skills, low budget and curiosity are able to build up their own mobile native applications by using a Web based UI. This section will present the most popular and innovative mobile application platforms and frameworks, outlining their strengths and weaknesses.

An extensive Web-based survey within the context of this thesis work showed that there exists already a lot of mobile application frameworks and platforms. Most of them insist on paying a monthly fee in order to use their service. The following seem to be the most commonly used (or best-known) free mobile application platforms:

- AppMakr[13]
- MobiCart[14]
- Red Foundry[15]
- WidgetBox[16]

And mobile Web application frameworks:

- Appcelerator[17]
- MotherApp[18]
- PhoneGap[19]
- Sencha[20]

Based on the aforementioned frameworks and with regards to the utility, applicability and a multiplicity of the useful features, the following mobile

[13] AppMakr: http://www.appmakr.com/, Last visited: 07.06.2011
[14] MobiCart: http://www.mobi-cart.com/index.html, Last visited: 07.06.2011
[15] Red Foundry: http://redfoundry.com/, Last visited: 07.06.2011
[16] WidgetBox: http://www.widgetbox.com/, Last visited: 07.06.2011
[17] Appcelerator: http://www.appcelerator.com/, Last visited: 07.06.2011
[18] MotherApp: http://www.motherapp.com/index.php, Lastvisited: 07.06.2011
[19] PhoneGap: http://www.phonegap.com/, Last visited: 07.06.2011
[20] Sencha: http://www.sencha.com/, Last visited: 07.06.2011

platforms and mobile Web application frameworks have been chosen to be discussed and analyzed in more detail:

- AppMakr
- Red Foundry
- PhoneGap
- Sencha touch

AppMakr:

AppMakr is a free Web based mobile application platform offering customers the possibility to build up their own feed-based mobile Web applications. At the moment AppMakr support only iOS as mobile operating systems, but Android and Window Phone 7 will be supported soon. AppMakr offers the following services:

- Content based on text based feeds up to Flickr photos and Twitter activity.
- Customizable UIs.
- Unique application icon, splash screen, application header, tab bar icons.
- AppMakr provides also predefined icons.

If you want to publish your application on the App Store you need to have your own developer account, and AppMakr guides you through each step of the publishing process.[Dachis 2011]

Red Foundry

Red Foundry is a free Web based mobile native application platform. The main idea behind Red Foundry is to offer customers the possibility of developing their own mobile native applications without having the knowledge or the required skills for developing mobile native applications. Red Foundry provides a big set of different tools, which provide the possibility to build custom iOS applications. This platform has its own mark-up language, which enables customers to build new and custom layouts and functionality. At the moment Red Foundry works only for iOS, but Android and other mobile operating systems will come soon.

One big difference between Red Foundry and other mobile application platforms is that applications build with Red Foundry look and feel unique, whereas mobile applications on the other platforms always look similar to each other. [Evans 2011]

Red Foundry offers you the following services:

- Connect to social networking content (Facebook, Twitter, and Wordpress, and private or custom APIs)
- Free simulator app for testing
- Intelligent analytics
- Push notifications
- User-generated content
- Social networking and media capabilities
- Online learning tools for first time developers

PhoneGap

PhoneGap is a mobile Web application framework based on open Web standards. This framework enables developers to build mobile Web applications based on standard Web technologies like HTML5 and JavaScript, as well as utilize native device APIs. The PhoneGap Framework has been downloaded over 350000 times and thousands of apps have been built using PhoneGap. Regarding to the fact that PhoneGap apps are standards compliant, they are proofed to work with browsers as they evolve. Developers and content providers can use PhoneGap for mobile applications that are free, commercial or open source. [Dohrmann 2010]

Sencha-touch

Sencha Touch is a very powerful cross-platform framework aimed at touch enabled devices. Using standard common technologies like HTML5, CSS3 and JavaScript for the highest level of power, flexibility, and optimization, developers are able to build powerful mobile Web applications. Sencha-touch gives you the possibility to access system resources on the mobile devices and to save data for offline usage by using a local storage proxy. The SenchaTouch

framework uses CSS3 in stylesheets to provide the most robust styling layer possible. The most powerful feature of the Sencha touch framework, are the gesture recognition patterns. Sencha touch is able to recognize touchstart and touchend, tap, double tap, swipe, tap and hold, pinch, and rotate. Sencha Touch offers to developers the possibility to use a variety of sources to fill the UI elements with data (AJAX[21], JSON[22], or YQL[23]).

2.4 Summary

This chapter gave an overview on some currently relevant topics in the mobile device market and reserach area. Based on the statements presented in this section, it can be stated that nowadays it is not easy and trivial to decide which technology to use when developing a mobile application. Mobile native applications offer API's to access operating systems specific system resources like Touch Screen, WiFi, GSM Network, Camera. This in turn guarantees flexible solutions with high performance – but are dependent on a specific mobile operating system. HTML 5 based mobile Web applications provide a native like experience on a mobile device through the Web browser and can be used on all Web enabled mobile devices. The disadvantage is that mobile Web applications run up to 2 times slower then mobile native applications. Furthermore they may have a cost disadvantage, regarding the use of roaming data across countries. Furthermore, download delays and badly designed UIs are the main reasons why mobile Web applications did poorly.
In addition, this chapter discussed Hybrid applications which allow developing applications on cross media platforms and accessing the specific mobile device APIs. In this way the hybrid applications leverages all the features of a mobile device. The disadvantage of hybrid applications is that developers need to know how to develop HTML 5 based mobile Web applications, as well as how to develop the native part of the mobile application using e.g. native Objective-C, Java or C#.

As mentioned in section 2.2, comparing mobile native applications and mobile Web applications with respect to the distribution, maintenance, performance, usability and devices coverage, each mobile technology has its own strengths and weaknesses. Mobile operating systems are equipped with online application market stores. Mobile Web applications in turn are searchable by crawlers and can be discovered by billions of consumers. Mobile native applications use their own programming languages and proprietary

[21] AJAX: https://developer.mozilla.org/en/AJAX, Last visited: 08.06.2011
[22] JSON: http://www.json.org/, Last visited: 08.06.2011
[23] JQL: http://developer.yahoo.com/yql/, Last visited: 08.06.2011

device specific API which has to be trained to developers. Mobile Web applications are based on common programming languages like HTML5, CSS and JavaScript based applications. Based on the fact that mobile native applications use device specific API's, this API's guarantee flexible solutions with high performances. Mobile native applications written for operating systems like iOS are very straightforward to develop, and supported on all mobile iOS devices. Android in comparison has big problems with heterogeneity due to the huge number of different devices with specific display resolutions, branded Android OS and hardware differences. Mobile Web applications in turn can be used on all Web enabled mobile devices.

Since mobile applications are becoming popular all content providers want to publish their content on the mobile channel. Thus, Section 2.3 focused on mobile application platforms and mobile Web application frameworks offer to content providers a very easy and effective way to build their own individual mobile applications. Mobile Web application frameworks provide to Web developers, skilled with HTML, CSS and JavaScript knowledge, a way to build high performance mobile Web applications.

Web based mobile application platforms enable customers to automatically build mobile native applications targeted for special mobile operating systems. Customers without special programming skills, low budget and curiosity are able to build up their own mobile native applications by using a Web based UI.

In short content providers are willing to publish their content over mobile native and mobile Web applications in the most efficient and reliable way. Therefore the target of this thesis is to implement a solution approach for a Mobile Application Publishing Platform (MAPP), whereat the application domain focuses on the field of LBS (see chapter 6).

Chapter 3: Content Management Systems

According to the findings from the previous chapter, several challenges can be identified while choosing whether to publish content through either a mobile Web application or mobile native application. Since hybrid applications require advanced know-how regarding different programming languages, this technologie is disregarded within the scope of this thesis. In order to better understand and restrict the application´s scope and application domain of the LBS-oriented prototype solution described in chapter 6, this chapter deals with the topic of using CMS for content delivery.

Section 3.1 pictures a brief overview of the technologies used in and for CMS, i.e. the Access Control List (ACL)[24] and permissioning systems, as well as short analysis of the CMS source code. Section 3.2 presents the given module-oriented approaches to expand these CMS. Section 3.3 deals with the subject of Mobile Content Management Systems (MCMS)[25]. The target of this chapter is to render in a comprehensible manner the challenges of choosing the right open source CMS for the solution approach described in chapter 5.

3.1 Moving Towards a CMS Development Basis

The main idea behind CMS is to provide a simple way of publishing, maintaining and delivering content on homepages in the most efficient and reliable way. This chapter will analyze different aspects of the chosen open source CMS. An extensive survey about the mainly used open source CMS would go far beyond the scope of the context of this thesis work. Regarding my personal expertise the CMS discussed within the range of this thesis are:

- Drupal[26]

[24] ACL: http://www.freebsd.org/doc/en/books/handbook/fs-acl.html, Last visited: 17.08.2011
[25] MCMS: http://www.insight-corp.com/%5CExecSummaries%5Ccontent08ExecSum.pdf, Last visited: 17.08.2011
[26] Drupal: http://drupal.org/, Last visited: 17.08.2011

- Joomla[27]
- Silverstripe[28]

The next purpose of this section is to outline these systems´ main technological aspects they are based on (cross-browser compatibility, technologies in use, server support, source code, design patterns and content management). As one main aim of this work is to easily integrate a cross-platform deployment module for mobile applications on an existing system, the analysis of the chosen CMS will be arranged in the following way:

- Use of common Web standards and Web technologies
- Source code expandability and design patterns
- Content implementation and navigation
- ACL and user permissioning

3.1.1 Drupal

Drupal is based on open and common Web technologies like HTML1[29]/XHTML2[30], CSS3[31], MySQL[32] (PostgresSQL[33] since Drupal 6) and PHP4[34]. This CMS is cross browser compatible and provides the customers with full support on Firefox, Safari, Internet Explorer, Google Chrome, and Opera. Drupal is written in object oriented PHP. Server-side requirements are Apache, MySQL and PHP (the needed PHP version has to be higher 4.3.3). [Graf 2006]

Drupal delivers a brilliant source code which strengthens coding conventions and coding standards. All versions of the Drupal source code follow the current coding guidelines. Accordingly developers get very fast used to this standard and are able to understand foreign code sections easily. Drupal engages developers in using common coding techniques and PHP tricks, making their code more readable and maintainable. Thus developers are able to expand the CMS easily. Drupal´s source code is highly elaborated and the system´s architecture is well designed.[Graf 2006]

[27] Joomla: http://www.joomla.org/, Last visited: 17.08.2011
[28] SilverStripe: http://www.silverstripe.com/, Last visited: 17.08.2011
[29] XHTML1: http://www.w3.org/TR/xhtml1/, Last visited: 17.08.2011
[30] XHTML2: http://www.w3.org/TR/xhtml2/, Last visited: 17.08.2011
[31] CSS3: http://www.w3.org/TR/CSS/, Last visited: 17.08.2011
[32] MySQL: http://www.mysql.com/, Last visited: 17.08.2011
[33] PostgresSQL: http://www.postgresql.org/, Last visited: 17.08.2011
[34] PHP4: http://openbook.galileocomputing.de/php4/kapk.htm, Last visited: 17.08.2011

The dominant design patterns in Drupals systems architecture are the Observer´s, Visitor`s, Factory and the Command patterns. The Observer pattern is built on a super class (called subject) and multiple observer classes observed by the subject. If an event occurs in the system, the subject notifies the registered observer classes, which are able to react on this event. This mechanism makes the system´s architecture very flexible regarding the extension of the system without changing existing code. This guarantees that developers can easily add new observer classes listening to certain events and triggering the regarding actions. [Graf 2006]

The Visitor pattern in turn is quite similar to the Observer pattern besides the fact that it is used to extend the behavior of the subject rather than simply respond to events. [Graf 2006]

The Factory pattern instead consists of the factory objects and client objects, whereas the client objects ask the factory object for a piece of code with a special functionality. The factory object is able to handle this request and hand out the right implementation. When for example a client asks a factory for a database connection, but doesn't tell the factory either if he needs a MySQl or SQLite database, It is up to the factory to decide which kind of connection to hand out to the client. This pattern enables expanding the CMS and build richer and more powerful clients. [Graf 2006] [Garfield 2010]

The Command pattern itself consists of an abstract class, which is responsible for doing specific actions. Instances of this class can define how this action will look like and how specific data will be treated and executed. Concerning flexibility this pattern facilitates to build up a lot of different commands and encapsulate different behavior. [Garfield 2010]

Dependency Injection is the next very powerful design pattern in Drupals system architecture.

> „A given algorithm should never have to request the resources it needs but should be given its resources. That is, its dependencies should be "injected" into it rather than it having to go out and get them“[Garfield 2010]

The Drupal core system provides a user permissioning management system out of the box. Hence registered users get equipped with usernames and specific permissions within the CMS backend. User can be registered as content author as well as editors up to super admins (who are able to access all areas in the CMS backend). Furthermore Drupal offers an own ACL module. This module enables other modules to access user management and permissioning over the

appropriate API. In the following part of the paper on hand some user permissioning modules are listed and analyzed [Graf 2006]:

- **Content Access:** The Content Access module allows to define fine-grained permissions for content types by role and author. In this case users have the permission to view, edit and delete content - content driven persmissioning [Jeavons and Knaddison 2010]

- **Flexi Access:** The Flexi Access module is another user management module using Drupals ACL API. This module offers a well-designed UI for setting user permissions and allows users to access, update and delete content. [Jeavons and Knaddison 2010]

- **Image Gallery Access:** As long as Drupal offers an own image gallery management page, this module manages the permissions for this area. [Jeavons and Knaddison 2010]

3.1.2 Joomla

Concerning Web technologies Joomla is similar to Silverstripe (see section 3.1.3) and Drupal (see section 3.1.1). The Joomla CMS is based on open Web standards like PHP, MySQL, JavaScript, and HTML. Joomla is cross browser compatible. Accordingly customers are able to use Joomla on Firefox, Safari, Internet Explorer, Google Chrome, and Opera. Furthermore Joomla has some requirements on Web servers. Web servers should be able to handle many processes at once, in order to guarantee a better performance. Moreover Joomla needs a fully operational Web server like Apache[35] or Windows IIS[36] to run on. Hence the Web server should support MySQL and PHP (specific modules within PHP for MySQL, XML, and Zlib[37] functionality) amongst others. [A3Webtech 2011]

Generally speaking Jommlas system architecture is very elaborated, expandable and straight forward, which makes this CMS superior to others. Regarding expandability Jommla architecture is based on three design patterns. The Observer pattern, the Composite pattern and the Strategy pattern. All these patterns are surrounded by a Model View Controller (MVC) pattern. This MVC pattern divides the system in:

- Models - Jmodel (PHP files executed on the server)

[35] Apache: http://www.apache.org/, Last visited:17.08.2011
[36] Windows IIS: http://www.iis.net/, Last visited: 17.08.2011
[37] ZLIB: http://zlib.net/, Last visited: 17.08.2011

- Controller - JController (executed on the server; extentions such as search etc.)

- View - JView (templates for content representation).

Each of these parts of the system can be developed and expanded separately without affecting the other parts of the system. This guarantees a high level of flexibility.

The Joomla CMS assumes every content to be integrated in a section and a category. Joomla differentiates between categories, sections and articles.

> „Articles are the basic form of content. Articles are organized into categories, and categories are organized into sections. This terminology comes from traditional print models, where a newspaper might have News, Sports, Business, and Features sections, and those sections might have different categories, such as local and national news in the News section, baseball and basketball in the Sports section, the stock market and retail business in the Business section, and home furnishings and cooking in the Features section“[Marriott and warning 2011]

Joomla´s frontpage manager is responsible for the article organisation, order and offers users also the possibility to change the order of existing articles intuitively by drag and drop. [Marriott and warning 2011]

Joomla´s content area offers a WYSIWYG editor, which generates HTML for the front-end. Joomla offers its customers also a media manager, which is a tool to organize images. The difference to Silverstripe´s module (see section 3.1.3) is, that this tool itself is not able to resize images. If users add big images on their website, the performance results to be poor. Accordingly, users have to resize images on their own, before uploading them on the CMS. [A3Webtech 2011]

Like Drupal and Silverstripe also Jommla offers a basic ACL system from the scratch. Joomla offers the possibility to attach permissions to categories and users. The existing user permissions are listed below [A3Webtech 2011]:

- **Registered:** Users can only read content but neither edit or publish content – Which means that they have no privileges.

- **Author:** Users can add new content to the CMS backend.

- **Editor:** Editors have the same right as authors but can also edit existing content of different users.

- **Publisher:** The role of publishers is to decide whether content is ready to be published or not.

3.1.3 Silverstripe

The core system of the Silverstripe CMS is based on common open source technologies and open Web standards, like PHP, MySQL, HTML, CSS and Javascript. The deliberate adoption of these technologies makes this CMS futureproof. Regardingly, users are able to use and access the Silverstripe backend using Firefox, Safari, Internet Explorer, Google Chrome, and Opera. These are the most widespread Internet browsers nowadays. Furthermore the Silverstripe CMS is supported on Windows, Mac and Linux based operating systems. Putting all these facts together, this illustrates Silverstripe`s cross-browser compatibility now and in the future. [Borschart and Schommer 2009]

Silverstripe´s backend is based on Javascript and PHP, which makes the debugging simpler, and allows faster and easier customizations. Silverstripe fits quite well in all IT environments according to the fact that it needs only PHP to run on the serverside. Silverstripe can be installed on all popular Windows, Mac or Linux Web servers like, Apache and Microsoft IIS. [Borschart and Schommer 2009]

Silverstripe differentiates between two types of users: Non-technical users and PHP programmers. Thus the Silverstripe system is divided into two parts. The first part is the Silverstripe CMS, which is intended for the non-technical user. Since these kinds of users are content authors in most cases, anything technical is strictly kept out of this part of the system, to guarantee a better usability. The Silverstripe CMS backend allows non-technical users to update Web pages, images, links and moderate blog comments easily. [Borschart and Schommer 2009]

The second part of the Silverstripe CMS is the Sapphire framework. This framework is intended for developers, to expand and customize their websites by writing HTML, CSS, JavaScript, and object-oriented PHP code. The source code of this system is documented very well, and is very easy to customize and expand. According to the fact that software like Silverstripe should be as flexible, developer-friendly and easy to expand as possible, the next step is to analyze the design patterns used in Silverstripes source code. [Borschart and Schommer 2009]

Silverstripe offers different content representation pages, which the user has to choose when adding new content to the CMS. Each content representation page belongs to a specific page type. These page types are

presented with a MVC design pattern. A content representation page is put together by a model class, a controller class, as well as a view class. The Sapphire Framework pretends a specific folder structure to the developers, which also represent the MVC structure, as well as the folder naming conventions. [Borschart and Schommer 2009]

The next very important design pattern used in the Sapphire Framework is Object Relational Mapping (ORM). This design pattern bears the responsibility to enable dataflow and mapping between the Silverstripe PHP files and MySQL databases. In detail the design pattern used to implement this content to database mapping is called Active Record design pattern. The main idea of the ActiveRecord design patterns is to represent a database table at code level by using appropriate data objects. Each object maps a row in a specific database table (If a new data object is allocated, the database table creates a new row; if a data object is updated the appropriate content in the database table is updated automatically). Therefore, this design pattern enables developers to communicate with the database in a very simple and flexible way without the need to fire up complex queries. [Borschart and Schommer 2009]

> „Specific concepts in a relational database have to be matched up to types that can be represented in-memory by PHP. There are three simple conventions for this: A database table maps to a class. A database row maps to a class instance. A database column maps to an instance attribute.“ [Borschart and Schommer 2009]

Comparing Silverstripe to CMS like Drupal and Joomla, it becomes quite clear that the Silverstripe CMS with the Sapphire framework have the clearest, simplest, cleanest and most intuitive UI and backend structure.

Silverstripes UI is divided into three main areas:

- The main navigation in the top-most bar.
- The tree view of pages on the left side, which gives them an overview of the created website pages and the hierarchy.
- The content- editing area on the right side, which also contains a What You See Is What You Get (WYSIWYG) editor (translate the content to HTML) and buttons along the bottom to perform common tasks like saving and publishing pages.

Silverstripe enhances the user´s experience by using common Web technologies

like AJAX[38]. These technologies facilitate the upgrading of the backend´s experience and intuitiveness. Users are able to add items, saving pages, inserting images and links, by using drag&drop. The content editing area offers the users the possibility to work on a content page in an own draft view until it is finished and ready to go online. Silverstripe offers a version repository for managing, merging, and recovering older versions of a content page. [Borschart and Schommer 2009]

Silverstripe uses a flexible user permissioning management system to support content authors in their collaboration activities. In this way different authors are able to work and manage the same content. Therefore Silverstripe offers an own user management interface (available in the Silverstripe security area). This area offers the possibility to create groups and assign permissions to this groups – which can be seen as group based access control. Silverstripe offers four different permissions:

- CMMAIN: Users are able to edit page and site content.
- Comment admin: Answer comments in the Silverstripe framework (forums).
- Model admin: Gives Access to the comment management system of Silverstripe.
- Custom permission: Developers are able to develop own user or group permissions.

Once a user joins a group this user will also adopt all other user permissions. Accordingly each user receives adequate permissions depending on the group he belongs to. [Borschart and Schommer 2009]

3.2 Module-based Expandability and Customization of Open Source CMS

The previous section aimed to outline the strengths and weaknesses of the chosen CMS regarding cross-browser compatibility, development and ACL. The next step is to analyze to which extent these CMS are easily expandable by taking advantage of existing modules to fulfill commonly required website features.

[38] AJAX: http://api.jquery.com/category/ajax/, Last visited: 17.08.2011

The advantage of open source CMS is that experienced developers are able to customize the entire system´s architecture as well as expanding the source code without limitations. According to this, developers have implemented new features and modules and have placed these modules to everyone´s disposal. With thousands of different modules available, even the worst open source CMS can be turned into a fully functional super software. Accordingly open source CMS are not worse than expensive ones. [Webhostingfan 2010]

> „You don't need to pay, because all of the features you're "paying for," are available for free through Web applications that can be added to your free CMS in a matter of minutes." [Webhostingfan 2010]

To that fact the next step is to analyze the most useful modules of the given CMS.

3.2.1 Drupal

Drupal offers a module-based approach of expanding the core systems feature-set. The core system is very simple but powerful although the main idea of giving the CMS more functionality works by installing modules. Since Drupal 6 developers are able to choose only specific modules to run, which keeps the systems performance very high. Beyond modules Drupal offers a possibility to develop own themes easily. [Graf 2006]

All modules use Drupals core API concerning security and consistency reasons. Accordingly bugfixes and performance optimization on the systems core API will influence the performance of all modules positively. Drupal offers more than 200 different modules as well as a lot of different themes and over 25 supported languages. Subsequently four of the mostly used modules are listed [Graf 2006]:

- TinyMCE WYSIWYG-Editor
- Image-Module
- Drupal Views
- Drupal Logingtoboggan

TinyMCE WYSIWYG-Editor

The TinyMCS1 module represents a WYSIWYG editor written in Javascript, which is able to transform simple text in HTML code. This feature is perfect for

customers without HTML skills. Regarding user permissions, this editor is able to show special content manipulation tools to a specific user group. [Graf 2006]

Image-Module

The Image module offers the possibility to upload images easily. It's the base module for all other image manipulation modules. Drupal isn't able to process big picture on it´s own like Silverstripe is. Big pictures will lead to performance and usability problems. [Graf 2006]

Drupal Views

Drupals Views module offers the possibility to control how lists of content are presented. [Graf 2006]

Drupal Login Toboggan

The Login Toboggan module allows customers to extend the registration forms and the procedure how users can register on a homepage. Drupal offers a logging mechanism out of the box. The Login Toboggan enables users to log in either with their username or their mail address. Furthermore users are able to set their own passwords. [Graf 2006]

3.2.2 Joomla

Joomla offers over 5000 modules, plugins and templates, where most of these modules are for free. There are different groups of modules: Content, Display, Utility, User, and Navigation; whereas each module has a unique set of configuration parameters. Joomla offers modules for mailing lists, calendars, photo albums and a lot more. It´s beyond the scope of this chapter to cover all modules that are available for the Joomla CMS. This section is intended to give a list of popular extensions that are helpful to a large amount of sites and popular in the Joomla! Community [A3Webtech 2011]:

- Joomla! Content Editor (JCE)
- RokBox
- JoomFish

- Jforms

Joomla! Content Editor (JCE)

The JCE modules provide a simple WYSIWYG Editor, which simplifies content editing. JCE gives customers the possibility for multimedia management. This module enables customers to upload images, mediafiles, link handling, and supporting plugins. The editor is able to differentiate between customers based on their permission in the CMS. [A3Webtech 2011]

RokBoxx:

RokBox is an all-in-one media plugin, which allows customers to easily insert media such as images, video, audio, files and even other Web sites into content items. [A3Webtech 2011]

JoomFish

The Joomfish module allows customer to publish content in different languages, although it is not able to translate content. [A3Webtech 2011]

JForms

JForms provides customers the possibility to build custom contact forms for collecting user specific data and embeds these forms in the appropriate content item. [A3Webtech 2011]

3.2.3 Silverstripe

Silverstripe is handed out with a powerful set of features. In addition customers have the possibility to expand the system easily by attaching different modules. Subsequently the mostly used modules are listed [Borschart and Schommer 2009]:

- Gallery Module
- Userforms module
- Advanced Workflow

Gallery Module

The Gallery module helps customers to display images on their websites. This module enables the customer use various JavaScript and CSS effects like enlarged popping up images using a pleasing visual effects. One of the Silverstripe´s benefits is that it offers a powerful and automatic image manipulation tool. This tool facilitates customers to upload 10 megapixel images onto their Websites, which automatically will be reduced to an appropriate size for the Websites. [Borschart and Schommer 2009]

Userform Module

The Userform Module enables customers to create individual forms on their homepages without having HTML skills. The modules offer many standard form fields including dropdowns, email fields, checkbox, and radio buttons. Furthermore customers are able to organize their forms by drag and drop the interface. [Borschart and Schommer 2009]

Advanced Workflow

The Advanced Workflow module adds configurable workflow support to the CMS. Customers are able to define new workflow definitions via an adequate UI. These workflow definitions can then be applied to any item in the main CMS. One of the killer features of this module is the ability to create publishing workflows with email notifications and multiple user group assignments. [Borschart and Schommer 2009]

The main idea behind the module-based approach of expanding the Silverstripe CMS is to prevent developers to reinvent and rewrite existing features, which is very time consuming. There are a lot of useful modules supported out of the box like a blog, forum, shopping cart, or any other relatively standard addition to a website. [Borschart and Schommer 2009]

3.3 Mobile Content Management System (MCMS)

The previous section outlined that Drupal, Joomla and Silverstripe provide a lot of modules for expanding the CMS functionality without developing own modules. The next step hence is to analyze the mobile Web and multi channel delivery capabilities of these CMS.

> „Most CMS products boast of multi channel delivery capabilities. What this essentially means is that you can manage content in the CMS repository and deliver that content to a Web site, a mobile handset, a PDA or whatever. However, since one of the important principles of Content Management is separation of content from presentation, multi-channel delivery in this form is straight forward – your content is stored in raw format and you apply appropriate presentation over it to deliver content to whatever device you want"[Durga 2007]

This chapter analyzes to which extent the chosen CMS systems support content delivery over mobile devices.

Drupal Mobile

Drupal's Mobile Tools provide customers the possibility to optimize their websites for mobile devices and create a custom user experience for their mobile visitors. [Kathryn 2010]

The mobile tools offer user agent detection, device detection and support for third party modules like Browscaps[39]and Wurfl[40]. These modules are used for optimized theme switching mechanisms depending on the mobile device (iPhone, android, black berry). Visitors of the mobile website can choose either if they want to open the desktop version of the website or the mobile version of the website on their mobile device. Further Drupal mobile supports role-based content representation. [Kathryn 2010]

Joomla Mobile

Regarding the fact that mobile-accessible website are getting more and more important Joomla offers a technology to its customers to automatically create mobile website, and render their website on mobile devices in a optimized way. This technology is called PDA-mambot.

A mambot is a task-oriented function, which manipulates the content of a website before this one gets displayed on the browser. The main function of the mambot is to ensure a mobile friendly representation of the content. The PDA mambot detects whether the visitor is entering the website by a desktop

[39] Browscaps: http://drupal.org/project/browscap, Last visited: 17.08.2011
[40] Wurfl: http://wurfl.sourceforge.net/, Last visited: 17.08.2011

computer or a mobile devices. Dependently the PDA mambot activates the appropriate rendering engine, either using a 'multi-client' approach or 'multi-site' approach. In the case of the 'multi-client' approach, the mambot determines from which device the site is being called. If the website is entered by a mobile device the multi-site approach offers the possibility to the customer to remove special HTML tags form the mobile version of the website for space saving reasons. [Dryabov 2008]

Silverstripe Mobile

Silverstripe provides to customers the Silverstripe mobile module for rendering websites on mobile devices and optimizing their websites for mobile devices. This module provides mobile optimized themes to display a website on mobile device browsers. Customers can either use the predefined themes included in the Silverstripe mobile module or develop own themes using appropriate HTML and CSS. This module enables customers to customize not only the look&feel but also the user experience on mobile devices. However, there is a considerable effort in achieving cross-browser compatibility across mobile devices. [Harvey 2010]

The module provides a simple UI for the module configuration process as well as automatic device detection for redirecting mobile devices to the mobile version or tablet version of the a website. [Harvey 2010]

3.4 Summary

Drupal, Joomla, as well as SilverStripe are based on open and common Web technologies like HTML, CSS, PHP and MySQL. Therefore, all three CMS are cross-browser compatible. SilverStripe is supported on popular Windows, Mac or Linux based Web servers. SilverStripe seems to be the leading CMS regarding serverside requirements.

Each of these three CMS have well-designed and elaborated system architectures and a clean and well-documented source code. Further, the developers of each community use coding conventions and standards as well as follow well-known design patterns, which makes their source code more readable and maintainable.

Silverstripe takes a step forward by dividing the entire system in the Silverstripe CMS and the Sapphire Framework. This system division makes the CMS very friendly for both, customers and developers.

Drupal and Joomla offers adequate ACL modules for user permissioning. Whereas Silverstripe uses a flexible user permissioning management system out of the box to support content authors in collaboration. In this case different authors are able to work on same content. Thus, Silverstripe offers an particular user management interface.

One of the main advantages of the analysed open source CMS is that experienced developers are able to customize the whole systems architecture. Drupal, Joomla and Silverstripe offer a module-based approach of expanding the core systems feature-set. Drupal´s core system is very simple but powerful although the main idea of giving the CMS more functionality is by installing modules.

Joomla offers over 5000 modules, plugins and templates, where most of these modules are for free. Silverstripe is handed out with a quite powerful set of features. In comparison to Drupal and Joomla, the Silverstripe CMS doesn't offer as many modules to its customers as the other CMS. These few modules in turn provide perfectly working core features matching with the core systems API.

This chapter outlined also the main techniques of rendering mobile Web pages out of a CMS. Drupal offers user agent detection, device detection, support for third party modules as well as optimized theme-switching mechanisms. Joomla instead, offers a PDA mambot which detects whether the visitor is entering the website by a desktop computer or a mobile device and activates the appropriate rendering engine, either using a 'multi-client' approach or 'multi-site' approach. Silverstripe in contrast enables developers to create own themes using appropriate HTML and CSS as well as providing a simple UI for the module configuration process. The SilverStripe mobile module provides mobile optimized themes to display a website on mobile device browsers.

Summing up it can be stated that all three CMS have comparable features. Drupal and Joomla offer their gestures by offering the customer the possibility to expand the CMS with different modules. SilverStripe instead is the only CMS, which offers all the necessary features out of the box. Furthermore, SilverStripe is a very customer- and developer-friendly CMS with an elaborated system architecture and very simple and clean content editor. Accordingly, SilverStripe is predestined to be used as CMS-backend for the LBS prototype solution described in chapter 6. Regarding mobile technologies, it can be stated that all three CMS are able to render mobile Web applications for multi channel delivery over mobile devices. However, neither Silverstripe nor Joomla nor Drupal are able to build mobile native applications. Therefore, the main target of the next chapters is to implement a cross-platform deployment module for mobile Web and mobile native applications on Silerstripe.

Chapter 4: Content Management Systems as Mobile Applications Platforms

Since one aim of this work is to easily integrate a cross-platform deployment module for mobile applications on an existing Content Management System (CMS), the previous chapters were destined to introduce different mobile technologies as well as choose SilverStripe as the underlying CMS. In order to better define and restrict the application scope of the solution approach described in chapter 5, this chapter presents an idea of a CMS based Mobile Application Publishing Platform (MAPP), considering Location Based Serices LBS as the focused application domain of the prototype implementation presented in chapter 6. In accordance this section presents the requirements of capital importance towards such a MAPP in form of story cards.

4.1 Semi-Automatically Generate Mobile Applications

Since the demand for mobile applications is rising, content providers like Red Foundry[41], AppMakr[42], Phone Gap[43] and APPlause[44] are developing mobile application platforms and mobile application frameworks enabling customers to build own mobile applications in a fast and reliable way. Mobile application frameworks provide Web developers, skilled with HTML, CSS and JavaScript knowledge, a way to build high performance mobile Web applications by using predefined stylesheets. Mobile application platforms in turn enable customers to automatically build mobile native applications targeted for special mobile operating systems (see chapter 2).

[41] Red Foundry: http://www.redfoundry.com/, Last visited: 20.10.2011
[42] AppMakr: http://www.appmakr.com/, Last visited: 20.10.2011
[43] Phone Gap: http://www.phonegap.com/, Last visited: 20.10.2011
[44] APPlause: https://github.com/applause, Last visited: 20.10.2011

The target of the solution approach, described in detail in chapter 5, is to offer customers a MAPP, which enables customers to build own mobile native and mobile Web applications. The advantage of the MAPP is that customers without programming skills, e.g. in Java, HTML5, JavaScript, and Objective-C, should be able to build custom powerful mobile applications. The challenge is finding a way to combine technologies like mobile native applications, mobile Web application frameworks (see chapter 2) and open source CMS (see chapter 3). Furthermore the MAPP should enable customers to semi-automatically build mobile native and mobile Web applications for Apple‘s iOS, Google's Android, Windows Phone 7 as well as mobile Web applications based on one single content base.

The CMS (backend) is responsible for data management, application export as well as content-page organisation and customer management (ACL). The mobile clients (frontend) in turn are responsible for displaying the content on the mobile devices and communicating with the MAPP via adapted Web-services. In accordance this section presents the requirements of capital importance towards such a MAPP in form of story cards. The storycards are categorised as follows:

- MAP Customer Interface
- Mobile Application Construction
- MAP Collaboration
- Mobile Application Building
- Mobile Applicatio Publishing
- MAP Communication-Interface

MAP Customer Interface

Story-Card	
ID	1
Date	07.04.2011
Name	Multi Browser Support
Description	The MAPP’s Web-interface shall be multi browser compatible and supported by as many browsers as possible. The main focus being Firefox, Safari, Internet Explorer, and Google Chrome.

Table 1: Multi Browser Support

Story-Card	
ID	2
Date	05.04.2011
Name	Customer Registration
Description	In order to let the customer use the MAPP as fast as possible, the registration process has to be easy and short. Useless, obligatory and time-consuming input-fields like name, surname etc. shall be avoided. Upon entering a valid email-address along with an account password, the customer receives an e-mail containing logging-information and an account activation URL. By clicking on the activation URL the customer completes the registration process and is ready to use the framework.

Table 2: Customer Registration

Story-Card	
ID	3
Date	07.04.2011
Name	Application Dashboard
Description	The Application Dashboard represents the customers's operation base to his MAPP account. The dashboard contains information about customer activities, all applications the cutomer is currently working on and the corresponding application status (in work, submitted, released). Furthermore information about application histories (content-change logs, reviewers, feedback etc) shall be provided.

Table 3: Application Dashboard

Mobile Application Construction

Story-Card	
ID	4
Date	08.04.2011
Name	Add / Delete / Update Content
Description	The MAPP offers the possibility to add, update, delete, publish and unpublish content.

Table 4: Add/Delete/Update Content

Story-Card	
ID	5
Date	07.04.2011
Name	Content Editor
Description	The MAPP is equipped with an intuitive editor for adding, deleting and updating content. One very important aspect of the editor is the customer User Interface (UI). The UI should be as simple as possible, in order to avoid confusion due to useless features. Complexity gets disclaimed consciously.

Table 5: Content Editor

Story-Card	
ID	6
Date	08.04.2011
Name	Choose the Applications Navigation Mode
Description	The MAPP should offer different predefined UI templates. At this point the customer shall be able to choose between these templates. Depending on the chosen mode an image-based preview builds and printed on the MAPP's preview screen. The customer is able to change or try different templates, based on this preview. The MAPP shall constantly show the customer how the edited properties take affect on the applications navigation mode.

Table 6: Choose the Application Navigation Mode

Story-Card	
ID	7
Date	08.04.2011
Name	Add Content Pages & Link the content to appropriate UI elements
Description	The MAPP offers the possibility to add content pages to the application. Every page is able to display content, based on predefined page templates. A content page has a unique content page title, which is also used as page ID. This page ID is mapped on the chosen UI element and be the cross-reference between the chosen navigation mode and the content page. The customer is able to change the order content pages by means of drag & drop.

Table 7: Add Content Pages

Story-Card	
ID	8
Date	08.04.2011
Name	Choose a Content Page Scheme
Description	The MAPP offers the possibility to use predefined color schemes to design the content page layout. These color schemes determine the background color as well as the look of the navigation bar and text within the mobile applcation. The customer chooses a color scheme, which then will be used for all content pages within an application.

Table 8: Choose a Content Page Theme

Story-Card	
ID	9
Date	08.04.2011
Name	Content Page Templates and Interactive Elements
Description	The MAPP offers a set of predefined content templates. These templates determine the overall layout of content within the mobile application. Customers should be able to choose UI elements like Tabbars, Grid Views or Tableviews. Depending on the chosen template the editor displays the corresponding fields to fill with content. The customer uploads images, videos and other interactive content to the MAPP's backend.

Table 9: Content Page Templates and Interactive Elements

Story-Card	
ID	10
Date	08.04.2011
Name	Multilanguage support
Description	Customers should be able to add content pages in different languages. Therefore the CMS and the mobile applications should support different languages.

Table 10: Multilanguage Support

Story-Card	
ID	11
Date	20.08.2011
Name	Location Based Service
Description	The MAPP offers a Location Based Service for the customer. Customers are able to choose a Map template and add corresponding Points Of Interests (POI). Customers should get the possibility to enter informations like name, description, long and latitude values and other detailed information about a POI. The mobile application should be able to display all POIs on a Map, in a List-view as well as over detailed information about all POIs by clicking on one of them.

Table 11: Location Based Services

Story-Card	
ID	12
Date	07.04.2011
Name	Preview Screen
Description	The MAPP's preview screen is designed to show how customers-actions affect the application layout. This preview screen provides the customer with an idea of how the application he is building will look like, either as native or mobile Web application. The preview considers the different mobile operating systems drawing the application layout. The customer is completely integrated in the whole application building process.

Table 12: Preview Screen

MAP Collaboration

Story-Card	
ID	13
Date	05.04.2011
Name	Collaboration
Description	The CMS should offer customers a possibility to communicate with each other over CMS intern communication channels. Therefore the CMS offers a social component for the customers.

Table 13: Collaboration

Story-Card	
ID	14
Date	05.04.2011
Name	Customer Management & „Share“ Accounts
Description	The MAPP supports three roles. Customers can act as administrator, content-maintainer, reviewer or consumers. Administrator: Customers acting as administrators have full access to all components and sections within the framework. Administrators are responsible for building, maintaining and distributing applications. Furthermore they are allowed to create „Share Accounts“ and give unregistered customers access to the MAPP. The main idea of „Share Accounts“ is to offer administrators an easy and fast ways of letting other people (reviewers) review their application before publishing it. Content-maintainer: This role allows customers to change content and content sources of an application. Content-maintainers are able to update the application content. Reviewer: Reviewers have limited access to applications. They are allowed to test applications and give the administrators feedback. Reviewers may not change the application layout, structure, navigation or content. Consumer: The people using the mobile applications.

Table 14: Customer Management & Share Accounts

Mobile Application Building

Story-Card	
ID	15
Date	09.04.2011
Name	Build a Mobile Web Application
Description	Concluding the editing process, the customer has the possibility to publish the content as a mobile Web application. Simply by clicking the export button the MAPP semi-automatically builds the mobile Web application und print the application URL to the customer. The mobile Web application can be hosted by the MAPP or by the customer itself. If the customer wishes to host his mobile Web application on his own domain he is able to download a whole package as output of the export process containing all mandatory data, frameworks, HTML-pages etc.

Table 15: Buil a Mobile Web Applications

Story-Card	
ID	16
Date	09.04.2011
Name	Application Signing Space
Description	The customer can choose between either downloading a native mobile application package, and sign it on his own, or using the MAPP to sign the application. The MAPP offers its own provisioning area to the customer. This area allows uploading provisioning profiles and keystores (depending on the mobile operating system) as wall as automated signing of the package by the MAPP. The MAPP offers the possibility to upload external application packages and sign these packages with the corresponding keys and profiles. This feature is very useful for customers who don‘t want to download the SDK of the corresponding mobile operating system just for signing and testing applications on a device.

Table 16: Application Signing Space

Story-Card	
ID	17
Date	09.04.2011
Name	Export Mobile Native Applications
Description	After the editing process, the customer has the possibility to publish his content as a mobile native application. By clicking the export button the framework semi-automatically builds the application bundle containing all mandatory information. The customer is able to download an application bundle (unsigned), which can be submitted to the App Store.

Table 17: Export Mobile Native Applications

Mobile Applicatio Publishing

Story-Card	
ID	18
Date	11.04.2011
Name	Mobile Advertising
Description	The customer should get the possibility to embed mobile ads in his mobile applications and mobile native applications.

Table 18: Mobile Advertising

Story-Card	
ID	19
Date	11.04.2011
Name	Hosting of Mobile Web Applications
Description	The MAPP offers customers the possibility to rent webspace, which will be hosted by the MAPP. The customers can use this webspace for hosting their mobile Web applications. Thus customers can quickly publish their mobile Web applications and start distributing the application URL subsequently.

Table 19: Hosting of Mobile Web Applications

Story-Card	
ID	20
Date	09.04.2011
Name	Social Networking
Description	The MAPP shall give the customer the possibility to share applications via Facebook, Twitter as well as on other relevant platforms.

Table 20: Social Networking

Story-Card	
ID	21
Date	09.04.2011
Name	Search Engine Optimization
Description	Search engine optimization supports the customer improving the visibility of his mobile Web applications in search engines by optimizing metadata

Table 21: Search Engine Optimizaion

MAP Communication-Interface

Story-Card	
ID	22
Date	09.04.2011
Name	Web-Service & Web-Interface
Description	The MAPP offers a communication interface (Web-service) which can be used by clients for desired content. This Web-service is able to offer the data as JSON object as well as XML feeds (XML feeds can be used in other mobile application frameworks).

Table 22: Web-Service & Web-Interface

Story-Card	
ID	23
Date	07.04.2011
Name	Synchronisation Mechanisms
Description	In order to keep the contents on the mobile Web applications and mobile native applications up to date, and give the customer the possibility to update content, the Web-service shall offer a possibility of requesting updated or new data manipulated since the last data synchronisation (system-time-stamps). Thus the CMS shall safe all changes on a content page by the date and server system time, minimizing unnecessary data traffic.

Table 23: Synchronisation Mechanism

Story-Card	
ID	24
Date	07.04.2011
Name	Multidevice Support
Description	To achieve multi device support the framework shall include a component, which is able to adapt the data depending on the device, which is asking for the data. Corresponding to different display sizes, resolutions and performance reasons data adaption has to be handled on the server side of the system.

Table 24: Multi Device Support

The listed requirements outline the complexity involved in CMS based MAPP.

4.2 Challenges Developing Cross-Platform Mobile Applications

Based on the requirements presented in section 4.1, this section outlines the challenges in integrateing and implementing a cross-platform deployment module for mobile applications on an existing CMS. Therefore, this section groups the requirements from section 4.1 as follows:

- Multi Browser Support & Multidevice Support
- Web-service & Web-Interface
- User Management
- Data Collection Over CMS
- UI Design via CMS
- Export of Mobile Application
- CMS Intern Communication & Mobile Applications Publishing ..

Multi Browser & Multi device Support

Customers access the MAPP via Mozilla Firefox, Google Chrome, Safari, Internet Explorer and mobile browsers. Therefore the CMS's UI should be supported and look equal on all of these browsers. To fulfill these requirements the MAPP should implement a CMS, which is based on, common and open Web technologies.

The next problem concerns heterogeneous devices entailing different mobile operating systems, offering different software development kits (SDK). Above all that, most of these different operating systems are incompatible to each other. As already mentioned in chapter 2 mobile native applications have big problems with heterogeneous devices. Since producers are offering a tantamount of new devices, the problem of incompatibility within one operating systems is overlapping also mobile operating systems like iOS and WP7. It is worth to mention that a huge number of different devices with specific display resolutions, and different hardware make the problem of developing mobile applications for different mobile operating systems more comprehensible. The framework should be able to handle this heterogeneity by building cross-platform applications.

Web-Service & Web-Interface

The MAPP enables customers to update the content on their mobile applications over the CMS backend. Accordingly customers creating mobile native applications will be able to update their applications without running through the appliction stores publishing process. For this reason the MAPP has to offer an adequate Web interface with corresponding Web services. These Web services enable the mobile applications to fetch new content from the platform.

According to the fact that one of the requirements is to offer reporting functionalities about his applications, the platform Web-services should offer the corresponding methods. The mobile applications should be able to send information about the consumer behaviour to the MAPP. The Web-services should be able to route this information to the adequate customer account. All this information should be displayed on the customer's application dashboard.

The Aapplication Dashboard is the home screen and therefore the customer‘s entering point to the MAPP. The Dashboard presents an overview over all applications a customer is working on. Furthermore customers can get detailed information about total downloads, active downloads, passive downloads, mobile advertising clicks, colaborators, reviewers, change logs, crash logs, device families, the application is currently running on as well as other statistics

User Management

One of the main requirements towards the MAPP is to let different customers work together on a mobile application. To achieve this the MAPP should offer a mechanism to authenticate and identify different customers. The platform should offer an adequate Web interface allowing customers to register for the MAPP's services. This Web interface enable the customer to experience a fast and easy way of registering and start creating own mobile applications. At this point the CMS offers a customer registration the backend should be able to handle different customers. Accordingly an adequate user management mechanism for keeping track of these customers should be guaranteed.

Since registered customers should be able to collaborate, the CMS should enable customers to invite other customers participating in their applications building process. Therefore the CMS should offer a communication channel for the customers.

Collaboration on mobile applications entails the support of, Access Control Lists‘. Since customers have different rights on different applications,

the CMS should equip customers with appropriate user permissions. Corresponding to their user permission, customers will be able to perform special tasks on the CMS. Accordingly, sections bounded with appropriate user permissions (admin, reviewer etc.) should be visible only to the customers with the required permission. Accordingly the CMS UI should be possible to be bounded to the user permissioning.

Data Collection over CMS

Since customers should be are able to add, update, and delete content, the CMS should be able to deal with the created content in the appropriate way. The CMS should have an elaborated database model for archiving all information about customer's applications with the according content pages. The CMS-Editor UI should be as simple and user-friendly as possible. This should be achieved by focusing on main editing and formatting functionality. The customer shouldn't be confused by too specific functionality he won't use.

Another important UI element coming into effect during the creation process of a mobile application using the MAPP is the preview screen. This screen is responsible to show the customer how the application he is working on will look like on the different mobile operating systems. Furthermore the preview screen is designed to show how customer-actions affect the application layout. This preview screen will provide the user with an idea of what the application he is building will look like, either as native or mobile Web application. The challenges of the preview screen is to find the right technology which can be embedded in the CMS UI and which is able to display mobile application via the Web browser on demand and without an own building process.

UI Design via CMS

Offering the possibility to export mobile application on different mobile platforms implicates complex UI structures. Therefore it is necessary to analyze all UI elements on the supported MAPPs (iOS, Android, Web applications) OS, in order to identify the least common denominator in respect to UI elements. According to the fact that mobile applications should follow the mobile OS UI system guidelines, finding common views on all mobile platforms will be very difficult. Regarding these UI system guidelines it is quite obvious that it is not possible to build mobile applications looking exactly the same on all mobile operating systems. Omitting mobile OS specific UI elements make similar looking applications on different mobile platforms imaginable. Elements like Tabbars, Lists and Grid View look equal on all actual mobile operating systems.

Since one requirement is to enable customers to use LBS, one UI element are maps. The challenge using maps is to define an adapted format for storing the POI' content, and bring in this content on the mobile application. Figuring out other UI elements equal on all mobile OS will entail the adequate studies in the terms of UI design. Furthermore UI elements should offer the possibility to edit different layout parameter over the CMS-Editor. The affect should be displayable on the preview screen.

The solution presented in chapter 5 will describe a template-based approach of defining the mobile applications UI. Customers should be able to enter content depending on the chosen user-interface templates. Therefore the CMS should have a sophisticated mechanism to encode the chosen template settings in a specific way that the mobile application can use to draw display the appropriate UI elements.

Export of Mobile Application

The MAPP enables customers to semi-automatically generate a mobile application out of the CMS, based on the created content pages. Furthermore the customer gets the possibility to choose between an export as mobile native application on iOS or Android, or as a mobile Web application. Accordingly the CMS has to be equipped with an powerful cross-compiler, which should be able to automatically generate application binaries for all platforms based on the customers content. The main challenge creating cross-platform applications, is defining a DSL for compiling all binaries based on this language.

CMS Intern Communication & Mobile Applications Publishing

Customers should be able to share mobile applications with other customers (reviewers). Accordingly the CMS has to offer an adequate „social" channel supporting multi-channel publishing and handling of platform intern communications.

Since the MAPP offers the possibility to publish mobile applications directly to the adequate application market (App store, Android Market), the CMS has to offer an own section for handling the whole application submitting process. When publishing iOS applications the appropriate provisioning profiles and developer certificates are needed. These certificates and profiles are used for application signing and compiling. Publishing Android applications assumes the application KeyStores with the appropriate passwords for signing and compiling the application. Mobile Web applications on the other hand should be

able to be published directly to the customer's domain. This in turn means that the CMS should offer an own domain hosting services.

4.3 Summary

This chapter introduced the idea of a Mobile Application Publishing Platform (MAPP) to automatically building mobile native applications and mobile Web applications out of an existing (and "well-known") CMS. This chapter gave an overview on some relevant requirements towards such a MAPP. As mentioned in Section 4.2, the requirements listed in section 4.1 point out the complexity of a CMS based MAPP. Thus, Section 4.2 focused on making the challenges involved in implementing multi-platform mobile applications more comprehensible.

The MAPP should be supported and look equal on all popular browsers. Accordingly, the MAPP implements a CMS, which is based on common and open Web technologies. Due to the fact that customers should be able to update the application content via the CMS the MAPP has to offer an adequate Web interface with corresponding Web-based services. The Web-based services should be able to route information to the adequate customer account for reporting functionality. According to this the CMS has to offer the adequate „social" channel supporting multi channel publishing and handling platform intern communications. CMS should equip customers with appropriate user permissions. Accordingly, the CMS UI should be possible to be bounded to the user permissioning. The CMS should have an elaborated database model for archiving all information about customer's applications with the according content pages.

Furthermore, this chapter discussed the challenge of building "same looking" mobile applications on all mobile operating system. Regarding the User Interface system guidelines it is quite obvious that it is not possible to build mobile applications looking exactly the same on all mobile operating systems. Omitting mobile OS specific UI elements make similar looking applications on different mobile platforms imaginable

Since the application domain of the prototype implementation presented in chapter 6 is Location Based Services, the presentation of the POI's on mobile devices (see table 11) has to be analized. Thus, an adapted format for storing the POI's content has to be defined. The CMS should have a sophisticated mechanism to encode the chosen POI content-template settings in a specific way that the resulting mobile application is able to understand.

Chapter 5: Easy Made Mobile Applications (EMMA)

Based on the findings and conclusions outlined in the previous chapters, this chapter deals with the design of a Mobile Application Publishing Platform (MAPP) system architecture. This platform is called EMMA - "Easy Made Mobile Applications" – and its main features will be discussed in more detail below, followed by a brief overview of its component-module based architecture. EMMA's system architecture is the cornerstone of the prototype implementation described in chapter 6.

5.1 A First Solution Approach

EMMA is a MAPP that enables non-programmers to generate custom mobile native applications and mobile Web applications. This MAPP fulfils the requirements presented in the previous chapter. EMMA offers customers a Web-interface, that allows non-programmers to build their mobile applications once they are registered for the MAPP. Furthermore, EMMA allows customers to provide template-based content pages (i.e. the content of a page is arranged on predefined templates) and to display these on mobile devices. Based on these content pages, EMMA enables its users to build mobile native applications (running on iOS, Android and Windows Phone 7) as well as to build mobile Web applications. From a scientific point of view, the content and goals of this chapter focus on the challenges of homogenizing the heterogeneity in the mobile device market (see chapter 2) by building template based cross-platform mobile applications.

Figure 4 illustrates how a customer may use a MAPP like EMMA to generate a mobile application. EMMA efficiently supports and simplifies this process. In addition, a CMS intern preview-screen serves to monitor the effect of user-action on the application's layout for both native and mobile Web application.

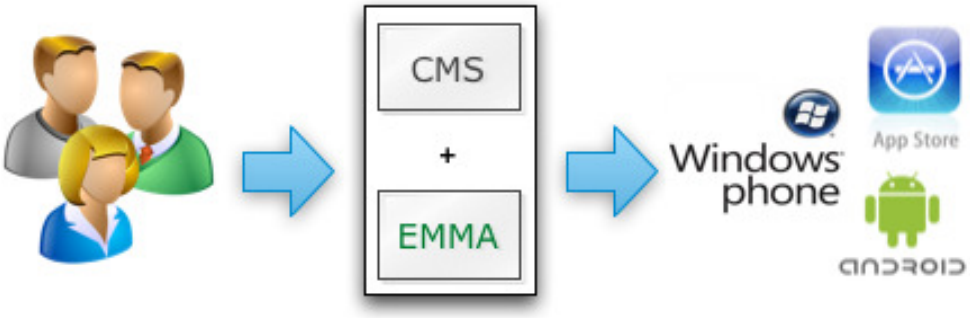

Figure 4: Custom Mobile Applications with EMMA

The process of developing a mobile application is time-consuming and expensive. EMMA offers a possibility to generate and publish content on mobile devices in a very simple, fast and cheap way. Programming skills are not a prerequisite for using EMMA. Customers don't need to install complex IDEs, nor do they have to run through complex application signing and provisioning processes or application submitting processes. EMMA handles the entire development process for the customer. As mentioned in chapter 2 it is rather difficult to decide which mobile technology to use when developing an application. Both a mobile native application offering access to operating systems specific system resources, which guarantee flexible solutions with high performance, and HTML 5 based mobile Web applications provide a native like experience on a mobile device through the Web. EMMA is able to generate mobile native applications as well as mobile Web applications based on the customer's content pages. With EMMA both the native and the Web approach are feasible at the same time.

From the technological point of view EMMA can be divided into two components:

- CMS responsible for the Web-interface allowing customers to generate template-based content pages
- Mobile application clients responsible for displaying the content on mobile devices.

EMMA's CMS

EMMA uses the open source CMS - "SilverStripe" - as a server-side CMS-backend. "SilverStripe" is a very customer- and developer-friendly CMS which uses common open source technologies and open Web standards like PHP, MySQL, HTML, CSS and Javascript. The adoption of these technologies, the elaborated system architecture and a clean and well-documented source code

make this CMS future-proof (see chapter 2). “SilverStripe” uses a flexible user permission management to support collaboration between content authors. Different authors are thus able to work and manage the same content. Furthermore “SilverStripe” offers a very simple and clean content editor, which is straightforward to use and not overloaded with unnecessary functionality. Accordingly, “SilverStripe” is predestined to be used as MAP CMS-backend.

EMMA’s Mobile Application Clients

EMMA comprehends mobile applications using Domain Specific Language (DSL)[45]. DSL allows solutions to be expressed more clearly and closer to domain. Furthermore it enhances understanding and communication. Therefore EMMA uses an open source library, named APPlause[46], which places an adequate cross compiler based on DSL at our disposal. APPlause is a cross-platform mobile development toolkit consisting of DSL for defining mobile apps and code generators for creating native apps for iOS, Android, Windows Phone 7 and Google App Engine.

Mobile Web applications can be created using the Sencha Touch mobile Web application framework. Sencha Touch is a very powerful cross-platform framework aimed at next generation, touch enabled, devices using standard technologies like HTML5, CSS3 and JavaScript for the highest level of power, flexibility, and optimization. In addition, Sencha Touch supports gesture recognition patterns like “touchstart” and “touchend”, “tap”, “double tap”, “swipe”, “tap and hold”, “pinch” and “rotate” (see chapter 2).

5.2 System Architecture

> „Quality, time, and cost are the three central factors determining the success or failure of any software project, and quality is the only one of those factors that can not be changed on the spot by management fiat. In addition, the effects of poor software quality can be dramatic and difficult to undo.” [Spinellis 2006]

In order to provide good software quality EMMA has to define some key characteristics to be geared to. Suitable functions, accurate results and a high

[45] DSL: http://www.itwissen.info/definition/lexikon/DSL-domain-specific-language-Domaenenspezifische-Sprache.html, Last visited: 01.10.2011
[46] APPlause: https://github.com/applause/applause, Last visited: 01.10.2011

degree of data security are to be attained. EMMA maintains a high level of performance and recovery concepts to achieve software reliability. EMMA's UI is designed to let customers perform actions in just a view steps. Therefore characteristics like understandability, learnability and operability an integral part of EMMA's UI design. Regarding EMMA's system architecture and software code one of the main targets is to offer a maintainable, analyzable, changeable and testable platform. Finally, the most important requirement towards software design is portability. According to the adoption of open Web standards EMMA can be used on all platforms. [Spinellis 2006]

In order to provide a reusable, reliable, efficient, maintainable, portable and extensible system, a flexible and elaborated architecture is needed. The next part introduces a first solution approach of EMMA's system architecture:

- EMMA's System Architecture in General
- EMMA's Mobile Application Client
- EMMA's CMS - Frontend
- EMMA's CMS - Backend

5.2.1 EMMA's System Architecture in General

The main goal of this section is to provide insight into the principles behind EMMA's system architecture (see figure 5). The architecture is based on areas, components and modules. Each area is arranged into system components, and these components are again divided into different modules. EMMA can be divided in three main system areas:

- The Mobile Application Client
- The MAP CMS - Frontend
- The MAP CMS – Backend

Mobile Application Client

The mobile application client is the final product of the mobile application development process. The client is responsible for fetching all the data from the CMS and displaying this data in the appropriate way. As long as the customer is able to choose different templates for displaying the content, the client is able to understand these templates and generate the adequate views. These views are going to be drawn on the mobile device's screen.

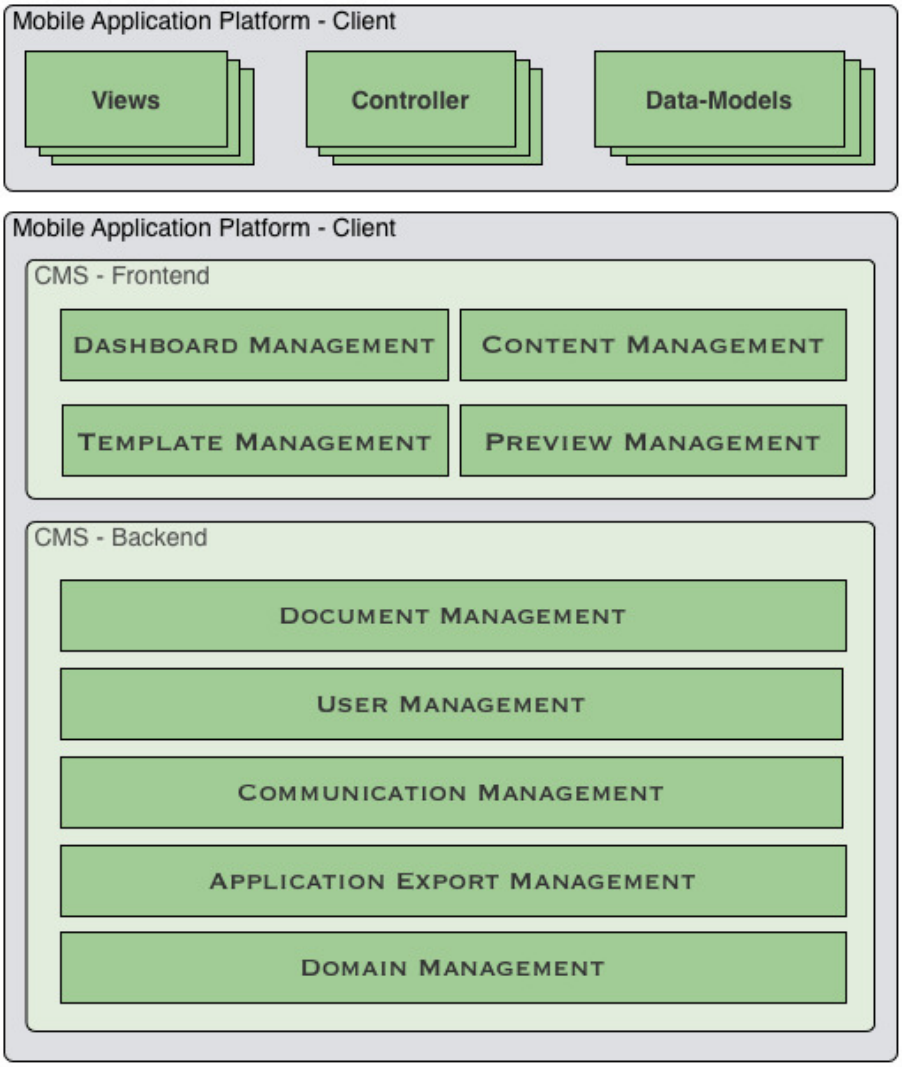

Figure 5: An outline of EMMA's system architecture

MAPP CMS-Frontend & Backend

EMMA's CMS forms the core of the MAPP system. The CMS offers a Web-interface for customers to work with the MAPP. Furthermore this area offers components for data management (generate, edit, delete content) and enters different types of data on the CMS. The CMS is able to handle different types of data and offers the adequate document interface for the customers. Furthermore the CMS enables customers to collaborate with other customers by offering an own user permissioning system and using Access Control Lists. By using secure communication channels customers will be able to communicate with each other and use EMMA as a communication platform.

5.2.2 EMMA's Mobile Application Client

The main goal of this section is to give a detailed explanation of the mobile application client's system architecture. The purposed architecture is based on a model-view-controller approach (see figure 6).

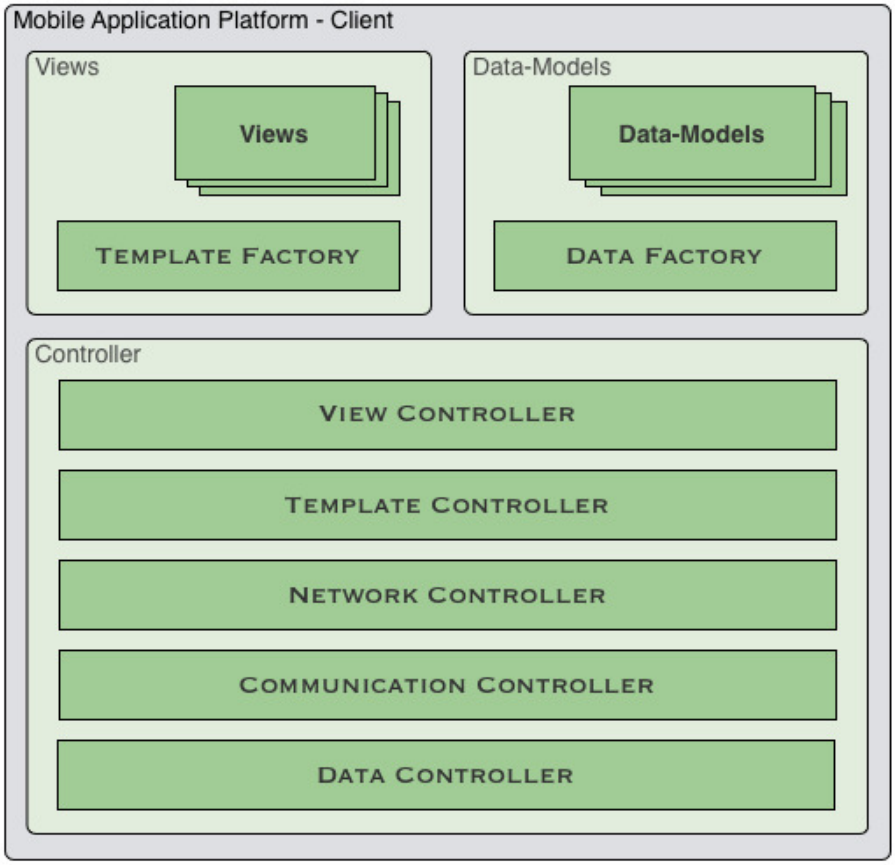

Figure 6: Mobile Application Client Architecture

Views

The View component is responsible for creating, managing, and displaying views on the mobile device. A Template factory module, implemented as a Factory pattern, enables the views component to decide which view to create and allocate. This pattern provides a clear mechanism to design a flexible UI mechanism. The Template factory contains predefined templates for arranging UI elements as well as filling these elements with the customer's content.

Data-Models

The mobile application client possesses a Data-Models component for saving the customer's content fetched from EMMA's CMS Web-service (see chapter 4). The

most important module in this component is the Data factory, which is also implemented as a Factory pattern. The Factory pattern enables the mobile application client to encapsulate the database creation process, which allows the system to change the database type easily. Furthermore this module is responsible for managing the data flow between the database and data models by using Object Relational Mapping (ORM). This design pattern enables dataflow and mapping between the data-models and the database. The design pattern used to implement this content to database mapping is called Active Record design pattern. The main idea of the ActiveRecord design patterns is to represent a database table at code level by using appropriate data objects. Each object maps a row in a specific database table (If a new data object is allocated, the database table creates a new row; if a data object is updated the appropriate content in the database table is updated automatically).

Controller

The Controller component is the core component of the mobile application client. This component is responsible for transferring the Data between the Views and the Data-Models. An appropriate View Controller module, Templates Controller module and Data controller module are available. These three controllers fulfil the data-view mapping thus implementing the Model-View-controller pattern. This separation allows the model to be built and tested independently of the visual presentation.

Further to this, the mobile application client offers a Network Controller module and a Communication Controller module. The Network Controller module is responsible for connecting the mobile application client to the Web. In addition, the Network Controller offers a Web interface for executing data-requests. The Communication Controller in turn is responsible for creating these data-requests, implementing the communication protocol, and for forwarding the fetched backend data to the data controller.

5.2.3 EMMA's CMS – Frontend

To reduce the complexity of the CMS's system architecture, the system is divided into the CMS-frontend and the CMS-backend (see section 5.2.4). Within the scope of our CMS-frontend (see Figure 7) there is a:

- Content Management Component
- Template Management Component

- Dashboard management Component
- Preview Management Component

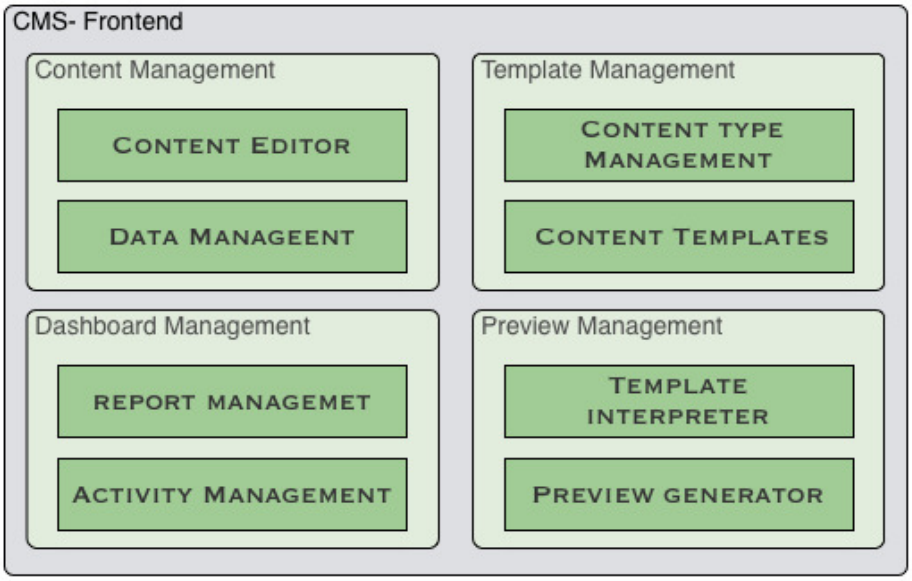

Figure 7: EMMA's CMS-Frontend

Content Management Component

The Content Management's main function is data collection. This component consists of the Content Editor module, and the Data Management module. The Content Editor is responsible for enabling customers to add content by providing the adequate input forms specific for the chosen template. The Data Management module is responsible for saving the customers content in the adequate Data-Models and prepares the content for the CMS-backend's Document Management Component.

Template Management Component

The Template Management component is responsible for preparing predefined templates and presenting the way in which content can be arranged on the mobile device. The Template Management component consists of the Content Templates module and the Content Type Management module. The Content Templates module is required for archiving the predefined templates. The Content Type Management module is responsible for saving the customers chosen template settings. Furthermore this module prepares the template settings for the Preview Management component.

Preview Management component

The Preview Management component is responsible for supporting the customer throughout the application development process. The component consists of a Template Interpreter module and a Preview Generator module. The Template Interpreter module collaborates with the Template Management component in preparing the chosen customer templates for the Preview Generator module. The Preview Generator is the core module of the Preview Management component and responsible for displaying the actual mobile application UI visual appearance on a preview screen. This module is implemented as a Model View Controller pattern surrounded by an observer pattern. The MVC pattern delivers the content to the Preview Generator views. The observer pattern in turn is responsible for updating the preview screen when new content is available or content has been edited.

Dashboard Management component

The Dashboard Management component is responsible for processing the customer's activities and his mobile applications. This component consists of a Report Management module and an Activity Management module. The Report Management module fetches activity data from the Activity Management module and displays the data on the corresponding views. The Activity Management module is the interface to the CMS backend's database and responsible for collection and preparation of all customer-account dependent reporting data.

5.2.4 EMMA's CMS – Backend

The core of the overall system architecture is the CMS-backend (see figure 8), consisting of:

- Document Management Component
- User Management Component
- Communication Management Component
- Application Export Component
- Domain Management Component

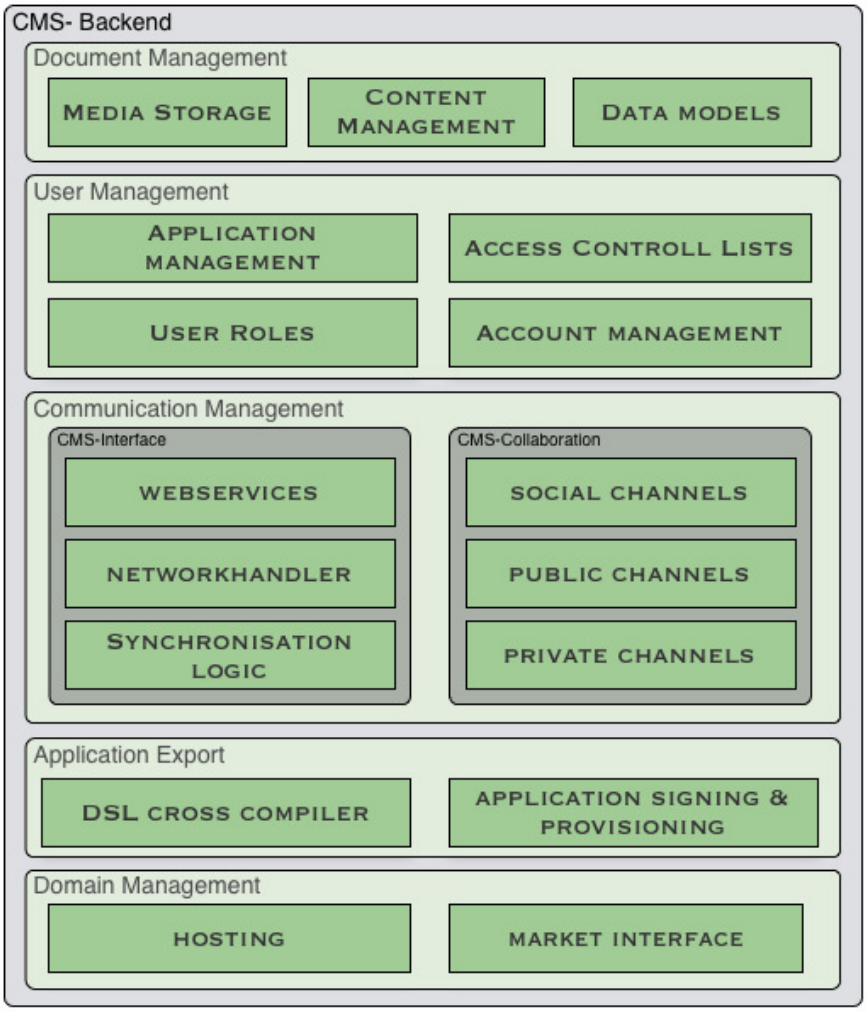

Figure 8: EMMA's CMS-backend

Document Management Component

The Document Management Component consists of the Content Management module, the Media Storage module and the Data-Models module. The Content Management module collects, organises and saves content added by the customer. Taking into consideration that different customers should be able to edit shared content this module stays in close collaboration with the user management component. The Media Storage module prepares multi media content to fit the customer's chosen templates. The Data Models module is the interface to the CMS database. This module is responsible for adding, updating, deleting and fetching content from the database. In addition, this module handles the content prepared by the Content Management component.

User Management Component

The User Management component is based on the Access Control Lists module, a User Roles module, an Account Management module, as well as an Application Management module. The ACL consists of Access Control Entries (ACE). Each ACE in an ACL identifies a customer and specifies the access rights allowed, or denied for the MAPP. Furthermore the ACL module works in close collaboration with the User Roles module and the Account Management module. The Account Management module is responsible for mapping customers to their accounts on the CMS. The User Roles module is responsible for tagging different customer with user roles. These user roles will equip the customer with specific rights within the CMS. As mentioned in chapter 4 users may act as administrators, content-maintainers or reviewers. The Application Management module handles different customers collaborating on one mobile application. This module decides which customer is allowed to work on the application as an administrator, content-maintainer or reviewer.

Communication Management Component

The Communication Management component is divided in the CMS-Interface part and the CMS-Collaboration part. The CMS-Interface consists of the Web-service module, the Networkhandler module and the Synchronisation module. The CMS-Interface is responsible for network connection activities and the communication with the mobile application clients. Therefore the Web-service module offers an adequate Web-interface. The Synchronisation module implements the communication protocol and synchronization mechanism.

The CMS-Collaboration part contains the Social Channel module, the Public Channel module and the Private Channel module. The CMS-Collaboration is responsible for CMS intern customer communication. The Social Channel module enables customers publishing content on social platforms like Facebook or Twitter. The Public and Private Channel modules are concerned with the communication within channels. These modules offer a public channel where a customer's "friends" will be able to "see" one another and they also offer the possibility to start private channels with selected people.

Application Export Component

The Application Export component builds the customer's mobile application. This component offers a DSL Cross-Compiler module, and an Application Signing & Provisioning module. The DSL Cross-Compiler module is able to

generate a DSL specified file based on the chosen template and content settings. This DSL file allows the cross-compiler to semi-automatically build a mobile application for different mobile operating systems like Android, iOS and WP7. The role of the Application Signing & Provisioning module is to manage several provisioning profiles, certificates, and keystore files for all mobile operating systems that the customer may have uploaded through the CMS. This module prepares the necessary data for the cross-compiler's application singing process.

Domain Management Component

Given that customers should be able to publish their applications directly to the Application Stores or publish mobile Web applications this component offers a Domain-Hosting module. Therefore EMMA also provides a CMS intern Domain Hosting service enabling the publication of mobile native applications directly on the App Store, Android Market and other mobile application stores. This abolishes the need not go through complicated publishing procedures. EMMA, will build the application, sign the binary with the adequate certificates and profiles and submit the application binary for review on the application market. Within the User Management Account module customers are able to enter application market specific user data, which are needed for login and uploading processes.

5.3 Summary

EMMA's system architecture is based on areas, components and modules. Each area is arranged in components, and these components again into modules. As mentioned in Section 5.1, the main incentive for developing EMMA was to offer a new innovative and feasible way of building powerful mobile applications on multiple mobile platforms by combining different established technologies. Section 5.2.2 focused on the system architecture of EMMA's mobile application client. The client's system architecture is dominated by a Model-View-Controller design pattern, which allows the model to be built and tested independent of the visual presentation. Furthermore, the View component of the mobile application client area is implemented as a Factory design pattern, which offers a convenient way to expand different view templates, and provides a clear mechanism to design a flexible and expandable UI mechanism. The data controller uses Object Relational Mapping and Active Record design pattern to map the data with the database and a Factory pattern to offer the right database to he controller.

In contrast, the CMS-frontend consists of the Content Management component, which is responsible for data collection and the Template Management component, which is responsible for preparing predefined templates and presenting how content can be arranged on the mobile device. The Preview Management component is responsible for supporting the customer throughout the application development process by displaying the actual mobile application UI visual appearance on a preview screen. The Dashboard Management component is responsible for processing the customer's activities and his mobile applications.

The core of the overall system architecture is the CMS-backend. The CMS-backend consists of the Content Management Module, which is responsible for organizing, collecting and saving the customer's content. The User Management component is based on the Access Control Lists module, a User Roles module, an Account Management module, as well as an Application Management module and is responsible for the user permissioning mechanism and collaboration activities. The CMS-Collaboration is responsible for CMS intern customer communication. The Social Channel module allows the customer to publish content on social. The Public and Private channel modules are responsible for the communication within channels. The Application Export component is responsible for semi-automatically building the customers mobile application.

The target of this chapter was to provide a component-module based architecture for MAPP. The focus of the next chapter is to implement a MAPP prototype system in the application domain of LBS.

Chapter 6: EMMA in Action

Based on the system description of EMMA in chapters 5, this chapter deals with the evaluation of the solution approach, regarding ease-of-use for a specific application domain. A complete implementation of EMMA would go far beyond the scope of this Master Thesis. Therefore, this chapter shows the implementation of the main parts of EMMA as a prototype system and focuses on a specific mobile application development process to efficiently support non-programmers creating mobile applications in the application domain of Location Based Services (LBS). Thus, section 6.1 introduces the context of the field of LBS and the Google Maps Application Programming Interface (API). Section 6.2 lists the basic requirements to simulate the mobile application development process. Section 3.2 presents the evaluation of the developed prototype system, focusing on ease-of-use and briefly outlines the evaluation results.

6.1 Working with Maps

Customers are getting over flooded with new powerful heterogeneous mobile devices. Due to the maturation and convergence of technologies and the fact that these devices are equipped with large displays, global positioning system (GPS) sensors, and mobile networks location based, services are getting very popular in the mobile market. This section presents the application areas of LBS in combination with Maps. [Grossniklaus et al 2006]

6.1.1 Location Based Services

> „Location-based services (LBS) are the delivery of data and information services where the content of those services is tailored to the current or some projected location and context of a mobile user.“ [Brimicombe and Li 2009]

The communication interface between customers and LBS are digital Maps. Maps represent the presentation layer where customer can select, classify, filter and display spatially related content within a certain context. The questions of capital importance LBS want to answer are: [Grossniklaus et al 2006]

- Where am I?
- Who/what is near me?
- Where is a certain location?
- How can I get there from my current position?

Where am I and who/what is near me?

The main functionality of LBS is to tell the customer his actual location (specified implicitly by a tracking sensor), and possibly interesting locations in his/her surroundings. LBS based mobile applications like Foursquare[47], Facebook- Places[48] and Gowalla[49] enable customers to „check in" at locations in the customer's surrounding. Customers are able to see where friends checked in and communicate with these via the application. [Grossniklaus et al 2006] Another popular aspect of LBS is of community-driven location based recommendation services like Qype[50], and Yelp[51]. Customers get location-proposals and are able to read other customers reviews about those locations. [Brimicombe and Li 2009]

Where is a certain location and how can I get there from my current position?

LBS provide the functionality to tell the customer his/her actual location, based on the data fetched from the GPS sensor. The customer gets the possibility to search other locations and tell the mobile device to display the route to a target location. Most popular LBS based mobile applications on mobile devices are Google Maps. Google Maps offers an appropriate API, which enables developers to use maps for LBS functionality. [Grossniklaus et al 2006]

[47] Foursquare: https://de.foursquare.com/, Last visited: 07.10.2011.
[48] Facebook-Places: http://www.facebook.com/about/location, Last visited: 07.10.2011.
[49] Gowalla: http://gowalla.com/, Last visited: 07.10.2011.
[50] Qype: http://www.qype.com/, Last visited: 07.10.2011.
[51] Yelp: http://www.yelp.com/, Last visited: 07.10.2011.

Consequently this section outlines some daily scenarios of location-based services. These scenarios are grouped in LBS activities:

- Navigation: Navigation applications on mobile devices enable customers to find routes to pre-specified destinations with a vehicle.
- Wayfinding: Mobile applications supporting customers finding routes to pre-specified destinations, landmarks and POIs
- Real-time tracking: So called „buddy systems“ allow customers to track friends.
- Near Field Communication (NFC): NFC and e-commerce allowing customers to make local money transaction
- User-solicited information: Mobile applications that offer data about weather, traffic and transport conditions
- Geofencing: Field-based activities for local ordering and delivery
- Digital expressions: Customers tag locations (location tagging) with digital expressions (digital graffiti).
- Mobile gaming: Gamers are able to play together in a digital world through location-based services. [Brimicombe and Li 2009]

6.1.2 Google Maps

The Google Maps API[52] enables mobile developers to use Google maps in their mobile native applications and mobile Web applications. This section gives an overview of the main advantages and disadvantages using the Google Maps API in mobile native applications and mobile Web applications.

Google Maps API in Mobile Web Applications

Android and iOS based mobile devices (HTML 5 and JavaScript support) support the Google Maps API within their Web browser. Advantages of the Google Maps API embedded in a mobile Web Application are:

[52] Google Maps API: http://code.google.com/intl/de-DE/apis/maps/index.html, Last visited: 07.10.2011.

- One source for all platform (desktop and mobile devices)
- On demand content updates
- W3C-Geolocation[53] support on some mobile devices (iOS and Android)

Disadvantages of using the Google Maps within mobile Web applications are:

- Does not support compass, and
- Accelerometer sensors

Google Maps API in Mobile Native Applications

Currently only Android and iOS offer Google Maps API support for their developers. Advantages of the Google Maps API in a mobile native application are:

- No Webserver running
- Better performance
- Local device storage is available to store relevant content for offline mode
- Full sensor support

Disadvantages of the Google Maps API in a mobile native application are:

- Implementation is complex
- Content updates only possible with an application update
- Users may be unsettled by additional permissions
- Different sources for different mobile operating systems
- Not as much features as the JavaScript based Google Maps API for mobile Web applications

[53] W3C-Geolocation: http://dev.w3.org/geo/api/spec-source.html, Last visited: 07.10.2011.

6.2 Requirements and User Roles

The application domain chosen for the prototype system is LBS. This section defines the main requirements regarding the prototype's implementation. Since chapter 4 defined different user roles for EMMA (see table 14), the requirements are grouped in:

- Requirements for Customers Acting as Administrators (i.e. developers of mobile applications)
- Requirements for Mobile Application Consumers (i.e. the endusers of the deployed location-based service)

Requirements for Customers Acting as Administrators

As mentioned in chapter 4, customers acting as administrators have full access to all components and sections within the MAPP. Further they are responsible for building, maintaining and distributing applications. Therefore the prototype's implementation envelops the following requirements:

- EMMA offers a member login area (see table 2), where customers are able to authenticate themselves with a unique username and password (see figure 9).

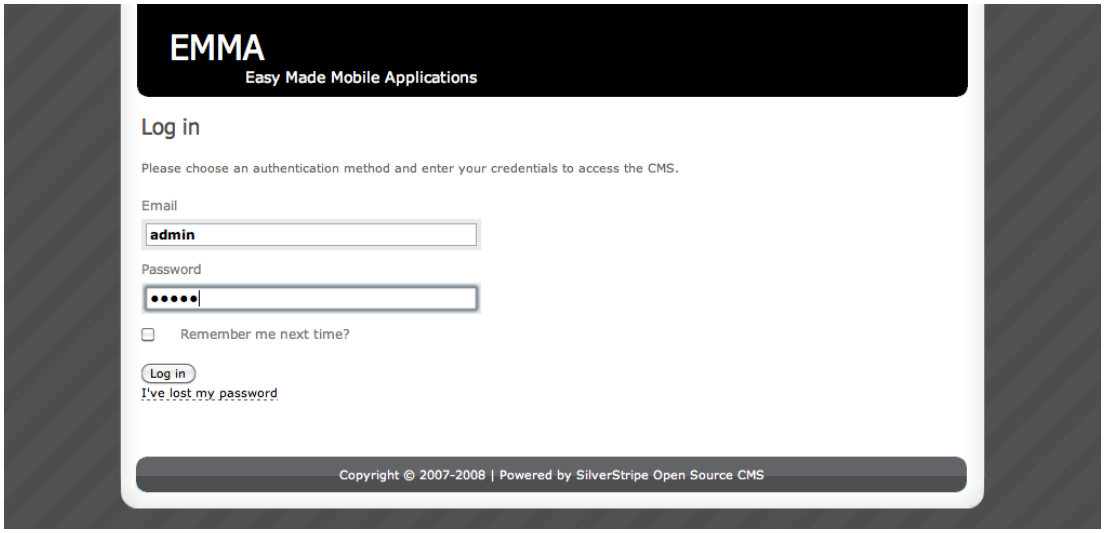

Figure 9: EMMA's Customer Login Area

- Regardless of whether the customer wants to build an iOS or Android mobile application, the first step is to enter an application name and choose an application icon (see table 3). Furthermore, this Page enables

them to export the mobile application either for iOS or Android (see figure 10).

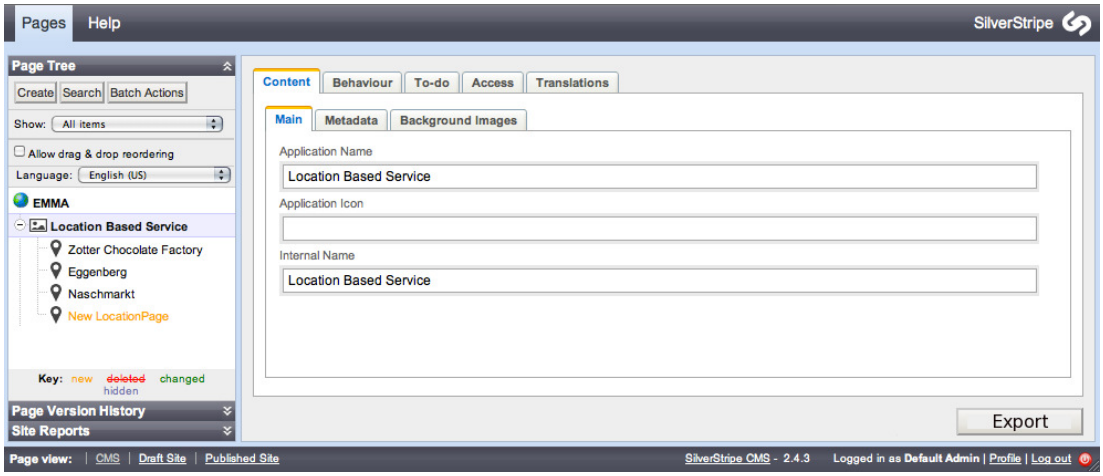

Figure 10: Mobile Application Dashboard

- Customers should be able to create new POIs (see table 7). POIs should offer the possibility to add general location information (see figure 11), location-specific information (see figure 12), images (see figure 13), as well as multimedia content like audio and video (see figure 14)

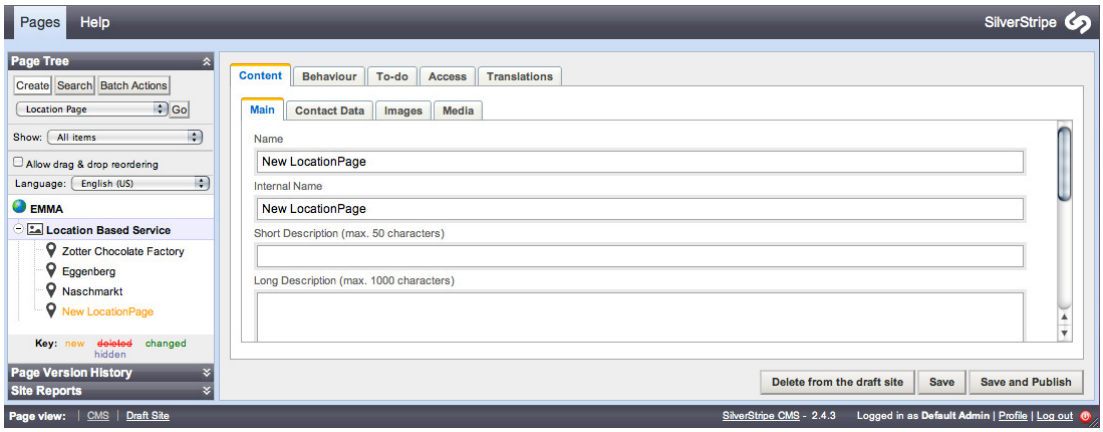

Figure 11: Location-Specific Content (Part 1)

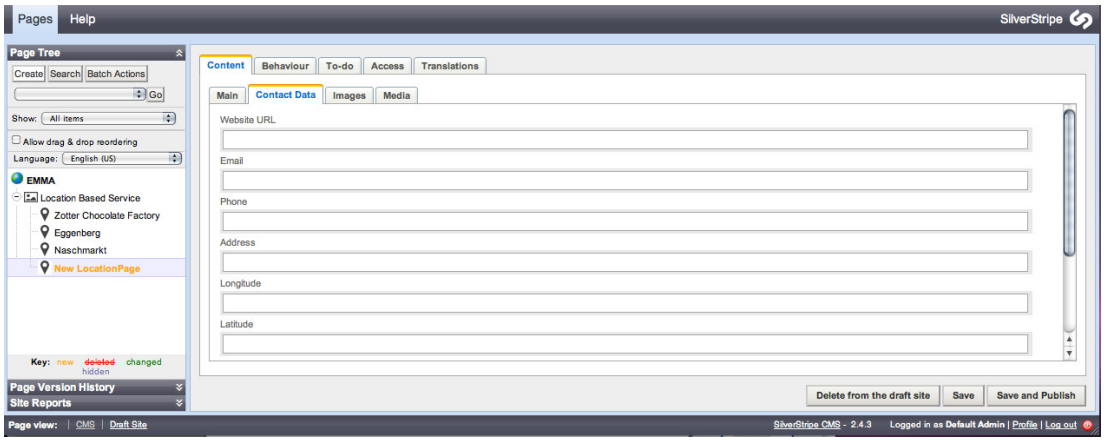

Figure 12: Location-Specific Information (Part 2)

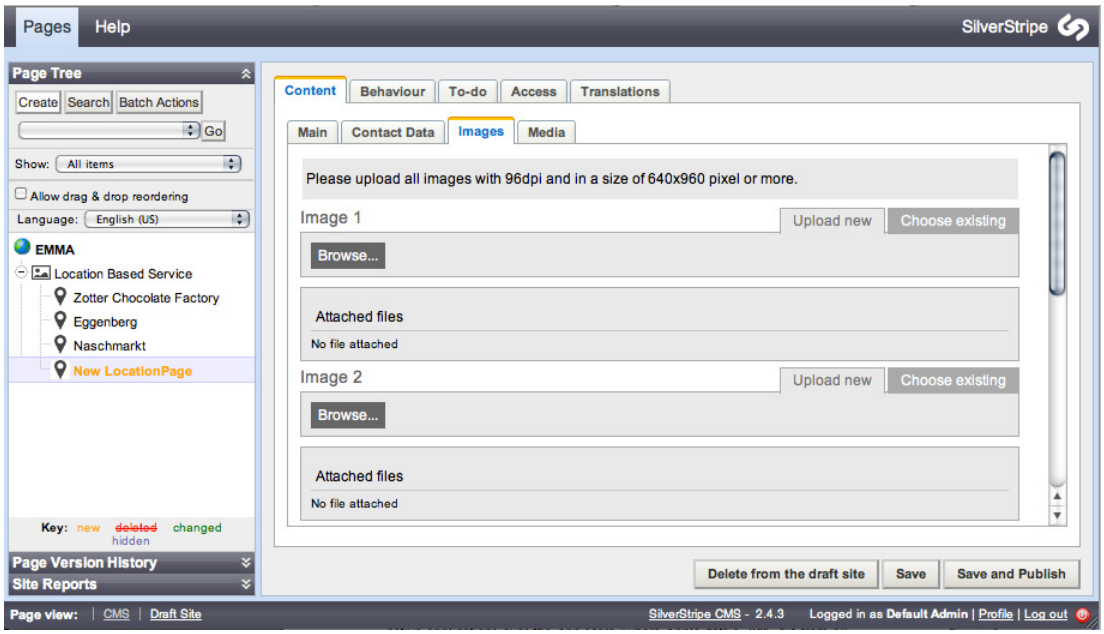

Figure 13: Location-Specific Information (Part 3)

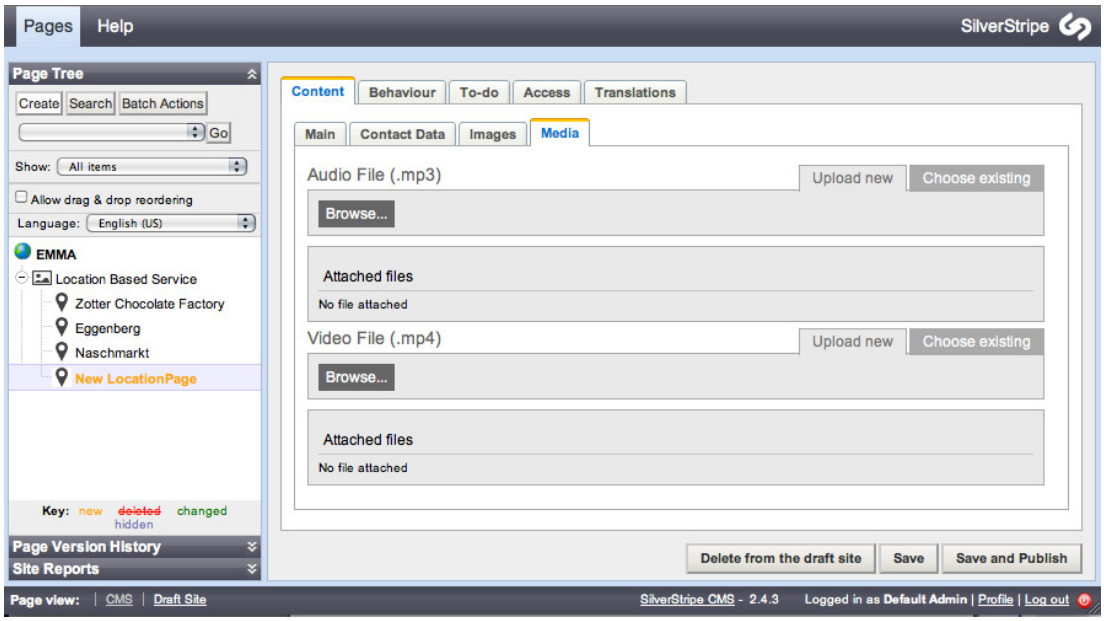

Figure 14: Location-Specific content (Part 4)

- Customers should be able to collaborate (see table 14) with other customers (see figure 15).

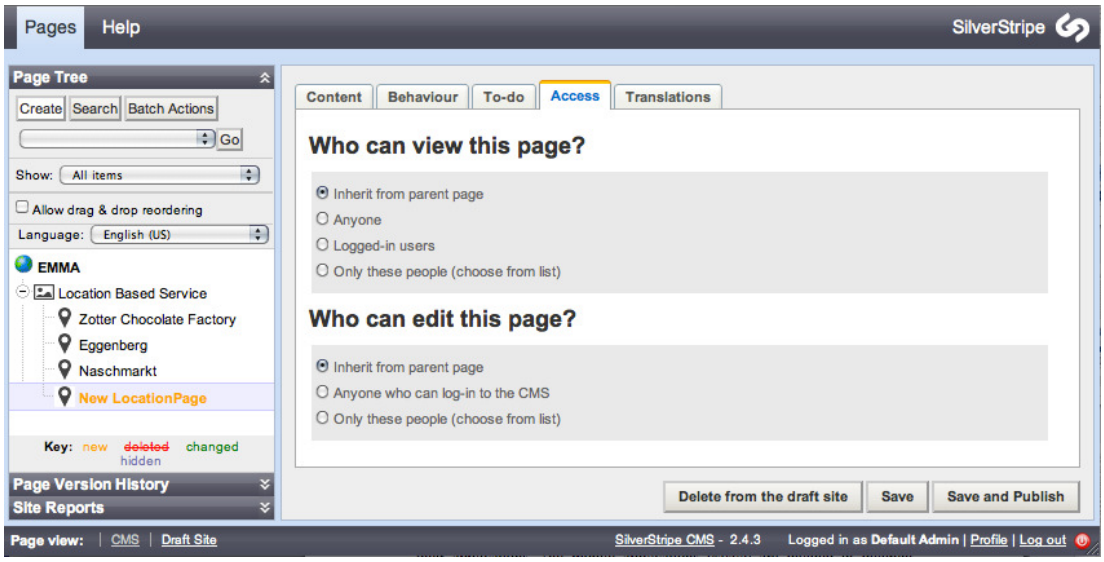

Figure 15: Access Controll Lists

Requirements for Mobile Application Consumers

Within the range of this master thesis the mobile application client is implemented for Android and iOS. This section gives an overview of the mobile application client's prototype implementation:

- Like mentioned in chapter 4 (see table 11) the mobile application client should be able to display all POIs on a map-view (see figure 17 and figure 19) as well as a on a list-view (see figure 16 and figure 18).

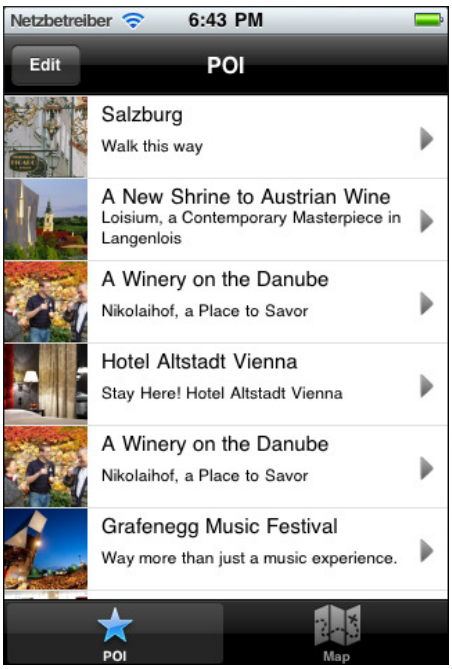

Figure 16: iOS POIs on a List-View

Figure 17: iOS POIs on a Map

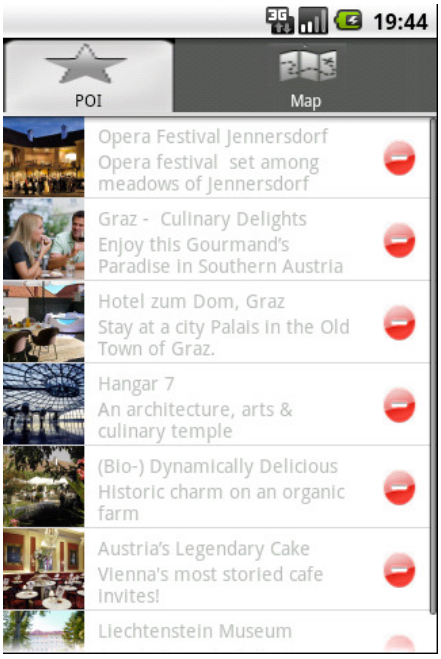

Figure 18: Android POIs on a List-View

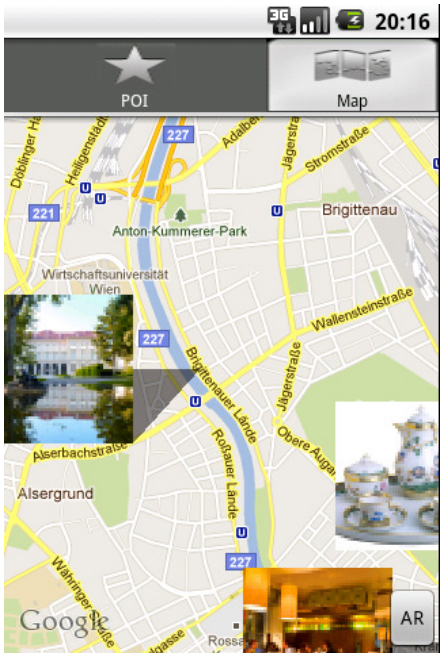

Figure 19: Android POIs on a Map

- As well, the mobile application client offers specific screens for showing a POI's content (see table 11) based on the predefined template (see figure 20 and figure 21).

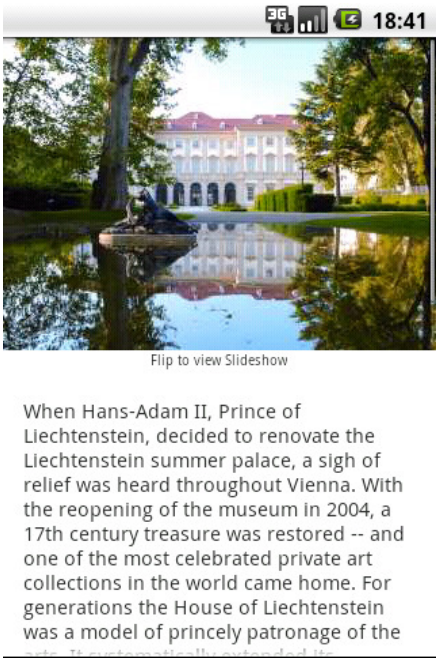

Figure 21: Android POI Details

Figure 20: iOS POI Details

6.3 EMMA – Evaluation Results

This section analyzes the results of a customer expectation evaluation regarding ease-of-use of EMMA's prototype implementation. The target audience of this evaluation is represented by smartphone users without technical background. The purpose of this evaluation is to gather information about:

- The straightforwardness of EMMA's CMS-frontend (see section 5.2.3)
- The expectations of the participants regarding the resulting mobile application

The evaluation is classified into three phases. The first phase deals with the introduction of EMMA. The next phase deals with the adoption of EMMA's CMS-frontend by giving the participant certain tasks to accomplish. The third phase consists of an interview focusing on the participants' expectations among the resulting mobile application. The next step is to briefly outline the main evaluation results (detailed information about the evaluation setup and interview protocols can be looked up in Appendix B).

The aim of the evaluaion was to find out if the participants are able to easily create new locations using EMMA's prototype CMS-frontend, and to which extent the resulting mobile application cope with their expectations. Based on the observations during the evaluation, as well as on the feedback of the participants the most relevant findings can be summarised as follows:

- The participants were fascinated by the clean UI and simple content-editor while using the CMS-frontend.
- However, the participants had problems in understanding how the actions they took affected the mobile applications layout. Therefore a preview-screen (see chapter 5) would be very helpful.
- Participants had problems filling in location specific input fields. Subject-specific terms like „longitude“,“„latitude“ and „internal name“ should be avoided.
- Regarding the resulting mobile application, it can be stated that nearly all expectations were met. Besides this, the participants were fascinated by the Augmented Reality component.
- Moreover, one of the main requirements the participants expected of an LBS based mobile applications is navigation. The mobile application should be able to propose routes to selected locations.

- Almost all participants required the possibility to share locations over the CMS. Furthermore the CMS should offer a pool of „public“ locations, which can be applied by the customers.

Due to the fact that all participants confirmed having been able to easily create own locations and have easily (and intuitively) used the resulting mobile application client, the results of this evaluation attest the easy-of-use and usefulness of EMMA’s prototype implementation.

6.4 Summary

This chapter dealt with the presentation and evaluation of the prototype system regarding ease-of-use in the application domain of LBS. Section 6.1 presented the application areas of LBS in combination with Maps and the Google Maps API. As mentioned in Section 6.1, maturation and convergence of technologies and the fact that mobile devices are equipped with large displays, global positioning system (GPS) sensors, and mobile networks, LBS is getting very popular in the mobile market.

Section 6.2 defined the main requirements regarding EMMA’s CMS prototype implementation as well as the resulting mobile application clients. The requirements were grouped based on the user roles (administrator and consumer) defined in chapter 4.

Section 6.3 analyzed the results of a customer expectation evaluation regarding ease-of-use of EMMA’s prototype system. The purpose of this evaluation was to gather information about the straightforwardness of EMMA’s CMS-frontend (see section 5.2.3) and the expectations of the participants regarding the resulting mobile application. From the findings depicted in Section 6.3, it can be stated that the participants were positively impressed by the clean UI and simple content-editor of EMMA’s CMS-frontend. Further, the participants said that they have required a Preview-screen (see chapter 5) to monitor the effect of user-actions on the application’s layout. In addition, based on the observations during the evaluation, it could be deducted that subject-specific terms like „longitude“,„latitude“, „internal name“ should be avoided.

Regarding the features of the deployed mobile application, participants were fascinated by the Augmented Reality component. Furthermore, the participants expected the mobile application to propose routes to selected locations as well as offer the possibility to share locations with other people.

Chapter 7: Summary & Outlook

This chapter outlines and lists all the results obtained from the previous chapters. Based on the findings and conclusions of chapter 5, this chapter also presents the main features concerned with an upgrading of the existing MAP architecture.

7.1 Summary

In short, developing a multi-device mobile application and building up the needed know-how to efficiently utilise the corresponding SDKs is quite expensive and time consuming. Companies are facing the problem on which mobile technology to use for delivering content, covering as many mobile devices as possible. This thesis addressed this problem by identifying and coping with the challenges of choosing whether to publish content through a mobile Web application or mobile native application. Therefore, different mobile application frameworks and Web based Mobile Application Platforms had been analyzed. The target of this thesis was to present a solution for an overall system architecture of a Mobile Application Publishing Platform (MAPP) for cross-plattform, user-friendly, semi-automatically deployment of mobile applications through CMS in the application domain of Location Based Services (LBS). Therefore, in order to show the roadmap utilised towards the proposed and developed solution, this thesis was structured and dealt with the encountered challenges as follows.

Chapter 2 outlined the strengths and weaknesses of mobile native applications and mobile Web applications as well as focused on Mobile application platforms and mobile Web application frameworks. Chapter 2 showed that no solution is currently available that targets at non-programmers, and that provides easy-to-use Web-based User Interfaces for the design and development of cross-platform mobile applications.

The goal of chapter 3 was to render in a comprehensible the challenges of choosing an available open source CMS to be used as basis for the solution approach described in chapter 5. Section 3.1 outlined that SilverStripe is a very

customer- and developer-friendly CMS with an elaborated system architecture and very simple and clean Content Editor. From the findings depicted in chapter 3, it can be stated that SilverStripe is predestined to be used as CMS within the solution approach in chapter 5.

Chapter 4 gave an overview on some relevant requirements on a MAPP that efficiently supports non-programmers in the mobile application development process. Thus, this chapter pointed out the complexity of a CMS based MAPP and provided an overview over the main challenges involved in implementing multi-platform mobile applications.

Based on the findings and conclusions outlined in chapters 2, 3 and 4, the target of chapter 5 was to present a possible system architecture of a MAPP. This chapter outlined how to design and implement the mobile application development process to efficiently support non-programmers in creating mobile applications. In order to provide a reusable, reliable, efficient, maintainable, portable and extensible system, the solution approach followed well-established software technologies and diverse application-specific design patterns. Furthermore, this chapter proposed a DSL based approach of semi-automatically compiling template based cross-platform mobile applications for homogenizing the currently existing heterogeneity in the mobile device market.

Chapter 6 showed the implemented prototype system based on a specifc and storycard-based mobile application development process to efficiently support non-programmers in creating mobile applications within the application domain of LBS. Furthermore, chapter 6 demonstrates an evaluation of the solution approach, regarding ease-of-use in the application domain of LBS.

7.2 Open Issues

Since the specific application and research within the scope of this thesis was represented by the field of LBS, some critical open issues in the general context of this paper are worth mentioning. Regarding the different components of EMMA's architecture, the Communication Management component as well as the User Management component has to be defined in more detail and reanalyzed. Factors and implications such as who may edit an applications content, who may be in one communication channel and who may listen to certain conversations need to be clarified and evaluated. Furthermore, a security and privacy-enhanced component has to be efficiently integrated,

More efforts has have to be invested in the Application Export component containing the DSL Cross-Compiler module, and the Application

Signing & Provisioning module (see Section 5.2.4). It should be analyzed to what extend it is possible to sign the application binaries outside of the SDK's and upload application binaries directly to the application markets. To successfully implement the whole proposed system architecture the DSL Cross-Compiler module has to be enhanced to fulfill EMMA's requirements regarding different templates (see section 5.2.2).

List of Figures

List of Tables

Bibliography

[A3Webtech 2011] A3Webtech. Drupal CMS vs Joomla CMS Comparison. A3Webtech. 2011. Available at: http://www.a3Webtech.com/index.php/compare-drupal-v-joomla-cms.html. Last visit: 18.08.2011.

[Balaji 2011] Balaji. Native Vs Web Mobile Application. Android Technology. MAY 3, 2011. Available at: http://sbalaji-android.blogspot.com/2011/05/native-vs-Web-mobile-application.html. Last visit: 08.06.2011.

[Borschart and Schommer 2009] Steven Borschart and Ingo Schommer. SilverStripe, Das umfassende Handbuch. Galileo Computing. 2009.

[Brimicombe and Li 2009] Allan Brimicombe and Chao Li. Location-Based-Services and Geo-Information Engeneering. Wiley-Blackwell. 2009.

[Caroll 2010] Eileen Carroll. Mobile applications: native v Web apps – what are the pros and cons?. mobiThinking. September 2010. Available at: http://mobithinking.com/native-or-Web-app. Last visit: 08.06.2011.

[Conors and Sullivan 2010] Adam Connors and Bryan Sullivan. Mobile Web Application Best Practices. W3C. December 14, 2010. Available at: http://www.w3.org/TR/mwabp/. Last visit: 08.06.2011.

[Dachis 2011] Adam Dachis. AppMakr Helps You Create Your Own iPhone App for Free, No Coding Necessary. Lifehacker. October 26, 2010. Available at: http://lifehacker.com/5673309/appmakr-helps-you-

create-your-own-iphone-app-for-free-no-coding-necessary. Last visit: 08.06.2011.

[DogHousemedia 2009] DogHouseMedia. Mobile Web Applications. DogHouseMedia. 2009. Available at: http://www.dhmedia.com.au/technology/mobile-Web-applications. Last visit: 08.07.2011.

[Dohrmann 2010] Rolf Dohrmann. Softwareentwicklung für Smartphones mit PhoneGap. July 20, 2010. Available at: http://blog.rolfdohrmann.de/2010/07/softwareentwicklung-fur-smartphones-mit-phonegap/. Last visit: 08.06.2011.

[Durga 2007] Apoorv Durga. Content Management for Mobile Delivery. PCM-Blog. MAY 26, 2007. Available at: http://apoorv.info/2007/05/26/content-management-for-mobile-delivery/. Last visit: 18.08.2011.

[Dryabov 2008] Dryabov. Mobile Sites with Joomla! CMS. mobiForge. 2008. Available at: http://mobiforge.com/developing/story/mobile-sites-with-joomla-cms. Last visit: 19.08.2011.

[Jeavons and Knaddison 2010] Benjamin James Jeavons and Gregory James Knaddison. Drupal Security White Paper. Cc-by-nd. July 27, 2010.

[eMarker 2010] eMarketer. Mobile Users Prefer Browsers over Apps. eMarketer Digital Intelligence. OCTOBER 27, 2010. Available at: http://www.emarketer.com/Article.aspx?R=1008010. Last visit: 26.04.2011.

[Evans 2011] Jonny Evans. Red Foundry: Is this App development for the rest of us?. Computerworld. January 26, 2011. Available at: http://blogs.computerworld.com/17721/red_foundry_is_this_app_development_for_the_rest_of_us. Last visit: 08.06.2011.

[Ferner 2011] Matt Ferner. 17 Solutions to Build Your Own Mobile App. Practical Ecommerce. February 08, 2011. Available at: http://www.practicalecommerce.com/articles/2573-17-Solutions-to-Build-Your-Own-Mobile-App. Last Visit: 26.04.2011.

[Garcia-Barrios et al 2009] Victor Manuel Garcia-Barrios, Korab Qerkini, Christian Saffran. What students really need beyond learning content: ubiquitous shared-connectivity services to foster learning communities on the campus. ACM. 2009.

[Garfield 2010] Larry Garfield. Design Patterns of Drupal. Drupal Watchdog. Available at: http://drupalwatchdog.com/1/1/design-patterns-of-drupal. Last visit: 18.08.2011.

[Graf 2006] Hagen Graf. Drupal – Community-Websites entwickeln und verwalten mit dem Open Source-CMS. Open Source Library. 2006.

[Grossniklaus et al 2006] Michael Grossniklaus, Moira C. Norrie, Beat Signer and Nadir Weibel. Putting Location-Based Services on the Map. 2006. Available at: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.73.896. Last visit: 20.09.2011.

[Harvey 2010] Sean Harvey. SilverStripe mobile development has just become easier. SilverStripe. August 25, 2010. Available at:http://www.silverstripe.org/silverstripe-mobile-development-has-just-become-easier/. Last visit: 19.08.2011.

[Johnston 2010] Mike Johnston. SilverStripe mobile development has just become easier. CMS Critic. August 26, 2010. Available at: http://www.cmscritic.com/silverstripe-mobile-development-easier/. Last visit: 26.04.2011.

[Kaminitz 2011] Shahar Kaminitz. Native, Web or Hybrid App Development?. Scribd. March 15, 2011. Available at: http://www.scribd.com/doc/50805466/Native-Web-or-Hybrid-Mobile-App-Development. Last visit: 06.06.2011.

[Kathryn 2010] Kathryn. Mobilize Your Drupal Site With Mobile Tools. Digett. May 24, 2010. Available at: http://www.digett.com/2010/05/21/mobilize-your-drupal-site-mobile-tools. Last visit: 19.08.2011.

[Linn 2010] Michele Linn. The Dos and Don'ts of B2B Content Reuse. Savvy B2B Marketing. March 30, 2010. Available at: http://www.savvyb2bmarketing.com/blog/entry/594191

/the-dos-and-donts-of-b2b-content-reuse. Last visit: 26.04.2011.

[Lohr 2011] Steve Lohr. In a New Web World, No Application Is an Island. Th New York Times. March 26, 2011. Available at: http://www.nytimes.com/2011/03/27/business/27unboxed.html?_r=3. Last visit: 26.04.2011.

[MacManus 2006] Richard MacManus. Mobile Web applications - do they need the browser?. Read Write Web. March 23, 2006. Available at: http://www.readwriteWeb.com/archives/mobile_Web_appl.php. Last visit: 08.06.2011.

[Makhija 2009] Amit Makhija. Drupal Vs Joomla Whitepaper. Info Beans. April 25, 2009. Available at: http://www.scribd.com/doc/9735899/InfoBeans-Drupal-vs-Joomla-Whitepaper. Last visit: 26.04.2011.

[Marriott and warning 2011] Jennifer Marriott Elin Waring. The Official Joomla!TM Book. Addison-Wesley. 2011.

[Marshall 2011] Matt Marshall. How HTML5 will kill the native app. Mobile Beat. April 7, 2011. Available at: http://venturebeat.com/2011/04/07/how-html5-will-kill-the-native-app/. Last Visit: 26.04.2011.

[Marshall 2009] Matt Marshall. Google's mobile jihad: Support the Web, but live with the app. Venture Beat. May 12, 2009. Available at: http://venturebeat.com/2009/05/12/googles-mobile-jihad-support-the-Web-live-with-the-app/. Last Visit: 26.04.2011.

[Merce 2008] David Mercer. Building powerful and robust Websites with Drupal 6. Packt Publishing. March, 2008.

[Metz 2011] Cade Metz. Apple handcuffs 'open' Web apps on iPhone home screen. Developer. March 15, 2011. Available at: http://www.theregister.co.uk/2011/03/15/apple_ios_throttles_Web_apps_on_home_screen/. Last visit: 26.04.2011.

[MMA 2008] MMA - Mobile Marketing Association. Mobile Applications. September, 2008. Available at: http://www.mmaglobal.com/mobileapplications.pdf. Last visit: 08.06.2011.

[Pimmel 2011] Kim Pimmel. Making It a Mobile Web App. Smashing Magazin. January 26, 2011. Available at: http://www.smashingmagazine.com/2011/01/26/making-it-a-mobile-Web-app/. Last visit: 08.06.2011.

[Price 2009] Chris Price. Compare CMS - Part 1 – Introduction. A3Webtech. October, 2009. Available at: http://www.a3Webtech.com/index.php/compare-cms.html. Last visit: 18.08.2011.

[Rowehl 2010] Mike Rowehl. "App" Doesn't Have to Mean "Native". September 16, 2010. Available at: http://www.thisismobility.com/blog/2010/09/16/app-doesnt-have-to-mean-native/. Last visit: 08.06.2011.

[Sanjay 2008] Sanjay Acharya. Worldwide mobile cellular subscribers to rach 4 billion mark late 2008. International Telecommunication Union. 2008. Available at: http://www.itu.int/newsroom/press_releases/2008/29.html. Last visit: 16.10.2011.

[Sæterås 2010] Jon Arne Sæterås. Mobile Web vs. Native Apps. Revisited. Mobiletech. April 9, 2010. Available at: http://www.mobiletech.mobi/blog/mobile-Web-vs-native-apps-revisited/11672/. Last visit: 08.06.2011.

[Spinellis 2006] Diomidis Spinellis. Code Quality: The Open Source Perspective. Addison Wesley. 2006. Available at: http://www.spinellis.gr/codequality/. Last visit: 30.09.2011.

[Tetzchner 2010] Jon von Tetzchner. State of the Mobile Web. December, 2009. Opera Software. Available at: http://www.opera.com/smw/2009/12/. Last visit: 26.04.2011.

[Vayen.ch 2011] Vayen.ch. Updating iOS Applications from CMS. Vayen. January 24, 2011. Available at: http://blog.vayen.ch/2011/01/24/cms-ios-update/. Last visit: 26.04.2011.

[Webhostingfan 2010] Webhostinfan. Open Source CMS: Expandability Results in the Growth of Your Business. Webhostingfan. January 21, 2010. Available at: http://www.Webhostingfan.com/2010/01/open-source-

cms-expandability-results-in-the-growth-of-your-business/. Last visit: 18.08.2011.

[Yared 2010] Peter Yared. iPhone now as fragmented as Android. Mobile Beat. June 9, 2010. Available at: http://venturebeat.com/2010/06/09/iphone-now-as-fragmented-as-android/. Last visit: 26.04.2011.

Appendix A

EMMA - Introduction

The achievements in the field of mobile devices such as smartphones and tablets have deserved great attention over the last years. Contemporaneously, the development of the technical capabilities of these mobiles devices proceeds rapidly. Further, the demand on mobile devices is growing remarkably. Customers are getting over flooded with new powerful heterogeneous mobile devices. Due to the maturation and convergence of technologies and the fact that these devices are equipped with large displays, Global Positioning System (GPS) sensors, and mobile networks, Location Based Services (LBS) are getting very popular in the mobile market. In addition, the mobile market segment developed a high rate of mobile technologies for the development of mobile applications. EMMA (Easy Made Mobile Applications) is one of these mobile technologies. EMMA is a Mobile Application Publishing Platform (MAPP) that enables the creation of custom mobile applications in the application domain of LBS. EMMA offers a well strcutured and simple web-interface, to create content (location pages) for LBS based mobile applications in an efficient way. Based on these location pages EMMA is able to build mobile applications, running on Apple's iPhone and Android-Devices.

Assignments:

- Task 1: Login using "admin" as e-mail and password, and cotton up with the different sections of EMMA

- Task 2: Create a location about the "Stephansdom" and enter the adequate information about this location. Use the Internet to gather information about the Stephansdom!

Interview:

- Question 1: What are your first impressions on EMMA?
    - Have there been any difficulties to fulfill the given tasks?
    - Did you miss something (functionality/design/description) in order to fulfill the tasks sucessfully?
    - Is there something (functionality/design/description) you would dismiss in order to make the system more convenient?
- Question 2: What are your expectations regarding the resulting mobile application?
- Question 3: Did the mobile application fulfil your expectations?

Protocols

Participant 1	
Assignment	**Notes**

Task 1	• Imporove wording
	  ○ Longitude
	  ○ Latitude
	  ○ Location pages

Task 2	• Validation failed – but why?
Interview	
Question 1	• Help screen • Application preview screen • Clean UI • Simple editor
Question 2	• Show locations in customers surroundings • Navigation and routing • Augmented reality
Question 3	• The participant loved the mobile application • Augmented Reality screen works perfect • Show the distance to a certain location. • Easy to use • Collaboration features would be very useful • Share locations with friends • Where are my friends

Table 25: Protocol - Participant 1

Participant 2	
Assignment	**Notes**

Task 1	• Imporove wording
	  o Longitude
	  o Latitude
	  o Internal Name

Task 2	• Participant wasn't able to create a new „Location page" • Wasn't able to publish a new location – Validation failed
Interview	
Question 1	• Help screen • Preview screen • Mark text fields as required or optional • Public locations
Question 2	• Live information • Notifications about near locations • Navigation • Sharing locations y
Question 3	• The participant loved the mobile application • Social component and live feeds • Augmented Reality screen should show the distance to a certain location

Table 26: Protocol - Participant 2

Participant 3	
Assignment	**Notes**
Task 1	
Task 2	
Interview	
Question 1	• Improve exception handling • More optional fields. • Location specififc should be divided from other CMS settings • UI design • Easy to create new locations
Question 2	• List with the locations • Location pictures and descriptions • Rank locations based on distance • Search locations on the device • Map view for all locations • Save location on the device • Rate locations from other users
Question 3	• Impressed by the mobile application • Totaly fulfills expetations • Combination of list, map and detail screen is awesome - high resolution images • Favourite section • Categorise locations • Augemented Realiy component is fabolous

Table 27: Protocol - Participant 3

Participant 4	
Assignment	**Notes**
Task 1	

Task 2	• Wasn't able to create a new location page immediately
	• Imporove wording
	  o Longitude
	  o Latitude
	  o Internal Name

Interview	
Question 1	• UI was confusing • Imporve exception handling • Participant wasn't able to upload location – validation failed • Partcipant wasn't able to detect why the validation failed. • All fieds should be optional
Question 2	• Navigation • Fetch infos about locations online • Checklist of the locations already visited • Categorisation • Own templates for details.
Question 3	• The mobile application completely fulfills his exceptaitons. • Categories would be nice • Ranking would be nice

Table 28: Protocol - Participant 4

Participant 5	
Assignment	**Notes**
Task 1	• To many CMS specific settings deflected the participant • User permissioning - how to invite other users
Task 2	
Interview	
Question 1	• Simple and clear editor • Easy to create and organize new locations • Improve wording • Offer an easy way to find long and latitude values of a location • Integrated Google map in the CMS • Exception handling needs to be better.
Question 2	• Live feeds. • Navigation to locations • Locations on a Map with location pictures • Location order important – distance, alphabet
Question 3	• The mobile application completly fulfills the participants' exceptaitons. • Listview is very usefull • Picturegallery with high res images about an location • Public locations

Table 29: Protocol - Participant 5

Participant 6	
Assignment	**Notes**
Task 1	
Task 2	
Interview	
Question 1	• Easy to use • Public locations • Simple editor • Clear UI • Tourist guide
Question 2	• Augmented reaity with detailed information (see Wikitude) • Public locations • Own locations highlighted
Question 3	• Very nice and easy to use mobile application • Fascinatd by the Augmented Reality component • Navigation • List and Map based presentation convinced

Table 30: Protocol - Participant 6

Participant 7	
Assignment	**Notes**
Task 1	• Colourful UI • Navigating only through location specific sections • Didn't find the sample locations
Task 2	• Missed the validation dialog
Interview	
Question 1	• Imporve exception handling • All fieds should be optional • More templates
Question 2	• Detailssscreen with interactive elements • Image gallery • Video players • Locations presented on a map • Routes to different locations • Shortest path covering a certain group of locations.
Question 3	• Participant liked the swith between map and list • Was impressed by the Augmented Reality component • Distance to locations • Share locations

Table 31: Protocol - Participant 7

Participant 8	
Assignment	**Notes**
Task 1	• Clear UI • Maybe too technical wording
Task 2	
Interview	
Question 1	• Other application domains • Different templates • Public locations • Sharing information with others • Ranking locations • EMMA should offer different location-categories
Question 2	• Category-based representation of locations • Show locations on the map
Question 3	• The mobile appliction exceed the participants exceptaitons • Fascinated by the Augmented Reality component • Show the distance to locations • Show only certain categories of locations on the map

Table 32: Protocol - Participant 8

Participant 9	
Assignment	**Notes**
Task 1	
Task 2	
Interview	
Question 1	• Hard to find the application main node • Simple way of entering new locations • Simple way of searching loctions within the CMS • Improve wording
Question 2	• Show all loctions on a map. • Routing to different locations • Groupe locations in categories
Question 3	• Mobile application is perfect • Include categories • Augmented Reality component is realy useful

Table 33: Protocol - Participant 9

Participant 10	
Assignment	**Notes**
Task 1	• Simple UI
Task 2	• Public location pool
Interview	
Question 1	• Improve wording • Easy way to find long and latitude values oa a loctio!? • Integrated google map in the CMS • Different templates • Sharing locations
Question 2	• Show locations on a map • Is anybody in my surrounding • Group my locations • Public locations
Question 3	• Nice mobile application • Augmented Reality component very impressive and good working • Useless without live content

Table 34: Protocol - Participant 10