Haaga-Helia
ammattikorkeakoulu

Journal of Mobile Multimedia, Vol. 12, No.1&2 (2016) 138-158

RAPID PROTOTYPING OF A MOBILE SAAS APPLICATION

ARI ALAMÄKI AMIR DIRIN

Haaga-Helia University of Applied Sciences, Finland
ari.alamaki@haaga-helia.fi amir.dirin@haaga-helia.fi

JOUNI HUOTARI NIKO KORHONEN
JAMK University of Applied Sciences, Finland
jouni.huotari@jamk.fi niko.korhonen@jamk.fi

Rapidly developing a customizable mobile application and the related software as a service (SaaS) is challenging and rarely studied. Traditionally, SaaS solutions are mainly accessed using personal computers, but the mobile SaaS solutions are needed in the tourism sector, for example, where users are mobile. This paper presents a case study where the original need was to design a customizable mobile tourism guide service for use by several small tourism companies, and to assess its functionality in a field study. The result of applying the Vaadin 6 Java web framework and LAMP technologies was a robust mobile application SaaS prototype system that fulfilled the essential design needs in the eight field test cases. This study shows that the field testing of a mobile concept can be completed easier when using Vaadin Java web framework, as it provides support for cross-platform functionality and GUI design, and completes, for example, LAMP-based SaaS solution. However, results point it out that new digital navigation features were needed to develop or improved and mobile web approach causes some usability challenges especially in the compass based navigation and user tracking. This study provides an example of how to develop a SaaS-based mobile service prototyping environment, which is needed while field testing new B2B mobile services with various groups of stakeholders. Our case study analysis reveal that the Vaadin development environment facilitates the rapid prototyping for digital services in an affordable way. The overall contribution of this paper is predominantly for software engineers and web application developers.

keywords: Rapid development, mobile services, SaaS, software development, Vaadin

1 Introduction

With the increasing popularity of mobile applications and smartphones, there is a unique opportunity to create new mobile web services for our daily activities and business needs in the SaaS (software as a service) business model. Those mobile services in the SaaS business model require the utilizing mobile web technologies, as they cannot be single and stand-alone native mobile applications sold and delivered through the market places of mobile business ecosystems such as AppStore, Google Play and Windows Markets. The successful traditional Internet or web services cannot be necessarily migrated to the successful mobile service [1]. There are, for example, technological challenges such as various screen sizes and device and platform specific requirements in addition to the issues concerning to the behaviour and usage situation of end users. All mobile applications are not stand-alone downloadable native applications, but some of them can be built on the web technologies. Thus, the mobile web technologies are alternatives for the native mobile applications in building mobile software solutions [2-4]. Although both of them have their advantages and disadvantages, they have often different roles. The third main category in selecting development approach for the new mobile application is the

hybrid mobile application. The hybrid application is a cross-platform solution, which still can access to the devices' hardware resources [2,3]. They run content and services inside of the application framework, and are thus more native ones than generic or dedicated web applications which are connected to the Internet via the browsers. Well-known examples of hybrid applications are the Linkedin, Facebook and Twitter mobile applications.

The native mobile applications offer integrated ability to utilize the interfaces and resources of mobile devices' hardware and its features, such as camera, storage, GPS and push notifications [4]. Disadvantages from the viewpoint of rapid prototyping is, that developers need to build separate native applications to support various requirements as the most common platforms and operating systems, iOS, Android and Windows support different programming languages, such as Objective C or Java. Unlike the native applications, the mobile web applications deploy the browser of mobile devices instead of the direct usage of operating system, and the users are not required to download applications for using them. Hence, the development of mobile web applications seems to be easier for basic needs, where end-users are using them with several different mobile devices [3]. This brings some advantages in prototyping new mobile services especially in utilizing a cross-platform development framework in situations where end-users are having various platforms and device models, they are not easily required to download an application for testing purposes and development requires several improvements and releases in the short time period.

The advancement in the new architectural solutions [5] such as cloud computing [6–10] and service-oriented delivery models [11-14] are driving mobile applications towards service-oriented business models. The cloud computing is the key technological enabler for service-oriented delivery and business models, which are categorized as the software as a service (SaaS), platform as a service (PaaS) and infrastructure as a service (IaaS) [15]. The mobile web and hybrid applications can delegate heavy tasks to the cloud services which enlarge their functionalities and enrich user experience [16]. Thus, scalability is a key advancement in the cloud computing as resources can be spread globally over the continents [17]. Furthermore, the advancements in web technologies such as service-oriented technologies and web services [18–21] and web programming enablers such as AJAX [22] and JQUERY [23] have provided a unique opportunity to develop faster prototypes and more robust mobile web applications for various business segments. In addition to the architectures and program languages, there are advanced web software development frameworks such as Vaadin and PhoneGap [24–26], which make mobile web development easier. Moreover, to ease and speed up the mobile development process with end users, various user centred software development principles have also been developed, such as Lean Startup and mLUX [27-28].

This research project applies mobile web technologies to design and develop a customized mobile guide service to be used simultaneously by several SME tourism companies. Therefore, the aim is to design a mobile tourism system that supports most outdoor activities provided by tourism companies [29,30]. In addition to supporting the outdoor activities, the tourism companies would be able to improve customer satisfaction by offering new types of digital guidance and navigation services as added value for their customers. The existing tourism applications in the market mainly provide information and support for city and urban navigation services [31, 32], and there are only few or if any application for outdoor tourism services. The research goal of this study is to design and develop a mobile guidance service for simultaneous use by several small and medium-sized (SME) outdoor tourism companies. Therefore, based on the gathered needs and requirements, this study aims to apply

Vaadin 6 Java web framework and LAMP technologies (Linux, Apache, PHP and MySQL) to design a mobile web service for the outdoor tourism context. Vaadin Java web framework was selected as it offers Vaadin Touchkit mobile add-on which provides "native look-like" user interface themes and GUI elements, thus seemed to fulfil the basic technological requirements.

For fulfilling the research goals, this study aims to experiment how Vaadin would support the rapid prototyping of mobile tourism guide application and is it possible to build a robust SaaS solution combining the Vaadin java web framework, the LAMP-based editor and administration software and the external map data services. To achieve the research goals, the project was divided into the following phases:

1) Architectural design phase: Defining a customizable mobile guide system to serve several tourism companies simultaneously. The mobile guide system consists of a mobile web application, a content editor and administration software.
2) Development phase: Developing the high-fidelity mobile guide system and other related software, e.g. an editor for developing new route content and a tool for administration.
3) Evaluation and assessment phase: Piloting the mobile service in the field to ensure the functionality of the mobile web application and the SaaS architectural solution.

2 Related Research

2.1 Software as a Service

In a SaaS model [33–38], the application is delivered as a service over the Internet; in other words, it can be characterized as the online delivery of software where the software system itself is hosted by the service provider. Thus, the SaaS refers to the services which are hosted and managed in the cloud systems [39]. This allows the customer to start using the software in a matter of minutes with few or no software configurations needed. Although the graphical user interface and the navigation appear similarly for all customers, they can still create their own unique content on the system and configure the software to some extent according to their personal needs. The users do not need to install or update applications or worry about the suitable operating systems or memory capacity in the SaaS model [40].

The concept of multi-tenancy [41,42] is essential in reviewing scalable cloud services. While SaaS applications can be deployed and hosted as separate instances for different users, it can be much more beneficial to host a single, scalable application for multiple users. Scalable applications come with an automatic load-balancing system that recognizes when there is too much strain on the server, in which case it deploys more instances of the application and distributes some of the service requests on the new instances, or removes existing instances during the quieter periods. The data resources are shared, so no matter which instance is serving the customer, he will always be able to access his own data. However, this is only required for large-scale services, and smaller applications can usually manage with only a single instance for each service.

2.2 Mobile Web Technologies

The Internet or web applications are not any longer only simple web sites. For example, Mainetti et al. [43] point it out that *"a web-based common computing platform introduced new technologies that dramatically transformed the traditional web paradigm. These new technologies are the main factor that leads to the definition of a new class of internet applications called rich internet applications (RIAs)"*. In addition, social computing and web 2.0 technologies have modified the traditional understanding of web technologies. The web 2.0 technologies connect people, enables online communication and create the sense of community on the networks without the need of physical interaction [44]. Examples of such applications are social media applications which have significantly impacted on our daily lives and our perspective to the Internet services [45].

However, mobile web applications [46–48] are very similar to the web applications usable through a normal PC's browser. A developer can achieve much with HTML5, CSS and JavaScript when creating new applications, and mobile devices are able to access most of the "traditional" websites without any issues if they are built as the responsive websites. There are, however, some technical problems that the developers need to keep in mind while working on the application, such as the different screen sizes and resolutions of mobile devices. The application might look good on a tablet, but at the same time be almost unusable on a smartphone besides of responsiveness. The interface's layout needs to scale depending on the available screen space, in most cases going as far as completely relocating some elements on the interface.

Aside from functioning properly, the application should also look pleasing to the user's eyes. Native applications have a consistent style and are usually the correct choice, but the problem is that they are only compiled for specific operating systems. Web applications can be run on most devices, but the developers need to spend more time forming a functional and stylish interface. There are style guides and interface frameworks available [49] for making the web applications look like and behave very similarly to native applications. The developers have different choices, ranging from complete custom components and elements to editing CSS files and using the system's own style guidelines.

The web applications on mobile devices usually have very basic functionality, such as button presses, but in order to provide extensive support for touch screens the application needs to understand more complex touch gestures. The application can be compiled into a native web application with tools like PhoneGap, or the support could be added with libraries like jGestures for JavaScript [50], or components such as the Vaadin TouchKit.

2.3 Vaadin Java Web Framework

Vaadin [24, 25] is an open-source rich Internet application Java framework consisting of a server-side programming model and client-side development tools based on Google Web Toolkit and HTML5. It utilizes server-side architecture, where the majority of the software's logic is run on the server. One of Vaadin's key features is the easy-to-make user interface, which in most cases does not require any adjustments to integrate with the website style sheets, although the developer is free to modify the style sheets and even create new themes if he so desires. It can also be used to create mobile applications using Vaadin TouchKit, which makes it possible to capture touchscreen events and have the application understand touch gestures.

Vaadin heavily utilizes Google Web Toolkit, which is used to convert the Java code to JavaScript, which is then rendered on the website. Additionally, the browser side uses Ajax technology to transfer data asynchronously and reload only parts of the pages instead of whole pages, thus improving the interactivity and user experience. The framework is based on event-based programming and provides widgets and add-ons to help developers design and build richer web applications and solutions more quickly.

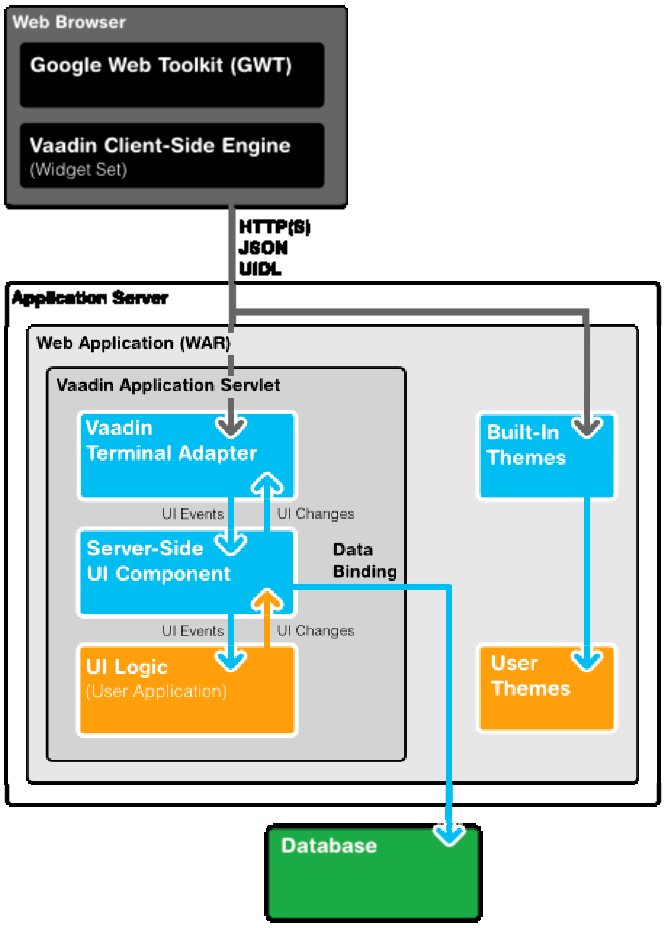

Figure 1 The Vaadin 6 architecture [25].

The Vaadin framework itself is free to use in any project, as are most components available in Vaadin's Add-on Directory; however, to use the more advanced tools, such as Vaadin TouchKit or Vaadin TestBench, the developer must subscribe to Pro Tools, which requires a monthly fee.

3 Research methodology

The aim of this study was to analyze qualitatively the Vaadin 6-based mobile web service and its functioning in the SaaS architecture, and to determine whether it was a viable technological solution during the proof-of-concept phase.

To achieve this, the research approach involved a qualitative case study of the mobile tourism service of SMEs. Data collection was accomplished by organizing practical field test cases, where the mobile SaaS system was used in eight test cases in the outdoor environment. The technological functioning was evaluated by users who tested and evaluated the mobile service in the real usage environment. Feedback was collected during semi-structured interviews, from the feedback function of the mobile application and/or focus group discussions. This study focuses on the description and evaluation of development experiences using Vaadin 6 and related web technologies through the action research method. The qualitative case study was selected because the research goal was to experience both the Vaadin framework and the mobile SaaS solution. In addition to the eight filed tests, the developers created nine new instances for testing the technical functionality of the SaaS system prototype.

4 System Development

4.1 Application Concept

During the elicitation phase, we conducted a short survey to investigate the possibilities and the possible benefits of mobile applications for small tourism companies. The survey mainly focused on revealing the possibilities of mobile services in this sector, what was expected from them and the tourism companies' business needs from mobile-based services. From the analysis of the survey data, we identified two main requirements:

1. Location-based services for outdoor activities, such as route information, location guides, maps etc.
2. The ability for the companies themselves to maintain and update the content of the mobile services.

This was valuable information for our designers in proposing the first concept to these companies. As the designers had prior experience of outdoor activities such as kayaking, camping and hiking, coming up with additional features was significantly easier. The additional features were proposed to the companies, and their feedback was collected and analyzed. As soon as the confirmations for the proposed features were received, we investigated the technological feasibility of the features with the project programmers and architects. Thus, the product concept of a mobile web application that guides and informs kayakers and those participating in other outdoor pursuits was designed. In addition, the concept fulfils the companies' needs by enabling them to maintain and update the application content themselves. As a result, each company can provide their own supported content to their customers.

4.2 Requirement Specification

The next phase of the development process after accepting the proposed concept was to produce a list of the product requirements. The product targets and their main use cases were as follows:

- Mobile application users, e.g. kayakers, paddlers and hikers, should be able to use the app during outdoor activities.

- The tourism companies' content producers should be able to create, update or remove route points of interest and other related data.
- The system developers should be able to manage the SaaS system and its instances.

The three main functional requirements were resolved based on the survey in the elicitation phase: 1) the users must see their current location and direction on the kayaking route; 2) the users must see the landing and camping places on the kayaking routes; and 3) the tourism company must easily maintain and update the content of the mobile application, e.g. by using a PC application. In addition to the three main requirements, several optional requirements were also defined: 1) the application provides hints and additional information to the users; 2) the application contains a link to the tourism company's website; 3) the application shows the location of other kayakers if they allow it; 4) the application saves the kayaking route information on the map, and the kayakers can later see their routes; and 5) the application saves the kayaking routes and the tourism company can see the summary of all saved routes.

4.3 System Architecture

The mobile guide system architecture consists of three main parts: 1) the mobile application; 2) the content editor; and 3) the system administration. The end users can use the mobile application through their mobile devices' browsers. The tourism companies can create, update and maintain their own company's specific content using the content editor software, and the service providers can manage different software instances using the system administration software. Figure 2 presents the general elements of the system architecture.

The digital maps are physically located on the web map servers (WMS). Open Street Map was used in the first trailed concept, which did not have support for qualitative geographical maps of the rural areas. Consequently, the developers decided to use the Finnish map services Karttakeskus and Maanmittauslaitos, which provide exact geographical digital maps that include details such as contours, paths and small streams. The utilized maps' coordination forms were WGS84 (the World Geodetic System) and KKJ (the Finnish coordinate system).

The mobile application was built on the Vaadin Rich Internet Application framework for Java [16, 17], which employs server-side programming logic. The back-end components were built on an open source LAMP stack (Linux, Apache, MySQL, PHP), given that this is one of the fastest ways for developers to start building a functional system. The management software for the mobile guide service was built on open source web technologies such as HMTL, JavaScript and PHP programming.

Mobile web technology was selected as the architectural principle. Most of the existing mobile applications in the tourism sector are, however, provided as standalone native applications via marketplaces, in which the ready-maps, routes and content are embedded in the application. The tourism companies have virtually no means of later managing the content, as no content management software is available or integrated into the native application.

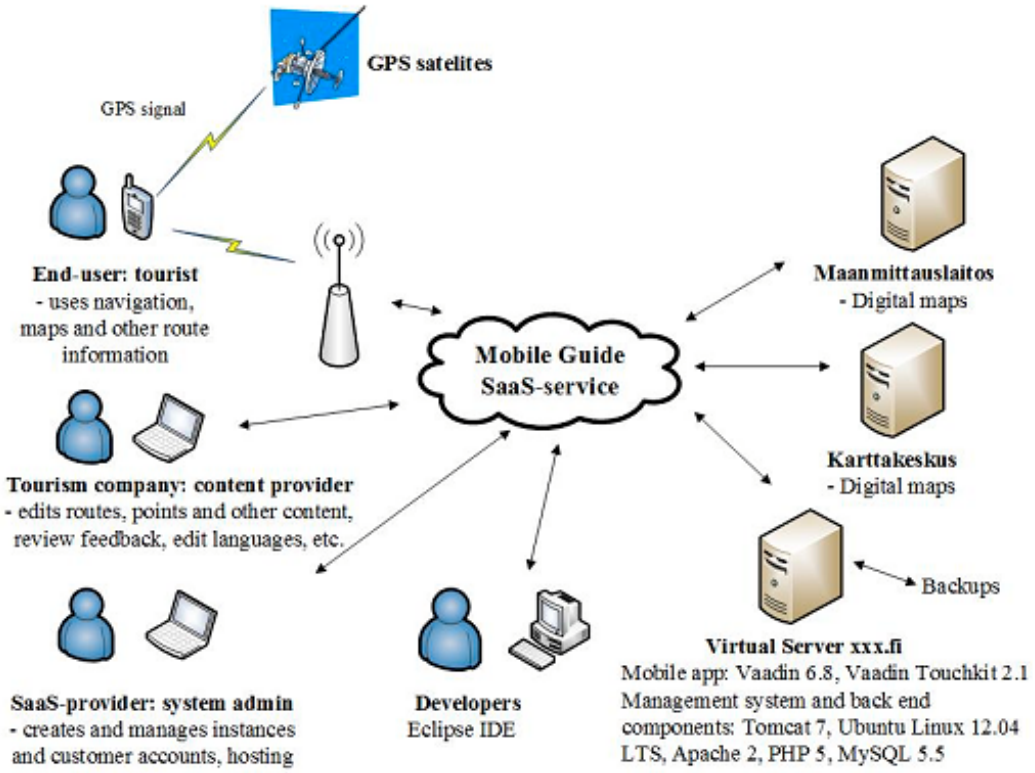

Figure 2 Mobile guide system architecture.

4.4 Software Development Process

The mobile web application

The mobile web application was developed using the Vaadin 6 Java web framework and the Vaadin TouchKit add-on for mobile services. The Vaadin framework was selected for the development environment of the mobile application as it provides a client-server web development framework and support for responsive mobile web development. Figure 2 presents the screenshots of the tourism guide mobile application.

The designed mobile application deploys a browser for rendering the content to the user's device. To begin using the application, the user types the URL address into the browser of their mobile device, scans the QR code or clicks the application icon on the home screen of their device. The system opens the application in its own browser window without the browser navigation bars; resultantly, the application looks like a native app on the mobile device, as shown in Figure 2.

Figure 3 Example screenshots of the tourism guide mobile application build on the Vaadin 6 Java web framework

The key features of the application are the maps and routes, which are represented as buttons on the application main page. The main page also includes "info" and "company" buttons. The info button provides topical information and the company button is linked to the tourism company's website. The map button on the main page shows the digital map, including points of interest, positioned to the user's location. The route button provides a page that gives a list of available routes, and also includes settings allowing users to hide or show routes from the list. The hidden routes are not shown on the navigation page. The navigation page displays the map with a digital compass, route information and distance, speed and time information related to the selected route. Those pages utilize technical interfaces to the GPS and digital compass of the user's mobile device.

The content editor

In addition to the mobile web application, the development team also created content editor software that allows the tourism companies to create and manage routes, points of interest and other information in the mobile guide system. The editor software is a simplified content management system built for the needs of this mobile guide service, and consists of the following features:

1. Point-of-interest management
2. Route management
3. Category management
4. Language localization
5. Feedback results monitoring
6. Info page management

The point-of-interest management provides a feature to add interesting locations to the map, such as camping places, cafeterias, campfire sites, landing points, beautiful views or potentially dangerous places. The feature includes a list of icons that allows users to select the most descriptive for each point of interest on the digital map. The digital map is accessible using the map button on the main page of the mobile application, while the route management provides a feature to draw a new route on the digital map. The category management allows for adding or removing new points of interest, and the language localization provides a feature to add a new language version or change the language used. The feedback result monitoring allows access to the numerical and textual feedback provided by users using the feedback function on the main page. Finally, the info page management allows for changing the link and text behind the info and company buttons on the main page.

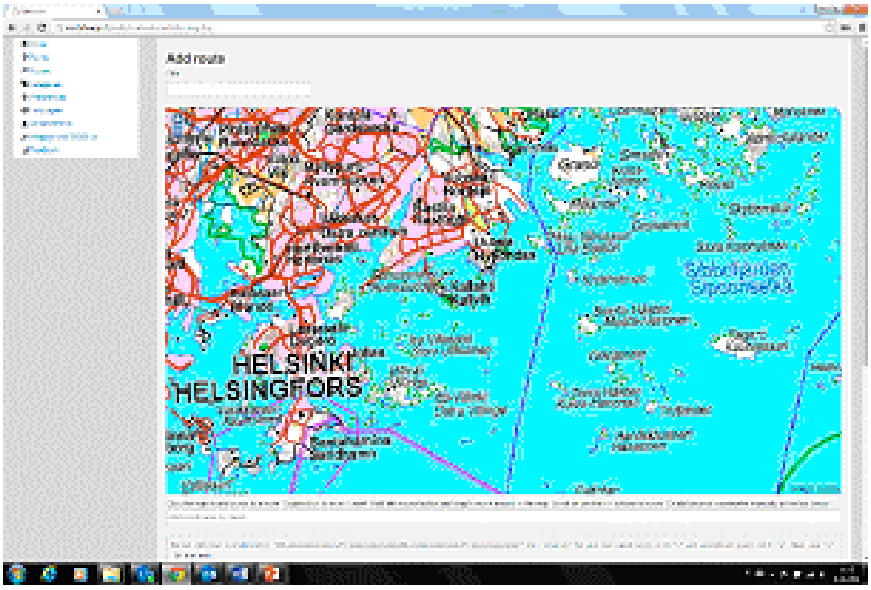

Figure 4 An example UI of route manager in the content editor.

The content created and added to the system was displayed on the mobile application's browser while navigating to the test case's URL. Although the first prototype of the content editor had some usability issues, the small tourism companies were still able to use the editor software to create their

own route content for mobile services. The ability to manage the content available to potential customers was viewed as a positive feature because these companies are often not financially secure enough to outsource their content management.

The prototype's field tests demonstrate that the proposed editor concept fulfils the companies' essential needs. The editor software is also an essential component for creating software as a service concept, where small tourism companies can register with the service, receive their own URL and web environment and start to build their own company-specific content into the mobile web service.

The system administration

The system administration software was designed and developed to help the management of several mobile guide service instances running on the same virtual server or on the public cloud infrastructure. It was essential to design, as the primary goal of the project was to build a mobile guide service for several small tourism companies to use simultaneously, as in the multi-tenant principle.

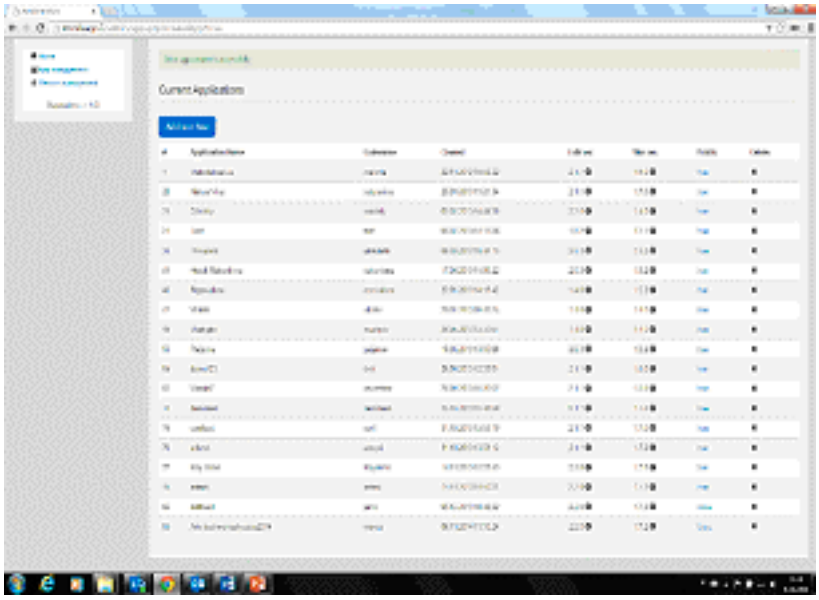

Figure 5 An example UI of the customer instance manager in the SaaS system.

If the system admin software had not been created, the development team would have been required to manually create, update or remove mobile web service instances for each new and existing company using the service. It consists of application management, authentication and version management features. The application management can create new instances for new companies, list, enable or disable instances and send a new URL, with login and training information, in an email to the new customers. The version management function allows for updating the current version and removing the customer URLs more easily, and it can also show a list of all user companies. The admin

software is for system administrators only, and is not available to the tourism companies or their customers.

An additional web form was added to the mobile guide service in the later phase and integrated with the admin system. The form allows companies to receive a dedicated URL and a mobile web service environment for their own use. When a new instance is created, the admin system automatically boots the web server application, and the new mobile guide service is in use and accessible on the Internet. The development team largely created new instances and emailed the URLs to the new pilot companies or other testers during the pilot tests using this admin system because the main focus of the pilots was the testing of the mobile application performance and its user experiences. Therefore, the automated admin system was not tested during the field tests, although technical functionality is already available and implemented.

4.5 Automated Installer to Generate New Apps

The developers programmed an installer that generates new application instances on the web server, and included it as a part of the SaaS admin software. This solution represents a lower-level SaaS architectural maturity [51] than systems with a higher SaaS maturity level that run single instances able to serve every customer with configurable metadata. In this case, however, a higher SaaS maturity level was not needed since the goal of the study was to assess the mobile guide service and evaluate the SaaS concept.

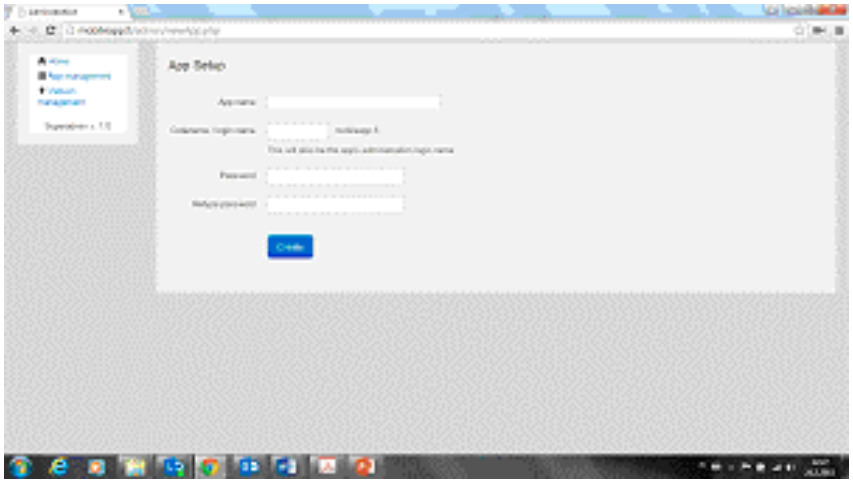

Figure 6 An example UI of the automated installer for generating new mobile tourism applications and content editor access for the use of potential new companies.

In addition to launching new instances via the admin system, the developers also later designed a web form into the project's test website, which made it possible for any potential customer to generate a new mobile application instance by completing the "free trial" web form. After submitting the necessary information into the form, such as the name of the application, a code name and a password,

the installer automatically created a new mobile guide application and granted access to the content editor software.

Whenever the installer receives a new form containing the required information it processes it according to the algorithm planned to manage the new application creation process. The installer compares the given passwords and checks information from the app form. If this is valid, the process continues to consider the database transactions, such as load settings, write the application information into the database and write Apache RewriteRule for the new URL. The program uninstalls the default content editor files and modifies the data in Apache's htaccess. Next, the program creates the settings file and the database connections, and runs a script that creates some example content and a new user into the database. After this, the program reconfigures the web server, uninstalls the mobile application codes and creates a settings file with the application-specific database information. Finally, it directs the browser to an address that displays the status message for the new customer or admin who created the new application instance.

Table 1 Test cases and evaluators.

Instance no	**Test case**	**Evaluators**
1	Field test A	End users (n=6 kayakers)
1	Field test B	End users (n=17 hikers)
1	Field test C	End users (n=6 hikers)
2	Field test D	End users (n=15 kayakers)
3	Field test E	End users (n=10 paddlers)
4	Field test F	End users (n=2 hikers)
5	Field test G	End users (n=25 kayakers)
6	Field test H	End users (n=5 kayakers)
7–17	Technical test (new URL instance created)	Developer (n=1 walking on the street or in the forest)

5 Testing the Mobile SaaS System

Seventeen instances of the mobile service were created; six were dedicated for separate tourism organizations and eleven were test instances. The design team created new instances using the SaaS

admin software, and the initial routes and other content was added to the new company- or test-specific mobile services. An instance is a mobile application service with its own URL address running in the SaaS system. Each instance has its own back-end software and database generated by the automated installer in the SaaS system.

The design team arranged eight test cases after the SaaS application concept and functional prototype were deemed ready for performance and usability testing. The goal of the test cases and company pilots was to assess the functionality of the mobile service with users and companies who did not have prior experience of the concept. The selection criteria for the test cases was that they had to present potential end users or customer companies with business needs that could be met by the system. The evaluators used the application while kayaking or hiking. In the field tests A, C, D and G, some users used the application in pairs while hiking, or else it was demonstrated to the users at the tourism centre. The end user experience of all evaluators, and the validation of design iterations from the viewpoint of mobile application design and use, is described in another research article by the authors [52, 53].

The mobile web application and its instances were technologically successful in the test cases as the users were able to use the application. For example, a user wrote the following feedback: *"the application basically works, but minor shortages disturb the usage".* The feedback shows that the mobile web application concept met the expectations of end users while kayaking, paddling or hiking. The concept includes the basic features needed while participating in the outdoor activities, such as points of interest on the map, routes complete with speed, distance and navigation information, tourism service provider information and instructions. Although the basic features worked in the application and the SaaS system provided a stabile service during the tests, several recommendations for usability improvements were raised. The following short quote shows a typical example of feedback which was concerned to the usability issues: "*noticed in the nature… …change the narrow arrow to point directly to the destination. These would work as an info that users would understand better, especially when destination and user* [location] *do not fit to the same screen".* Most development recommendations or test feedback were concerned to the features which are not Vaadin-specific. The following example shows how a developer comments the need to design new features to the Vaadin framework: *"Vaadin 6 did not provide compass things, also support for GPS was weak. Thus, we had to build Vaadin add-on which supported better accuracy GPS navigation".* The developers ended up to the designing and developing a compass and its GUI element, as the Internet connection was not stable on the sea, and it was hard to read a small map in the kayak. The compass with GPS navigation also ensures that the continuous Internet connection is not needed while navigating. While testing the application, we found that the digital compass was not still stable in some devices, such as in iPhone, and it pointed tens of degrees to the wrong directions. However, Sandle [54] has also found accuracy errors in his study related to the iPhone compass. Transition of location information from one route point to another route point did not always work and the compass elements vibrated on the screen. Those errors were found in Android devices, for example. The screen savers also disturbed testing in some cases, as they shut the browser off. However, the content is also important to the users although navigation functions should also be stable and reliable. For example, a user stated that *"content is important, especially its quality. The approach that users are content creators* [like in social media applications] *does not work in this kind of guidance applications".* The field tests show that content related features, such as route, map and interest point information, worked fine and their usability met expectations of users,

The analysis of eight different test cases shows that the architectural solution fulfilled the requirements of piloting the service, and made it possible to collect end-user experience during field tests. Applying Vaadin 6 and Vaadin TouchKit mobile add-on to the outdoor navigation needs shortened the design and development process by providing cross-platform support, design themes, user interface elements and server-side logic support. It also offered direct functionality for Android- and iOS-platforms, which decreased the need for device testing. While reflecting the development experiences, the developers stated that, for example; *"it is hard to say which one would have been faster to develop, native app or browser based. Now it, however, works directly with iPhone and Android, as it deploys the browser. Nevertheless, you need to take into account in development, what device models there are as the compass is depend on it, for example"*. In the first requirement specifications, the saving of paddled or hiked route and the locations of users were included to the development plan, but they needed to be rejected. This requirement is possible to realize in the mobile web or hybrid applications which are sending user information to the server, and whose data connection is stable. However, during the field tests, data connection was sometimes low or it cut off and that fell the application in the browser. The stability of browser and data connection are important in the web application as the application deploys the browser and data connection.

The programming of Vaadin-based system did not require high or special technological competences from the programmers, as all four developers involved in this project were students of higher education. Two students involved to the programming of Vaadin application and they had no prior experiences from the Vaadin [55]. Only basic competence in Java programming and software development was needed in starting this project. The studying of Vaadin-based Vornitologist-project helped to get inside of Vaadin programming logics and architecture. The concept definition and requirement specification was mainly conducted by developers and project management in design iteration with stakeholders [52, 53]. Vaadin provides suitable instructions and a user forum for using their Java framework, which is an important requirement in selecting new frameworks.

This mobile service development experiment shows that small tourism companies would have use for and benefit from a customizable mobile guide system delivered as a SaaS solution. The system developed in this project is only a proof-of-concept, and does not fulfil all requirements of multi-tenancy, which is a requirement for scalable cloud services. However, the SaaS system fulfilled the needs of the service provider who needs to be able to manage several instances and customer accounts. The implemented system based on the SaaS business model provides a mobile guide for tourism companies as a service without the need for hosting, maintenance or installation on premises. From the SaaS maturity viewpoint, it does not represent a fully scalable service with balance loaders or shared databases and their schemas, but in the business-to-business context the architectural solution is sufficient, especially in the start-up phase of commercialization where the number of customer companies is significantly lower than in business-to-consumer solutions. The mobile guide system resources are shared with new companies by copying and duplicating the default instance inside the SaaS software and server, whereas in shared models tenants are separated by IDs in the same database.

6 Discussion

The Vaadin 6 Java web framework and Vaadin TouchKit mobile add-on [25] assisted in this project by making it possible to build the mobile guidance application more quickly, which allowed us to start collecting development feedback from tourism companies and end users at an early stage. It provided

ready-made user interface elements and system components for the programmers to implement without any need for unnecessary adjustments. Vaadin also provided direct code-level support for Android- and iOS-based devices, which saved significant time in the design and development work. By using a mobile programming framework, the design team was able to test features that require a functioning prototype. For example, features based on geolocation technologies, such as GPS and a digital compass, are extremely difficult to test in any other way than by building a functional prototype as this study shows. The findings of this study support the results of the web framework evaluation realized by Maple [56], where they also value Vaadin's prototyping features and comprehensive support of GUI elements and design themes.

This study shows that not all development ideas and mobile service concepts can be tested with screenshots or mock-ups, as there are features that require building a functional prototype with the real functioning program code. However, gathering relevant data from user experience tests requires an error-free technological solution that fulfils the essential user needs, but this study focused on the testing of mobile SaaS concept and Vaadin's relevance for tourism application development. This study also points out that the mobile web approach has some advantages compared to native mobile application development, especially in the prototyping phases. For example, the native mobile application should be uploaded to a marketplace, which becomes more expensive or time consuming for smaller companies with limited technological and financial resources. A native application, such as an iOS application, also needs to pass a certification process in order to get to the marketplace. Unlike native mobile applications, mobile web solutions can be delivered, purchased and hosted on a public Internet site instead of commercial marketplaces. This makes it easier to offer mobile services for small tourism companies as a SaaS model that allows for a multi-tenant architectural solution. This solution allows for serving several companies simultaneously from a single mobile web service system. Moreover, the developers can easily test the service themselves and continually build and update content to the offered service without packing new versions to the marketplaces for downloading. This approach creates a unique and quick means of prototyping and piloting mobile services.

At the beginning of the project, it was identified that the mobile web approach presents some challenges, especially in the tourism sector where the end users use the services in the natural environment or in rural areas where data services and battery consumption may become critical. However, the main goal of the project was to test both the application and system concept, ignoring the mobile web approach challenges. Since mobile web technology allows for developing and testing the proposed concept more quickly thanks to the end users' involvement and prototype testing, the application development was not based on the technological robustness of the system; instead, the faster development process and field testing were considered the highest priority in this study.

7 Conclusions

When developing mobile application prototypes, choosing a mobile web application instead of a native application is a completely valid solution. This study shows that the building of a tourism-based mobile application does not require such platform-specific features, which often still require the use of a native application to ensure a smooth user experience. The digital compass and GPS interfaces worked well in the tests, although the developers improved a method included in the Vaadin GPS feature. Web technologies, such as the Vaadin Framework and Vaadin TouchKit mobile add-on, can

be used in quickly putting together prototypes that can be used and tested on multiple platforms simultaneously, while native applications would have to be created and tested across all platforms separately. Testing mobile web applications is also much simpler considering that the only tool needed is the mobile device's browser. In short, the development and testing of mobile applications can be achieved far faster when choosing to use mobile web technologies that have the required components included and do not cause any extra problems with the interface, of which the Vaadin framework is a good example.

Vaadin has released Vaadin 7 since this study was completed, and it offers an even more comprehensive set of UI add-ons and themes, improved usability and a developed visual editor for easing UI development [57]. Hence, productivity in the software development process will evidently be better than in the previous version and the new framework provides an improved user experience.

This study shows that the functioning beta-level service built on the Vaadin 6 framework and Vaadin TouchKit mobile add-on provided a solid solution for testing the concept of a mobile tourism guide service in eight field test cases in the real outdoor environment. An advantage of using the Vaadin-type framework for prototyping is that the developers can easily continue development from prototype and beta-versions to commercial and production-level versions, which shortens time-to-market in the commercial mobile service development. Therefore, Vaadin also suits web application prototyping as it provides comprehensive support and themes for UI design, as was also shown by Maple [56]. In addition, this study provides an example of how to rapidly develop a SaaS-based service by integrating the Vaadin-based mobile web service with the LAMP-based content editor, administration and installer software. As for the system developed with the chosen technologies, SME tourism companies can add and edit route information on the mobile service and launch their own mobile application instances for their customers' use. When commercialized and deployed properly, small companies could use the system to create their own mobile services without the need for extensive programming skills.

8 Future Works

Mobile services are increasingly delivered as cloud-based services by utilizing a SaaS business model. The mobile web technologies and their frameworks provide a faster way to prototype and pilot them with end users than building functioning native applications. Business-to-business (B2B) mobile services require a SaaS business model, including cloud-based hosting and delivery, as SMEs demand the possibility of dynamically maintaining their service content.

More research is needed on how to decrease the lead-time of development processes in the user-centred design of mobile SaaS systems. This calls for research on and the development of new frameworks that provide support for the multi-tenancy architecture of SaaS systems. In designing and developing cloud-based solutions for B2B business segments in the SaaS business model, the system needs to serve thousands of customers simultaneously in aspects ranging from providing a smooth user experience to security.

Acknowledgements

This work was supported by the DIGILE's Digital Services research program, funded by TEKES. We also wish to thank partner companies for their helpful contributions and support in this project.

References

1. Shuiqing Yang, S., Wang, Y. and Wei, J. Integration and consistency between web and mobile services, Industrial Management & Data Systems, Vol. 114 Iss: 8, pp.1246 – 1269, 2014
2. Technopedia. Native mobile app. https://www.techopedia.com/definition/27568/native-mobile-app [accessed 5.11.2015] 2015
3. Saccomani, P. Native, Web or Hyprid Apps? What's the difference. http://www.mobiloud.com/blog/2012/06/native-web-or-hybrid-apps/ [accessed 5.11.2015] 2012
4. Charland, A. and Leroux B. Mobile application development: Web vs. native. Journal of Communications of the ACM. Volume: 54 2011
5. Gomaa, H. Software modeling and design: UML, use Cases, patterns, and software architectures. New York: Cambridge University Press. 2011
6. Carlin, S., and Curran, K. Cloud Computing Technologies. International Journal of Cloud Computing and Services Science, Vol 1 (2): 59–65, 2012
7. Tahir, A.O and Adeyinka, A. Autonomic Service Management in Mobile Cloud Infrastructures. In International Journal of New Computer Architectures and their Applications, 4(2): 91-107, 2014
8. Kesavan, S. and Jayakumar, J. Network Performance Analysis of Cloud Based Multimedia Streaming Service. International Journal of New Computer Architectures and their Applications, 4(3): 156-166, 2014
9. Makoto S., Masami M., Hiroshi D., Yoshiaki Y. Ciphertext Diverge-Merge Scheme of Identity-Based Encryption for Cloud-Based File Transmission Service. International Journal of Digital Information and Wireless Communications 5(1): 52-59, 2014
10. Khanghahi, N., Nasiri R and Razavi M. A New Approach Towards Integrated Cloud Computing Architecture. International Journal of Digital Information and Wireless Communications 4(1): 24-34, 2014
11. Benlian A., Hess T. and Buxmann P.. Drivers of SaaS-Adoption – An Empirical Study of Different Application Types. In Business & Information Systems Engineering 5, 2009
12. Turner, M. and Budgen, D. and Brereton, P. (2003) 'Turning software into a service, *Computer*, 36 (10): 38-44, 2003
13. Low, C., Chen, Y. and Wu, M. Understanding the determinants of cloud computing adoption, Industrial Management & Data Systems, Vol. 111 Iss: 7, pp.1006 – 1023, 2011
14. Scale, M. "Cloud computing and collaboration", Library Hi Tech News, Vol. 26 Iss: 9, pp.10 – 13, 2009
15. Mell, P. and Grance, T. The NIST definition of cloud computing. Recommendations of the National Institute of Standards and Technology. U.S. Department of Commerce, 2011

16. Flores, H., Srirama, S. and Paniagua, C. Towards mobile cloud applications: Offloading resource intensive tasks to hybrid clouds, International Journal of Pervasive Computing and Communications, Vol. 8 Iss: 4, pp. 344 – 367, 2012
17. Hayes, B. Cloud Computing. Communications of the ACM – Web Science, Vol 51:7, pages 9.11, 2008
18. Liu, Z., Le Calvé, A., Cretton, F. and Glassey, N. Using Semantic Web Technologies in Heterogeneous Distributed Database System: A Case Study for Managing Energy Data on Mobile Devices. International Journal of New Computer Architectures and their Applications (IJNCAA) 4(2): 56-69, 2014
19. Laplante, P., A., Zhang, J., and Voas, J. What's in a name? Distinguishing between SaaS and SOA. In IT Professionals vol 10, 46-50, 2008
20. Josuttis, N. M. SOA in Practice. October (Vol. 253, p. 352), 2007
21. Snell, J., Tidwell, D., and Kulchenko, P. Programming Web Services with SOAP. Language (p. xiii + 244), 2002
22. Zakas, N., and Fawcett, J. Professional Ajax. Professional Ajax, 1–380, 2007
23. The jQuery Project. jQuery: The Write Less, Do More, JavaScript Library, 2011
24. Vaadin. Introducing Vaadin += GWT, Web Application Scalability□: Vaadin goes Movies. Dock: World of Web Applications, 1–52, 2010
25. Gröönroos, M. Book of Vaadin: Vaadin 6.4 https://vaadin.com/download [accessed 17.2.2015] 2010
26. PhoneGap. http://phonegap.com/community/ [accessed 17.2.2015] 2015
27. Ries E. The Lean Startup. How Constant Innovation Creates Radically Successful Businesses. London: Penguin Books, 2010
28. Dirin, A. and Nieminen, M. mLUX usability and User Experience Development Framework for m-learning, International Journal of Interactive Mobile Technologies (iJIM), 2015
29. Gavalas, D. and Kenteris, M. A web-based pervasive recommendation system for mobile tourist guides. Personal and Ubiquitous Computing 15:759–770, 2011
30. Kennedy-Eden, H. and Gretzel, U. (2012) A taxonomy of mobile applications in tourism. E-review of Tourism Research, 10 (2), 47-50, 2012
31. Gavalas, D., Konstantopoulos, C., Mastakas, K. and Pantziou, G. Mobile recommender systems in tourism. Journal of Network and Computer Applications 39, 319–333, 2014
32. Schwinger, W., Grün, Ch. Ch. Pröll, Retschitzegger, W. and Schauerhuber, A. Context-awareness in Mobile Tourism Guides – A Comprehensive Survey. Handbook of Research on Mobile Multimedia, Second Edition, Vol 2, 298-314, 2007
33. Benlian A., Hess T. and Buxmann P. Drivers of SaaS-Adoption – An Empirical Study of Different Application Types. In Business & Information Systems Engineering 5, 2009
34. Kesavan, S. and Jayakumar, J. Network Performance Analysis of Cloud Based Multimedia Streaming Service. International Journal of New Computer Architectures and their Applications, 4(3): 156-166, 2014
35. Turner, M. and Budgen, D. and Brereton, P. Turning software into a service, Computer, 36 (10): 38-44, 2003

36. Laplante, P., A., Zhang, J., and Voas, J. What's in a name? Distinguishing between SaaS and SOA. In IT Professionals vol 10, 46-50, 2008

37. Dubey, A. and Wagle D. Delivering software as a service. The McKinsey Quarterly. May (2007) 27

38. Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Konwinski, A., Lee, G., Rabkin, A., Stoica, I. and Zaharia, M., A view of cloud computing, Communication of the ACM, Vol. 53 No. 4, pp. 50-8, 2010

39. Ojala, A. and Tyrväinen, P. 2011 Value networks in cloud computing, Journal of Business Strategy, Vol. 32 Iss: 6, 40 – 49, 2011

40. Armbrust, M., etc. Above the Clouds: A Berkeley View of Cloud Computing. Electrical Engineering and Computer Sciences University of California at Berkeley. Technical Report No. UCB/EECS-2009-28, 2009

41. Schroeter J., Cech S., Götz S., Wilke C. and Aßmann U. Towards Modeling a Variable Architecture for Multi-Tenant SaaS-Applications. In Proceedings of the Sixth International Workshop on Variability Modeling of Software-Intensive Systems - VaMoS '12, 2012

42. Chong, F., Carraro, G. and Wolter, R. Multi-Tenant Data Architecture. http://msdn.microsoft.com/en-us/library/aa479086.aspx Microsoft Corp. [Accessed 30.01.2015] 2006

43. Mainetti, L., Paiano, R., Bolchini, D. and Pandurino, A. Dialogue□based modeling of rich internet applications: the Rich□IDM approach, International Journal of Web Information Systems, Vol. 8 Iss: 2, pp.157 – 180, 2012

44. Wood, B.A., Using Web 2.0 technologies for communication, collaboration and community building: a Caribbean perspective, Library Hi Tech News, Vol. 30 Iss: 6, 7 – 11, 2013

45. Westerman, G., Bonnet, D. and McAfee, A. Leading Digital: Turning Technology into Business Transformation. Boston: Harvard Business Review Press. 2014

46. Liu, Z., Le Calvé, A., Cretton, F. and Glassey, N. Using Semantic Web Technologies in Heterogeneous Distributed Database System: A Case Study for Managing Energy Data on Mobile Devices. International Journal of New Computer Architectures and their Applications (IJNCAA) 4(2): 56-69, 2014

47. Tahir, A.O and Adeyinka, A. Autonomic Service Management in Mobile Cloud Infrastructures. In International Journal of New Computer Architectures and their Applications, 4(2): 91-107, 2014

48. Rudolph, P. Hybrid Mobile Apps: Providing A Native Experience With Web Technologies. http://www.smashingmagazine.com/2014/10/21/providing-a-native-experience-with-web-technologies/ [Accessed 28.5.2015] 2014

49. Liew, K. Queness Web Design & development Blog: 11 Multi-touch and Touch events Javascript libraries http://www.queness.com/post/11755/11-multi-touch-and-touch-events-javascript-libraries [Accessed 1.6.2015] 2012

50. CodePlex. Project Hosting for Open Source Software. jGestures: a jQuery plugin for gesture events. https://jgestures.codeplex.com/ [accessed 06.11.2015] 2015

51. Chong F. and Carraro G. Architecture Strategies for Catching the Long Tail. Microsoft Corporation.https://msdn.microsoft.com/en-us/library/aa479069.aspx#docume_topic5 [accessed 6.3.2015] 2006

52. Alamäki, A. and Dirin, A. Designing mobile guide service for small tourism companies using user centered design principle. The International Conference of Computer Science, Computer Engineering and Social Media, Thessaloniki, Greece. December 12-14, 2014

53. Alamäki A. and Dirin A. The Stakeholders of a User-Centred Design Process in Mobile Service Development. International Journal of Digital Information and Wireless Communications 5(4): 270-284, 2015

54. Sande, S. Get lost! iPhone compass app struggles in tests. http://www.engadget.com/2013/10/18/get-lost-iphone-compass-app-struggles-in-tests/ [accessed 5.11.2015] 2013

55. Valta, V. Karttapohjainen mobiilisovellus matkailualan yrityksille. [Map-based mobile application for outdoor business, based on the Vaadin framework] Bachelor Thesis, Haaga-Helia University of Applied Sciences. 2013

56. Maple, S., etc. The 2014 Decision Maker's Guide to Java Web Frameworks. http://zeroturnaround.com/rebellabs/the-2014-decision-makers-guide-to-java-web-frameworks/ [accessed 11.10.2015] 2014

57. Grönroos, M. The Book of Vaadin - Vaadin 7 Edition. Volume 1 and 2. https://vaadin.com/book [accessed 10.10.2015] 2015