

WEB APPLICATION DEVELOPMENT UTILIZING CLOUD VIRTUAL MACHINE

Olli Muukka

Master's thesis
May 2014
Degree Programme
Information Technology

TAMPEREEN AMMATTIKORKEAKOULU
Tampere University of Applied Sciences

ABSTRACT

Tampereen Ammattikorkeakoulu
Tampere University of Applied Sciences
Degree programme
Information Technology

OLLI MUUKKA:
Web Application Development Utilizing Cloud Virtual Machine

Master's thesis 43 pages, appendices 3 pages
May 2014

The thesis goes through a development project where a web application was implemented to support the start-up company business operations. The main reason to implement a web application was the company needed a system where business data is centrally managed with cost-efficient, simple and easy tool. The deployed cloud service provided a platform for the web application. The alternative to the web application development was to deploy commercial customer relationship management tool, but the company had no will to invest for commercial tool.

The development project first application was to implement a tool to replace the task management on Excel-sheet. The idea of the tool was to have a simple, self-guiding and easy user interface in internet browser, which would provide task management operations and the tasks would be stored in cloud data storage. In principle, the application would enable round-a-clock data availability and accessibility when utilizing the cloud service, and task manageability between simultaneous users when utilizing database. The Excel-sheet task management was mostly lacking the previously mentioned qualities.

The development project contained mainly system design, programming and cloud service deployment tasks. The most popular programming languages and well-known programming techniques were used for implementing user interface on a web browser. The Virtual machine was deployed from cloud with the Linux operating system and with web application software, like web server and database.

The development project needed widely knowledge and skills of the system design, programming and cloud services deployment. Quite much work were spent on system design work and it helped greatly the project implementation to build the web application for the need, and to establish a web application platform for further development. The next application to be developed may be the inventory management for company equipment.

CONTENTS

1	INTRODUCTION.....	6
2	BACKGROUND.....	7
2.1	Main requirements	7
2.2	Technologies.....	8
2.2.1	Virtual machine	8
2.2.2	Operating system	8
2.2.3	Database	9
2.2.4	Server code.....	9
2.2.5	Web server.....	10
2.2.6	The device	10
2.2.7	User interface code	10
2.3	Programming languages	11
2.4	Software tools	11
2.4.1	LibreOffice	12
2.4.2	ProjeQtOr	12
2.4.3	MySQL Workbench.....	12
2.4.4	phpMyAdmin	12
2.4.5	NetBeans IDE.....	13
3	DESIGN.....	14
3.1	Project plan	14
3.1.1	Implementation plan	14
3.1.2	Test plan	15
3.1.3	Maintenance plan.....	16
3.1.4	Project risks	16
3.2	Architecture	17
3.3	Stakeholders.....	17
3.3.1	The Onion model.....	18
3.3.2	The list of web application stakeholders.....	19
3.4	Use cases	21
3.4.1	Use case diagram example	23
3.4.2	Use case specification example	23
3.5	Requirements	24
3.6	Web page design	25
3.7	Information security	26
3.7.1	Threat analysis.....	27
3.7.2	Security features	28
4	IMPLEMENTATION	30
5	TESTING AND VERIFICATION.....	31

5.1 Main requirements	31
5.2 Use cases	32
5.3 Internet browser usability	33
6 DEVELOPMENT IDEAS	35
7 CONCLUSIONS	37
REFERENCE	41
APPENDICES	44
Appendix 1. The Login window of the demo release	44
Appendix 2. The Requirement shell	45
Appendix 3. The Stakeholder definitions	46

ABBREVIATIONS AND TERMS

.NET	Microsoft software framework
ASP.NET	Microsoft dynamic web application framework
Azure	Microsoft cloud service
CSS	Style sheet language
Excel	Microsoft spreadsheet application
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IaaS	Infrastructure as a Service
InnoDB	MySQL database engine
Java	Client-server web application programming language
JSON	Java Script Object Notation
JSP	Java Server Pages
LAMP	Apache, MySQL, PHP/Perl/Python software bundle for Linux
LTS	Long Term Support
MongoDB	Open source NoSQL database
MySQL	Open source database
OS	Operating system
PaaS	Platform as a Service
PC	Personal Computer
Perl	General-purpose scripting language
PHP	Server-side scripting language
Python	General-purpose programming language
TAMK	Tampere University of Applied Sciences
UML	Unified Modelling Language
VPN	Virtual Private Network

1 INTRODUCTION

The internet has greatly influenced information sharing for 25 years, for example, decision making in business and contributed businesses on their customer relationships. The history of web applications started 1995 when Netscape introduced Java script language, for example, enabling user input check on a web clients instead of web server. Further development of web programming languages has enabled, for example, email applications and short message services between privates, communities and companies. Fast internet connections has enabled, for example, video meetings and remote web trainings. The database has enabled a system to store and manage information. Cost-effectiveness and reliability on sharing information is achieved using cloud technologies.

An example of web application can be an appointment system, where user manages his/her appointments using the web browser as a front-end system and in a background operates the cloud server having appointment database and server software, which communicates with web browser. In the appointment system users can look free appointments, manage current ones and one appointment is reserved for one user at a time. Previous principle is a base for many web applications where web browser offers user interface to manage information on database.

The document discusses the implementation of the web application for company daily task management. Web application is composed of user interface on web browser and back-end Virtual machine in cloud. The tasks can be viewed and/or modified through user interface based on web application user access rights. The access rights, tasks and other application related data are in the Virtual machine database. The server software manages the database data and communicates with the web browser when management operations are executed on web browser. Previously the tasks were managed in Excel-file, which was stored in a company server and accessed through VPN. With this way, the information sharing nor real time follow-up of the tasks were not possible. Instead of web application alternative would have been purchasing commercial software or other tailored tool, but there were no will to invest for commercial software, like Microsoft SharePoint, or will to subcontract such tool. The web application was the best option for the case also because company got an asset, which could be developed later on in-house, to similar needs.

2 BACKGROUND

2.1 Main requirements

The engineering project starts when someone, like customer, wants to tackle a technical problem there is and the when the main requirements are identified.

When the company did survey of how they should follow daily tasks, following main requirements were found:

- The task management must be easy and simple
- The tasks should be accessible despite of time and place
- Simultaneous task management shall be possible between users

The three main requirements lead to the following conclusions:

- Easy and simple task management requires the task data is structured and clearly presented, and the presented information cannot be misinterpreted. In addition, the manipulation of the task data is logical and happens in controlled way. To enable such concept there were a possibility to implement a web application with simple looking web pages where the information is presented, for example, as a two dimensional table and executable operations are minimal and logically arranged in web pages.
- Task availability despite of time and place would need a system, which runs round-a-clock with no interruptions providing access to data. As the idea was web application, the internet sounded as the communication channel. What would be then the system, which provides the round-a-clock access through internet? Understanding the current technologies the conclusion was Cloud services, which can fulfill the requirement.
- Simultaneous task management by several users would need some control mechanism to allocate single task to the one user at a time and provide a copy of the same task for another. The latest edit of the single task data should be saved. This kind of mechanisms exists in databases and fulfills the requirement.

The main requirements were now analyzed and system parts, like web pages, cloud service and database, formed now the design skeleton of the web application. Next there were need to analyze, which technologies would be needed to implement the system parts.

2.2 Technologies

2.2.1 Virtual machine

Microsoft Azure is published early 2010 to compete in a cloud markets for example with Amazon, AT&T, HP, Google and IBM. Azure provides platform service (PaaS) and infrastructure service (IaaS) for its customers to deploy, for example, development environment or even a server cluster to the cloud. As earlier discussed in Main requirements chapter 2.1, company had the cloud services from Microsoft Azure. The Azure has mainly the three options available for web application purposes, like the Web Sites, the Cloud Services and the Virtual Machines. The Web Sites and Cloud Services options were providing easy setup for some web application purposes to not to worry about the platform maintenance. As the company wanted to have control on platform maintenance to learn more of the service model, the choice of the service was the Virtual Machine (Microsoft Azure, Virtual Machines). The plan was also to have a deployment option of the web application into some other environment if the current one would not somehow be adequate after current cloud service agreement. There were no operational differences between the services from user point-of-view. When the service type was decided, the next thing was to consider the operating system on the Virtual Machine.

2.2.2 Operating system

The Virtual machine operating system choice was made out of Microsoft Server 2012 and several Linux OS distributions, like Ubuntu (Linux on Azure-Endorsed Distributions, Supported Versions). The decision to choose the Ubuntu Linux 12.04.01 LTS operating system was based on good experiences of Ubuntu Linux as a steady and reliable operating system and on the other hand, the aim was to have the web application independent of

commercial software. In addition, the Ubuntu Linux 12.04.01 LTS operating system support continues until autumn 2017. This will provide a solid ground for the web application system updates for coming years. Next was the place to decide, which database to select for the web application.

2.2.3 Database

When the Virtual Machine service was selected as the web application platform in cloud (2.2.1), there were MySQL (MySQL, Open Source Database for Web Applications) or MongoDB (MongoDB, The Leading NoSQL Database) as the databases to select. When company had experience of MySQL database management and no experience of MongoDB, the MySQL was chosen as the database. The MySQL software was included in LAMP software bundle and instructions of installing LAMP was found from Azure documentation. The next thing was to think of what kind of tool/system can be used to manipulate the MySQL database user data.

2.2.4 Server code

The task data is manipulated on database using the SQL statements. The SQL statements are executed, for instance, from specific database tool or from a code, which contains the SQL statements. The use of specific database tool with the web application would mean extra administration work with tool and user credentials, though it is not an option. The code could be located in the Virtual machine as an interface to database containing the SQL statements and the SQL statement execution would be controlled by the code. The web application would then need to implement some user interface where user manipulates the task data, based on the roles defined in a code. Applications normally works with databases in this way. So, next was to consider the programming language for the server code. The programming language for the server code should have MySQL software library to support the SQL statements executed in MySQL database. The company had a knowledge of the most popular server-side programming language called PHP (W3techs, Server Side Programming Languages) and PHPs' capability on this kind of systems. Therefore, the PHP was chosen. The PHP was also included in a server software bundle called LAMP, which was installed in the Virtual machine after MySQL database was

selected. The next phase was to define the communication between server code and user interface code.

2.2.5 Web server

The role of web server is to be a communication system between user interface code and server code and to provide access to information published on the server through the web pages. The selected application web server was Apache (Apache HTTP server), which was one of the installed software on the LAMP software bundle. The Apache HTTP server is the most popular web server software and most likely, the development of the Apache will be supported for many years ahead. The next was to define what the devices to use the web application are.

2.2.6 The device

The device to use web application would need a capability to connect internet using either fixed, wireless or mobile system network technologies. The device communicates with other devices with many kind of internet protocols, like Bluetooth (Bluetooth, Wireless Technology Standard) or HTTP (HTTP, Internet Application Layer Protocol). The HTTP is an application protocol designed for data communication in internet and obvious protocol on the web application operations. The web application was considered to operate in every modern PC, tablet or mobile phone, which connects to internet and has the graphical web browser. Then, there were need a code as user interface, which would be supported by the web browser.

2.2.7 User interface code

There is hundreds of different internet browsers, which can be used in internet devices. Most of the browsers are graphical and some of them are text based. The web application was planned only for graphical internet browsers to provide better readability and usability compared to text based browsers. The world largest web development site

(W3Schools, Online Web Programming Tutorials) refers to the few most popular graphical internet browsers, like Firefox, Internet Explorer, Chrome, Safari and Opera. The web application was to be built using the most common web programming techniques for each of the browsers, to enable similar operation of the web application.

The web application client was planned to be as a web page user interface into the Virtual machine database. There were no need for multimedia techniques on the user interface so the basic functions of HTML language, like forms, links and buttons were enough to build the user interface. To have a nice outlook for the web page the CSS language was needed for arranging HTML components on the web page. The jQuery components, which enabled usability, like having calendar date selector brought some usability to web pages. At this point, of the system analysis, the technologies were chosen and the web application main system components were defined.

2.3 Programming languages

The programming languages, like PHP (PHP5, Introduction), HTML (Hypertext Markup Language, Introduction), CSS (Cascading Style Sheets, Introduction) and Java (Java, Programming Language), were selected at the same time when the web application background analysis and system components were defined in chapters 2.2.4 and 2.2.7. These programming languages were taken to the implementation.

2.4 Software tools

The software project may need many tools to perform different tasks on project, like documenting, software development, project management and visual designs. The web application software project concentrated to utilize open source tools to have the experience of the maturity level of the tools compared to commercial tools. The following chapters introduces the tools used on this project.

2.4.1 LibreOffice

The LibreOffice software bundle was used for documentation in this project and it is freely available office suite having various tools, like Writer for writing documents or Impress for drawing diagrams. The capabilities of the LibreOffice tools are very much in a same level than, for example, the Microsoft Office tools, and the LibreOffice was seen appropriate software for the purpose. The LibreOffice is supported in Microsoft Windows, Mac OS X and Linux platforms.

2.4.2 ProjeQtOr

The ProjeQtOr (ProjeQtOr, Open Source Project Management Software) is freely available open source tool for project management. The ProjeQtOr has all needed features for the project management, like budget follow-up, but the main criteria for having the tool was to share the planned tasks and documentation among the project group. The sharing was achieved by installing the tool into the Virtual machine and by defining the user accounts and rights for the project group members. The ProjeQtOr is very usable project management tool, for instance, to try out ad-hoc projects.

2.4.3 MySQL Workbench

MySQL Workbench (MySQL Workbench, Visual Database Management Tool) is freely available tool for managing MySQL databases. User can manage, for example, database connections, structure and content. User can also test database building by generating SQL statements from the database model.

2.4.4 phpMyAdmin

The phpMyAdmin (phpMyAdmin, MySQL Administration Over the Web) is freely available graphical web tool for managing database done with PHP. The tool is included in LAMP software bundle and the tool has, for example, a capability to get database creation

script as input file to build the database. All database management tasks can be done, like adding users, removing columns or language settings.

2.4.5 NetBeans IDE

The NetBeans IDE (NetBeans IDE, Integrated Development Environment) is freely available software development tool made with Java and it was taken into use for programming the web application code based on previous experiences of the tool. The tool supports many programming languages, like CSS, HTML, Java and PHP. The tool has also debugging feature, which is not included in basic source code editors like Notepad++. The NetBeans IDE is very suitable on projects, like the web application development.

3 DESIGN

The design of the web application started from the project plan definition to state the resource boundaries for the project and to get an agreement of the project duties and tasks within a project group.

3.1 Project plan

The project plan is one of the main documentation for project. Project plan consists at least project description, objectives, project personnel, responsibilities, resources, implementation plan, test plan, communication, risk analysis and maintenance plan. All the previously mentioned project content was in a plan, but some of the main, from authors' perspective, project content is explained in following chapters.

3.1.1 Implementation plan

The implementation plan contained the main tasks of the project. The definition of the main tasks came out from prioritization of feature requirements where the importance of each application feature was evaluated based on the main application requirements and background investigations.

The main implementation parts were the Virtual machine setup with MySQL, PHP and Apache and the web application web pages. Several higher-level tasks were defined from these two implementation parts and tasks are listed in a following TABLE 1, as an example.

Each task on list contained many sub tasks and those were prioritized like High, Medium and Low –categories. The Period -definition on task list helped the designer to concentrate the tasks of each period.

TABLE 1. The task list example of the project main tasks

Task title	Priority	Period
Virtual machine and internet connections setup	High	1
Project management tool installation	High	1
LAMP installation	High	1
MySQL database setup	High	1
Apache web server setup	High	1
Design database information model	High	1
Web application database deployment	High	1
Web application web page design	High	2
Login page implementation	High	2
Main user window implementation	High	2
Add task window implementation	High	2
Edit task window implementation	High	2
Delete task feature implementation	High	2
Logout feature implementation	High	2

3.1.2 Test plan

The web application features must be tested and verified with the application requirements to find the defects of the design and implementation. The project concentrated mostly on usability requirement testing.

The test plan contained the testing of Main requirements (2.1), Use cases (3.4) and Internet browsers (2.2.7). The Main requirements were tested and verified how the implemented web application responses to the initial needs, called as Main requirements. The Use case testing was functionality testing and was performed using the use case specifications. The use case specifications contained the test case report part and no other test case document was needed when the test results were updated to use case document. The internet browser testing was usability tests of how the internet browsers behaves when using the web application.

3.1.3 Maintenance plan

The maintenance plan contained the task list of the web application maintenance operations. The task list stated what, when and how the maintenance shall be done. The following TABLE 2 shows an example of the maintenance task.

TABLE 2. An example of maintenance task description

Operation	Schedule	Main actions
Update Linux OS	Every 3 rd month, At Sunday, at 20:00	<ul style="list-style-type: none"> • Establish secured connection to the Virtual Machine as a root user • Execute full system backup • Perform the OS update • Test the web application works normally with Viewer/Manager user roles.

3.1.4 Project risks

The project risks were written down to a risk list to remind project group what kind of things may cause problems to the project and how those problems could be mitigated or solved. The following example shows one risk specification from the list TABLE 3 as an example:

TABLE 3. The example of a risk specification

ID	Risk	Mitigation alternatives
001	The cloud service provider sends a free account termination notice and sets a minimum price of 30 euros / month for all of the accounts.	<ol style="list-style-type: none"> 1. The new fee is accepted and no actions to be done. 2. Create, copy and install the Virtual machine image to another cloud service provider Virtual machine/own server and terminate the subscription with the current cloud service provider.

If the risk realizes the actions are taken according the plan. If there is no plan of actions it may cause some other problems of using the web application, like when the account is suddenly terminated and there is no image available of the Virtual machine to be installed into another environment.

3.2 Architecture

The Architecture of the web application was modeled after main components and technologies were defined. The purpose of the architecture model was to steer the implementation according to the selected technologies. The FIGURE 1 describes the architecture model with main components of the web application and the communication between the device and Microsoft Azure through the Traffic control. The communication ports, like SSH and HTTP, were opened in Traffic control. The port opening was mandatory operation before any communication could be established.

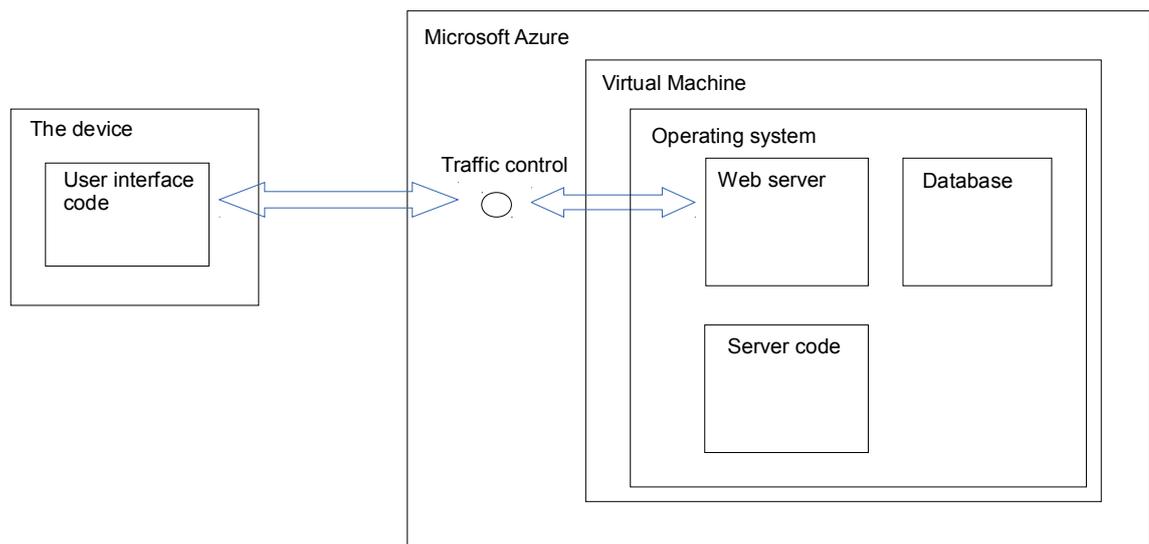


FIGURE 1. Architecture model of the web application

3.3 Stakeholders

The stakeholders are the most important influencers for the product in question. The stakeholders has some interests towards the product, which will affect negatively or positively to the product and its lifecycle. For example, the investor is a stakeholder, which

affects positively to the product by financing the product development. On the other hand, the competitor affects negatively to the product, for example, by judging the product publicly. It is important to find all stakeholders and to define their impact to the product. When the stakeholders are found and their impact towards the product is defined, it will clear out the business segment and the opportunities of the product.

To define the stakeholders of the web application there is need to describe somehow the environment where the application operates and works, what human actors and systems are related to the application. The process to describe such environment involves discussions with application owner (which is the one who orders the application), users and designers to get the idea of the product relations to its environment. The application architect may ask the following questions:

- Who affects the application negatively or positively?
- Who are the users and how they use the application?
- What is the physical environment where the application works/operates?
- What systems affects to application negatively or positively?

As an example of stakeholder, we can take the technology standard and its requirements for the product. Technology standard steers the application implementation and operation during its life cycle.

The Onion model is commonly used diagram to present product stakeholders. The reference of the Onion model (Onion model, The Proposed Taxonomy) explains the theory.

3.3.1 The Onion model

The web application stakeholders were defined utilizing the Onion model theory. The stakeholders of the product was defined by stakeholder role in a circles around the product in a middle. The FIGURE 2 shows the web application (Task Management Tool) stakeholders as titles, like IT specialist, of things or persons in the Onion model. The titles are added into the TABLE 4 in chapter 3.3.2.

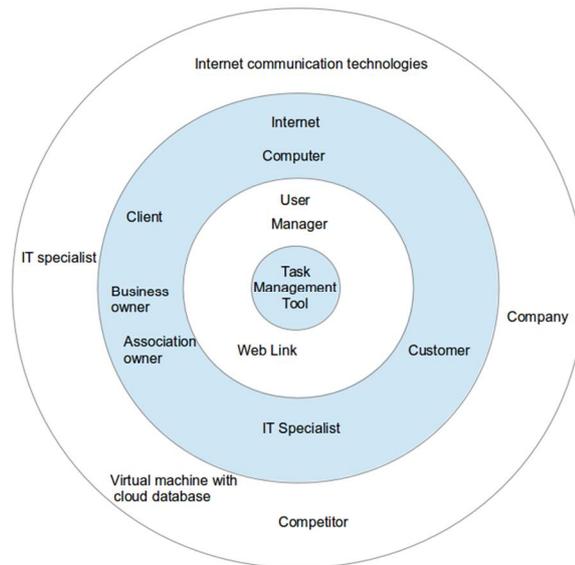


FIGURE 2. The Onion model of the web application stakeholders

3.3.2 The list of web application stakeholders

The part of the stakeholder definition process was to form a list, TABLE 4, of the web application stakeholders matching with the Onion model stakeholder role definitions. For example, the company had a title, like Manager, which took two stakeholder roles because the Manager would take care of the tool maintenance and advising other users to use the tool. The rest of the titles were matched with the same way until all web application stakeholders were identified. If new stakeholder is found later, it is added to the list and the role is defined. The new stakeholder may bring some changes to the operations done with the tool. The tool implementation change may also affect to some other stakeholder or application feature, and so on. The process to identify stakeholders is ongoing.

TABLE 4. The example of the list of stakeholder definitions and web applications stakeholders

Stakeholder role	Stakeholder definition	Web application stakeholder
Normal operator	Role that involve giving routine commands and monitoring outputs from the product, whether these are via a human-computer interface or not.	User
Maintenance operator	Role that involve maintaining the product, such as servicing hardware, and diagnosing and fixing faults.	Manager
Operational support	Role that involve advising Normal Operators of a Product about how to operate it.	Manager
Functional beneficiary	Role that benefit from the results or outputs created by the Product.	Association owner
Interfacing system	Role responsible for neighboring systems that have electronic or other interfaces to/from the Product.	Computer, Internet, Web link
Purchaser	Role responsible for having the Product developed.	Company
Sponsor	Role responsible for initiating development of the Product, for obtaining funding for it, and for protecting the development from 'political' pressures and funding cuts.	Business owner
Negative stakeholder	Any role that could be harmed by the product physically, financially or in any other way that might be found justifiable by the authorities (e.g. a court of law, a Regulator), or conversely that could attempt to harm the product.	Competitor
Political beneficiary	Any role in public office or private business that can benefit in terms of power, influence and prestige through the success of the Product.	
Financial beneficiary	Any role that can benefit financially from the success of a Product.	Company, Client, Customer
Regulator	Any role responsible for regulating the quality, safety, cost or other aspects of the Product.	Internet communication technologies, Virtual machine with database
Developer	Any of the many roles (requirements engineer, analyst, designer, programmer, tester, safety engineer, security engineer, electronics engineer, metallurgist, human factors engineer, project manager, etc.) involved directly in Product development.	IT specialist
Consultant	Any of the many roles (marketing expert, software expert, business analyst, management specialist, etc.) involved in supporting some aspect of Product development, characteristically from outside the development organization.	IT specialist

3.4 Use cases

To define a use case one should think what kind of actions system users want to do with the system and how the system will response. The use cases specify who uses the system, what are the conditions of the system before and after use case, what happens during the use case and what are the positive and negative events after use case has realized.

The use case could presented with a drawing the behaviour, writing the event or drawing and writing. The drawing and writing method was utilized on this project. The idea of using drawing is that it will open up the relations between user and system, it will clarify the role of user and system and it will also clarify new/obsolete dependencies, roles and use cases. Writing open the use case will help the system stakeholders to understand what the system is supposed to do when user wants to use the system with some way.

Following use case diagram FIGURE 3 shows all the main use cases (there are also sub use cases) where the web application development is concentrating on. The FIGURE 3 was drawn using UML notations, which are explained in UML (Unified Modeling Language), but the main idea is to form a diagram where are the users, system and relations explained as accurately as possible.

3.4.1 Use case diagram example

The FIGURE 4 shows the Log into system use case and the actors of the use case.

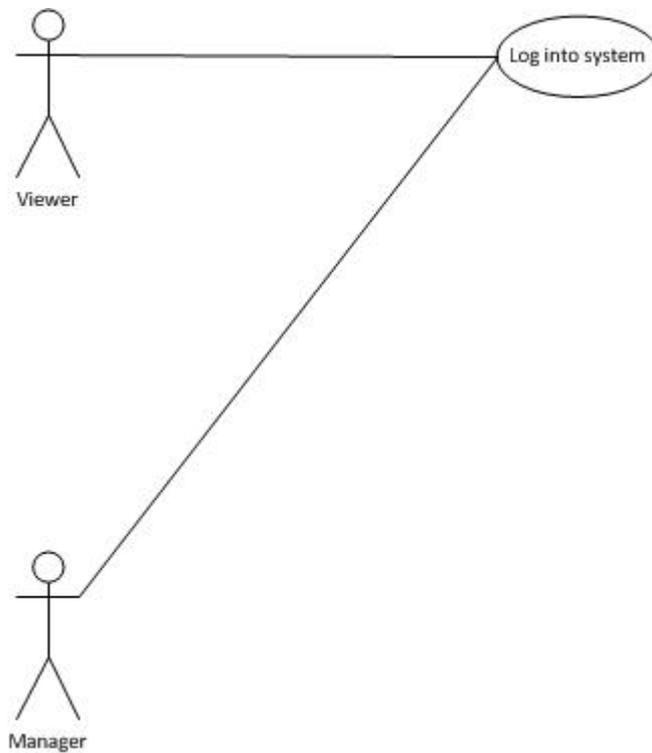


FIGURE 4. The Log into system use case diagram

3.4.2 Use case specification example

The use case specification helps to understand what happens on a system when actor uses the system in certain way. The use case specification should define at least the content defined in TABLE 5. The “Test result and remarks” –item is applied as extra item for the project reporting purposes.

TABLE 5. The Log into system use case description

Use case title	Log into system
ID	001
Pre-conditions	<ul style="list-style-type: none"> • User has entered the web application web site • User has his/hers username and password available (Viewer/Manager) • User is entering his/hers username and password
Description	<ul style="list-style-type: none"> • User enters his/hers username and password and press “Login” –button or Enter • User is successfully logged into system
Post-conditions	<ul style="list-style-type: none"> • User is logged into system
Exceptions	<ul style="list-style-type: none"> • User username and/or password is wrong and login fails. The error message ** is shown • User is not logged into system and login fails. The error message ** is shown ** “Wrong Username or Password! Try again.”
Test remarks and results	<p><i>This part of the use case specification must be used for writing test remarks and results.</i></p> <p><i>Result: Test passed/Test not passed</i></p> <p><i>Date: dd/mm/yyyy</i></p> <p><i>Version: x.x</i></p>

3.5 Requirements

The requirements starts to realize when the idea of the application or product is figured out. The requirement specification process must be iterative to gain usable requirement, for example, for use case specification process. The iteration means challenging the requirement through stakeholder discussions. The web application Main requirements, in chapter 2.1, were challenged and discussed with the stakeholders few times to get such requirements, which would steer further the system analysis, the web application technology selections and the design process.

The Volere (Volere Requirements Specification Template, Requirements Shell) Requirement shell (Appendix 2.) was used as a basis for requirement writing on the project. The Requirement shell collects all relevant information into one requirement specification. The web application requirements were saved into ProjeQrOr tool to match the requirement to the project tasks and to manage them in one place.

TABLE 6. The requirement specification example based on Volere Requirement shell

Requirement id	001
Requirement type	Functional
Use case	Login into system
Description	User must login into system using valid username and password
Rationale	The username and password are the login data used for login into system. The system administrator provides the valid username and password for the user.
Originator	System architect
Fit criterion	User is able to login into system after valid username and password is given
Customer satisfaction	5
Customer dissatisfaction	5
Conflicts	Not identified
Priority	High
Supporting material	Use case specification
History	Created on 2.5.2013

3.6 Web page design

Usability problems with web pages and applications leads to unhappy users and then applications are most likely abandoned. Therefore, web page design is becoming more important when different kind of internet devices, like tablets and mobile phones having internet browsers are used. Web page designs can be done either on paper or digitally with the specific design software as wire frame designs. The web page wire frame designs are guidelines to web page designers who then designs the web page with technology and usability in mind. The web page designer shall also follow the web page functionality and

outlook requirements to build the web page as a whole. The FIGURE 5 is an example of the wire frame sketching with LibreOffice Impress showing the information needed for Login window web page design work.



FIGURE 5. The wire frame sketch of Login –window

The Login window design may change on implementation, but the main idea of the page functions should be there according the design, which is based on stakeholder requirements. The Appendix 1 shows the implementation result of the Login window at demo release.

3.7 Information security

The information security has always been essential part of any business or government activities and it is easily forgot how much, for example, leaked information impacts on business in general. A company may lose its business advantage for competitors just by letting its systems open to internet, because of bad security design or lack of user control, on where hostile parties can spy or steal business critical information. To protect the web application from information thefts, some security designs were done to find out the web application weaknesses. A good starting point on defining the information security mechanisms for the web application is the architecture description. Every system part has its own defects and shortages, which may cause problems on securing information on system

and these must be acknowledged on web application implementation. The communication is another subject to investigate between system parts, like the user interface code and the web server.

Performing at least some kind of threat analysis of the system is always a minimum to be done rather than no threat analysis. The threat analysis of the system parts and the communication were done to define the security functionalities.

3.7.1 Threat analysis

The threat analysis may feel a bit special and may sound tedious task to do among the whole implementation, but the fact is if the analysis is not done, in any level, the information security shortages are harder fix when they realize unexpectedly. To analyse threats there is different kind methods, like threat tree or attack tree, but maybe the fastest and easiest way is to look at the use cases and think how one could misuse the system, as a Misuse case, and Wikipedia definition (Misuse case, Overview) says “A function that the system should not allow”. To define the misuse cases of the web application we used the use case diagram and listed all the use cases and misuse cases into a table with descriptions. As an example of misuse case definitions, the Login into system and Set HTTP address use cases were opened with the misuse cases with the use case diagram, FIGURE 6, and then with the use case- misuse case table, TABLE 7.

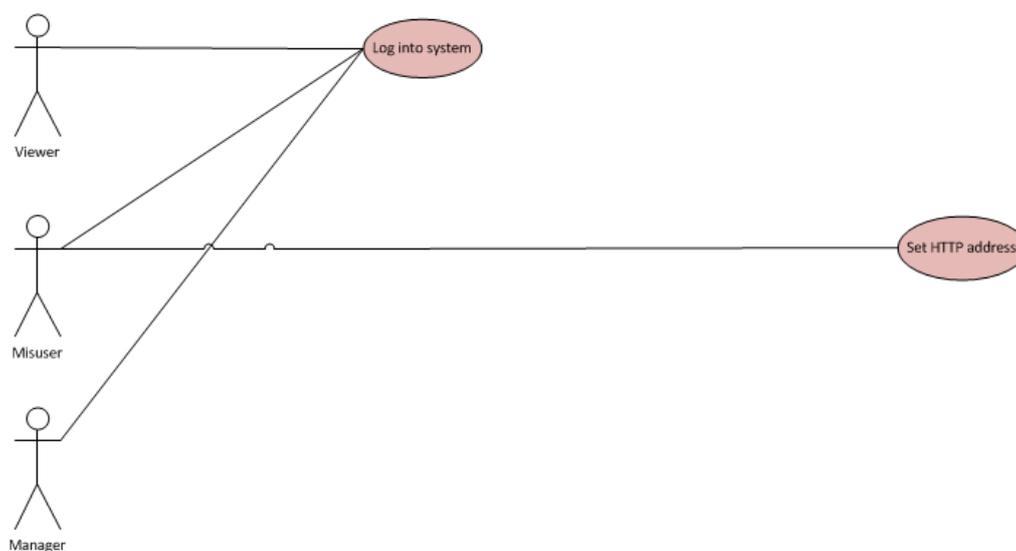


FIGURE 6. The example of Misuser and use cases

The Set HTTP address is a use case, which is taken care of by the system and user is not setting any HTTP address. Misuser tries to set the HTTP address when she wants to get a control of the web application. The following TABLE 7 explains the FIGURE 6 from Misuser point-of-view.

TABLE 7. A table to describe use cases and misuse cases

Use case	Misuse case : A function that the system should not allow	Measures
Log into system	Login with invalid username/password pair	Ensure the server code input validation accepts only registered username/password pair.
Set HTTP address	Inject code into http address field to authorise any user login	Ensure the web application web pages do not allow any code injection to login into system.

The quality of misuse case definitions depends the use case definitions, so the use cases should be defined carefully in first place. When misuse cases are defined it's a place to figure out the security features, according the measures stated on use case – misuse case table, which excludes the misuse cases. The security feature analysis work can be extended further by adopting methods defined by OWASP (OWASP, Application Security Verification Standard Project).

3.7.2 Security features

The security features were designed using the threat analysis documentation of the misuse cases. The TABLE 7 Measures were used as a basis for the TABLE 8 example of security measures and implementation strategies.

TABLE 8. The example of security measures and implementation strategies

Measures	Implementation strategy
Ensure the server code input validation accepts only registered username/password pair.	<ul style="list-style-type: none"> • Create a table to database where all users and user roles are listed. • The username/password pair in table is only accepted by the authentication and authorization system. • Define the roles to specific resources. <p>AND/OR read more information to protect application from SQL injection</p> <ul style="list-style-type: none"> • https://www.owasp.org/index.php/OWASP_Application_Security_FAQ#SQL_Injection
The login web page http address must not allow any code injection to login into system.	<p>Perform the measures to prevent Cross Site scripting, XSS</p> <ul style="list-style-type: none"> • https://www.owasp.org/index.php/OWASP_Application_Security_FAQ#SQL_Injection

In practise the output of threat analysis and security feature implementation strategy definitions is added to the same documentation to avoid redundant documentation of security measures and improving use case iteration process.

4 IMPLEMENTATION

The web application implementation started with setting up the Microsoft Azure Virtual machine environment, which was the backend of the web application. The Virtual machine deployment started by entering Microsoft user credentials on Azure web site and then the cloud service portal was opened. In a portal, user was able to deploy and manage available services, like the Virtual machine with specific operating system. The Virtual machine deployment was followed by the instructions found from Microsoft Azure web site. Then following the instructions of LAMP software bundle installation brought the rest of the needed software for web application. The deployment took one day and it was relatively easy task using the Azure documentation.

Implementation continued with web page designs. The web pages were designed as wire frame drawings where the application windows were drawn according to the usability requirements of the web application browser views. The next phase was to implement the browser views with HTML and CSS according the drawings. The HTML code contained basically tables and buttons. The table view showed the data queried from the database tables with specific columns and rows. The CSS code was written to a separate file to finalize the outlook of the browser view. The functionalities on web pages were mostly the SQL queries from database implemented with PHP.

The web application programming part was done with NetBeans IDE where the PHP web application project was established. The project contained all the source codes, like HTML, CSS and PHP. The connection to the Virtual machine was set up in a Netbeans IDE and every time the project was opened the connection was established to the Virtual machine. The web browser was set to show how the code behaves after code was saved on the Virtual machine.

The programming language instructions and examples were searched mostly from the internet. One excellent web site is called stackoverflow.com (stackoverflow.com, Questions and Answers for Programmers), which has many discussions of programming solutions and the found information helped and speeded up the web application development a lot. The internet was the main information source also for the rest of the implementation work.

5 TESTING AND VERIFICATION

5.1 Main requirements

The Main requirements (2.1) were the initial needs from stakeholders and therefore the implemented web application should be measured against the Main requirements to see whether the web application responded to the initial needs or not. If the testing of the Main requirements shows the web application has not responded to the initial needs, the Main requirements were not understood and the web application design must be changed. The TABLE 9 shows the Main requirements and the responsiveness of Excel- file and Web application systems for the Main requirements. The responsiveness of the systems was based on subjective evaluations done by stakeholders. The pre-condition for evaluation is the user operates with systems with internet connection.

The results of the Main requirements tests on TABLE 9 shows the task data accessibility and divisibility is better with the web application compared to the Excel file, based on the stakeholders experiences. The reason for better task data accessibility by web application was gained by using HTTP protocol instead of VPN, round-a-clock cloud service and centralized database. The better task data divisibility by web application was gained with the centralized database and the database capability to allow multiple user data modifications at the same time.

TABLE 9. The Main requirements testing with Excel file and Web application systems

Main requirement	Excel file	Web application
The task management must be easy and simple	<p><i>Values:</i> <i>Easy and simple = 2</i> <i>Easy/Simple = 1</i> <i>Not easy and nor simple = 0</i> <i>= 2</i></p> <p>The task data is easy and simple to manage Excel file sheet columns and rows.</p>	<p><i>Values:</i> <i>Easy and simple = 2</i> <i>Easy/Simple = 1</i> <i>Not easy and nor simple = 0</i> <i>= 2</i></p> <p>The task data is easy and simple to manage on internet browser forms.</p>
The tasks should be accessible despite of time and place	<p><i>Values:</i> <i>Accessible = 2</i> <i>Relatively accessible = 1</i> <i>Not accessible = 0</i> <i>= 1</i></p> <p>Tasks are accessible as long as company VPN server is alive and internet device has VPN capability.</p>	<p><i>Values:</i> <i>Accessible = 2</i> <i>Relatively accessible = 1</i> <i>Not accessible = 0</i> <i>= 2</i></p> <p>Tasks are accessible 24/7 on cloud with internet browser.</p>
Simultaneous task management shall be possible between users	<p><i>Values:</i> <i>Possible = 1</i> <i>Not possible = 0</i> <i>= 0</i></p> <p>Microsoft SharePoint is not installed in company server.</p>	<p><i>Values:</i> <i>Possible = 1</i> <i>Not possible = 0</i> <i>= 1</i></p> <p>MySQL InnoDB database engine takes care of simultaneous task management.</p>

5.2 Use cases

The web application functional testing was done based on the use case specifications and all the use cases were tested according the steps listed in use case descriptions. For example the use case of Log into system had following description from where the test environment was set having open the web application website and test credentials. After inputting correct and fault credentials the results were written into *Test remarks and result* in Log into system use case specification, shown on TABLE 10.

TABLE 10. The Log into system use case specification with test results

Use case title	Log into system
ID	001
Pre-conditions	<ul style="list-style-type: none"> • User has entered the web application web site • User has his/her username and password available • User is entering his/her username and password
Description	<ul style="list-style-type: none"> • User enters his/hers username and password and press “Login” –button or Enter • User is successfully logged into system
Post-conditions	<ul style="list-style-type: none"> • User is logged into system
Exceptions	<ul style="list-style-type: none"> • User username and/or password is wrong and login fails. The error message ** is shown • User is not logged into system and login fails. The error message ** is shown <p>** “Wrong Username or Password! Try again.”</p>
Test remarks and result	<p><i>The web site of the application http://www.php worked ok. The available username and password were admin/admin and user/user. The testing was done with both username/password pair and with different combinations. The right username/password combination let user log into system when pressing the Login –button/Enter on keyboard. The wrong username/password combination showed an error message:</i></p> <p><i>“Wrong Username or Password! Try again.”</i></p> <p><i>Result: Test passed</i></p> <p><i>Date: 12/12/2013</i></p> <p><i>Version: 0.2</i></p>

The advantages of testing the use case with the specification is, the testing will realize the possible logic defects and implementation shortages, which can be, for example, a missing operation on input data validation code.

5.3 Internet browser usability

As the aim was to build a web application, which can be used in various internet devices with the most popular graphical internet browsers (2.2.7), the user interface code behaviour was validated the same way in each browser. The behaviour of the internet browsers were observed with device-browser combinations listed in TABLE 11. The remarks of

the usability differences were written compared to Linux Mint 14, Mozilla Firefox 29 as reference system.

TABLE 11. Internet browser usability tests with web application

Device and OS	Internet browser	Remarks
Development PC, Linux Mint 14	Mozilla Firefox 29	Reference system
Jolla smart phone, Sailfish 1.0	Sailfish browser	Very little difference
Samsung NP550P laptop, Windows 7	Internet Explorer 11	Slight differences on object and page outlook
Nokia Lumia 800 smart phone, Windows 7.8	Internet Explorer Mobile	The main task window layout was bit different, but otherwise the same behaviour than Internet Explorer 11
Jolla smart phone, Sailfish 1.0	Opera Mobile Classic 12.1	Very little difference, date colours in main task window were blue instead of black
Acer 5755G laptop, Windows 7	Google Chrome 34.0	Slight differences on object and page outlook. Edit window Additional info –field too narrow.
iPad, iOS 5.1	Safari 7534.48.3	Slight differences on object and page outlook. Edit window Additional info –field too narrow. Text font size should be double.

The internet browser usability testing showed there is usability differences between device-browser combinations. For example, some of the input fields were too narrow, font size was not readable or the page scaling was not appropriate. The differences comes from the missing responsive web design (RWD, Responsive Web Design) methods, which improves usability between various device-browser combinations.

6 DEVELOPMENT IDEAS

The plan was to design the web application to work on different web browsers, like smart phone browsers. However, usability testing revealed problems on web page scaling, input field scaling and font readability. Some research for the problem was made and the reason was missing of responsive design methods (RWD, Responsive Web Design). The methods provides optimal viewing experience, where user would not need to do much resize, pan and scroll on web page to browse the needed information.

The inventory management could be the next application to be developed for managing the company photographing equipment. The equipment users would need to know, which equipment are available and which should be requested to be purchased. The application functionalities would be very similar compared to task management operations, like viewing, adding and deleting equipment, and the implementation would be pretty much reusing the task management application code.

The web application would need the event messaging system, where user could notify others after data updates. This could be an email notification, which would require an email server deployment in the Virtual machine or integration to company email system provided by email service provider. The email server would also benefit the project management tool where the email is used for change notifications.

The business data is many times valuable for business and therefore the data should be carefully managed. Part of the data management is the data backups, which are normally taken from the database, on some frequency. As the company business data is located on the Virtual machine database, the backups should be implemented on there. The backup implementation could be a “hot backup” where database is running and the data is copied as a snapshot, like the InnoDB Hot Backup (InnoDB Hot Backup) does. The company server or cloud storage service could be the place for a database snapshot. The copying shall be automated and run over SSH connection between the Virtual machine and the company server. At least, some kind of automated backup procedure should be implemented, than no backups.

The web application deployment may be needed to be switched to Web Sites or Azure Cloud services (Azure Web Sites, Cloud Services and Virtual Machines comparison) if company business or resources changes. The change for the business may be caused by need to integrate the web application into, for example, Azure Web Sites, where the company has planned to maintain all of the company web applications. The switch to other cloud service type must be planned carefully, because it affects, at first, to the controllability and manageability of the web application. Secondly, the web application code will need some changes. The service switch should be documented beforehand in concept level in company's risk planning, to be ready to response quickly for business changes.

If the web application is mostly used with smart phones the web application could be implemented as native application to increase usability from web browser use. The JSON (JavaScript Object Notation) messaging feature was deployed on the web application and JSON messaging was utilised on Windows 7 phone task browser.

The web application user login authentication mechanism could be replaced by using single-sign-on systems, like CAS (Central Authentication Service, Single-Sign-On Protocol) or Shibboleth (Shibboleth, Single-Sign-On Software Package). When company grows and various tools are used for managing business, the single-sing-on deployment would become more obvious. The single-sign-on system should be selected based on the company's future IT strategy, where identity management is one of the key development subjects.

7 CONCLUSIONS

When the idea of web application was born to support company business operations, there were not much understanding of the whole, what the web application would eventually mean for implementation. First thoughts were there is web browser, some software to be written and some physical server being the core system. It was a starting point the whole thing, but just start. Some studies were made about the web applications, what those really are and what those contain, and then it became clear there is quite much to be done after when you could call a system as a web application. There was a place to make some assumptions, what the company would really need from the web application. The assumptions realized soon to be the Main requirements (2.1) of the web application. The Main requirements were found very useful on the way to the planning, design and testing phase, because those kept the focus on the “real need” and not anything else, every time the Main requirements were reviewed. The reviews of the Main requirements were made during the project plan definition, when system components were defined during the design, and finally when verifying the Main requirements on testing.

When the technologies for the web applications were to be defined, the information reference of Web applications (Web Application, History) were explaining what the technologies would most likely be. The web applications has been there for two decades, and there were many alternatives available to build the web application. The first question was where the application would run? There were choice to build the application into company’s own physical server, but there were no experience of web hosting and the alternative was abandoned. Better choice for server would be somewhere else, where the web application hosting is handled properly. The company had an account on cloud service provider and the cloud offered the service called Virtual machine (2.2.1). The Virtual machine appeared to be a perfect choice, because it can be used for testing and running web applications. The Virtual machine provided also full control of the deployment, which enabled the possibility to learn how to manage the Virtual machine. The Virtual machine deployment with the operating system was very easy, because of the good documentation. The deployment took only some minutes.

As the operating system selection was Linux for backend Virtual machine, the software bundle called LAMP (LAMP, Overview) was widely included as web application software for Linux. The LAMP affected straightly the choices of other technologies, as Apache, MySQL and PHP, so the basis of web application was ready. The frontend, the user interface of the web application, technologies were obvious starting with HTML and following with CSS (Web Application, Programming language). The jQuery functionalities were also considered part of the implementation, to improve usability (jQuery, Simplifying HTML scripting). No other frontend technologies were needed, as the previously mentioned would do the thing. As the internet has a lot of information of all the selected technologies, the learning and the deployment of the technologies were easy, and the web application started to realize bit-by-bit.

The web application project was not possible to carry out without software tools. What the tools would be a where to get them? The open source software was one of the interests in a beginning of the project to keep the development costs low. All the tool selections were after the same low-cost principle and finally the found tools were fulfilling the needs remarkably well (2.4). All the software were freely available on internet. All the project tasks from design to implementation could be done with open source tools. When seeing commercial tools and comparing them to the open source offering, it is a bit wonder that commercial ones are still selling if the argument is only on software features.

The stakeholder definitions were based on the Onion model theory (Onion model, The Proposed Taxonomy). The stakeholder onion model was easy to make as the stakeholder definitions were well defined in theory material. Good theory material saved some time on design. Finding the stakeholders, needed the understanding what the application was about, as how it is going to be used. Eventually, the definitions process cleared out the role of the application in its operational environment. The stakeholder definition process realized requirements and will realize new requirements in future when new stakeholders are identified by the project maintenance group. The stakeholder definition process helped describing the web application use cases. The stakeholder definitions, on this project, would need some more work to get them complete.

The use case specification process was to think the behaviour of the use case, what kind of events occur when the use case is starting to realize. The question to define use case was: What user/system does with the system to get what user/system needs? Then the use

case definitions were written with the pre- and post-conditions, and exceptions, and that was pretty much what was needed in a specification (3.4.2). Some of the requirements were found during the use case definition process and that was very useful from the design point-of-view. The project utilised the use case specifications on testing (5.2) the web application functionalities, adding the test results into specifications, and this way reducing the need for specific test report document.

Dozens of requirements were written according the Volere requirement template (3.5) to obtain such requirements, which are easily understood and can traced to its origin and supporting documentation. To write requirement specification, one must think thoroughly the description, rationale and fit criterion of the requirement, which are the most essential for reader to understand what the requirement is about. The rest of the requirement specification information, like document references, supports the purpose of the requirement. The ProjeQtOr project management tool (2.4.2) was used to save the requirements with specified information, and the place for requirements along with the rest of the documentation, helped a lot the requirement management.

The testing should be performed for any system, because the testing will reveal the rest of the shortages of the system design (System Verification, Purpose and Principle of the Approach). The web application testing with the Main requirements were new experience. As the Main requirements were about usability the testing method was invented to be subjective evaluation of the Excel file use compared to Web application use. The evaluation was seen rational after stakeholders tested the Main requirements. The tests (5.1) showed the web application has advantages compared to Excel file system and the conclusion was the web application design has succeed. The feature testing was done utilizing the use case specifications (3.4.2) rather than writing a specific feature testing document. The feature testing retuned some use cases in logic level and realized some new requirements for the web application.

The web application security is one big concern, for example, for business, because web applications are attacked by multiple ways (Web Application Security Statistics, Summary). Therefore, it was reasonable to perform the Threat analysis (3.7.1) for the web application, to check which vulnerabilities the web application may have. The Threat analysis was done by defining Misuse cases (Misuse case, Overview) and, for example,

the Set HTTP address (3.7.1, TABLE 7) misuse case definition realized the web site address manipulation possibility. The Threat analysis helped to enhance the web application security, as the Security features (3.7.2) were developed based on the findings of the Threat analysis.

One of the key thing found on design process is to have enough iteration for designs, like in Requirements (3.5) definition process stated. It's better to have more iteration than less to get the development work, in the end, closer the required system. However, common sense must be used when iterating the designs and someone in organisation must decide when designs are enough mature for the implementation. The iteration shall be done as a continuous process within the project group when some design is to be shared or iterated (SEBok, System Life Cycle Process Models). In addition, centralized information sharing enables quick reacting for design changes and speeds up the move towards the implementation phase.

The web application development project gave wide experience of the related technologies, experience of how this kind of project should be done and what the project should contain. Some of the project areas (DEVELOPMENT IDEAS) would need some more work, but at this stage, the task management application provides the required functionalities (2.1) and supports the business. The project was excellent learning journey into the web applications and strongly recommended, by the document author, as development project for anyone who are interested of utilizing web applications.

REFERENCE

Apache HTTP server. Read 22.05.2014.

http://en.wikipedia.org/wiki/Apache_HTTP_Server

Azure, Pricing. Read 29.05.2014.

<http://azure.microsoft.com/en-us/pricing/calculator/?scenario=virtual-machines>

Azure Web Sites, Cloud Services and Virtual Machines comparison. Read 22.05.2014.

<http://azure.microsoft.com/en-us/documentation/articles/choose-web-site-cloud-service-vm/>

Bluetooth, Wireless Technology Standard. Read 22.05.2014.

<http://www.bluetooth.com/Pages/Bluetooth-Home.aspx>

Central Authentication Service, Single-Sign-On Protocol. Read 26.05.2014.

http://en.wikipedia.org/wiki/Central_Authentication_Service

Cascading Style Sheets, Introduction. Read 22.05.2014.

http://www.w3schools.com/ess/ess_intro.asp

Linux on Azure-Endorsed Distributions, Supported Versions. Read 22.05.2014.

<http://azure.microsoft.com/en-us/documentation/articles/linux-endorsed-distributions/>

Hypertext Markup Language, Introduction. Read 22.05.2014.

http://www.w3schools.com/html/html_intro.asp

HTTP, Internet Application Layer Protocol. Read 22.05.2014.

http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

How to Set Up Endpoints to a Virtual Machine. Read 22.05.2014.

<http://www.windowsazure.com/en-us/documentation/articles/virtual-machines-set-up-endpoints/>

Java, Programming Language. Read 22.05.2014. <https://www.java.net/>

JavaScript Object Notation, Open Standard for Data Object Transmitting. Read 22.05.2014. <http://en.wikipedia.org/wiki/JSON>

jQuery, Simplifying HTML scripting. Read 29.05.2014.

<http://en.wikipedia.org/wiki/JQuery>

InnoDB Hot Backup. Read 22.05.2014.

<http://dev.mysql.com/doc/refman/4.1/en/innodb-backup.html>

LAMP, Overview. Read 28.05.2014.

http://en.wikipedia.org/wiki/LAMP_%28software_bundle%29

Linux Endorsed Distributions – Supported Versions. Read 22.05.2014.

<http://www.windowsazure.com/en-us/documentation/articles/linux-endorsed-distributions/>

Microsoft Azure, Virtual Machines. Read 22.05.2014.

<http://www.windowsazure.com/en-us/services/virtual-machines/>

Misuse case, Overview. Read 22.05.2014. http://en.wikipedia.org/wiki/Misuse_case

MongoDB, The Leading NoSQL Database. Read 22.05.2014.

<http://www.mongodb.com/leading-nosql-database>

MySQL, Open Source Database for Web Applications. Read 29.05.2014.

<http://en.wikipedia.org/wiki/MySQL>

MySQL Workbench, Visual Database Management Tool. Read 22.05.2014.

<http://www.mysql.com/products/workbench/>

NetBeans IDE, Integrated Development Environment. Read 22.05.2014.

<https://netbeans.org/>

Onion model, The Proposed Taxonomy. Read 22.05.2014.

http://www.scenarioplus.org.uk/papers/stakeholder_taxonomy/stakeholder_taxonomy.htm#onion_models

OWASP, Application Security Verification Standard Project. Read 22.05.2014.

https://www.owasp.org/index.php/Category:OWASP_Application_Security_Verification_Standard_Project

PHP5, Introduction. Read 22.05.2014. http://www.w3schools.com/php/php_intro.asp

phpMyAdmin, MySQL Administration Over the Web. Read 22.05.2014.

http://www.phpmyadmin.net/home_page/index.php

ProjeQtOr, Open Source Project Management Software. Read 22.05.2014.

<http://www.projeqtor.org/>

RWD, Responsive Web Design. Read 25.05.2014.

http://en.wikipedia.org/wiki/Responsive_web_design

stackoverflow.com, Questions and Answers for Programmers. Read 22.05.2014.

<http://stackoverflow.com/>

SEBok, System Life Cycle Process Models: Iterative, Overview of Iterative-Development Process Models. Read 29.05.2014.

http://www.sebokwiki.org/wiki/System_Life_Cycle_Process_Models:_Iterative

Shibboleth, Single-Sign-On Software Package. Read 26.05.2014.

<http://www.internet2.edu/products-services/trust-identity-middleware/shibboleth/>

System Verification, Purpose and Principle of the Approach. Read 29.05.2014.

http://www.sebokwiki.org/wiki/System_Verification

Unified Modeling Language. Read 22.05.2014.

http://en.wikipedia.org/wiki/Unified_Modeling_Language

Virtual Machines Linux Install LAMP stack. Read 22.05.2014.
<http://www.windowsazure.com/en-us/documentation/articles/virtual-machines-linux-install-lamp-stack/>

Volere Requirements Specification Template, Requirements Shell. Read 22.05.2014.
<http://www.volere.co.uk/template.htm>

W3Schools, Online Web Programming Tutorials. Read 22.05.2014.
www.w3schools.com

W3techs, Server Side Programming Languages. Read 22.05.2014.
<http://w3techs.com/>

Web Application, History. Read 29.05.2014.
http://en.wikipedia.org/wiki/Web_application

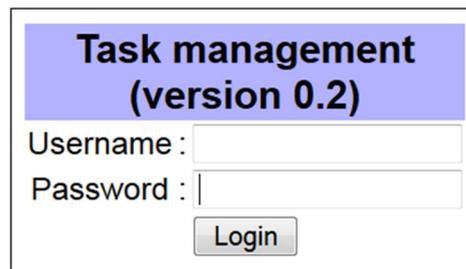
Web Application, Programming language. Read 29.05.2014.
http://en.wikipedia.org/wiki/Web_application

Web Application Security Statistics, Summary. Read 29.05.2014.
<http://projects.webappsec.org/w/page/13246989/Web%20Application%20Security%20Statistics#Summary>

APPENDICES

Appendix 1. The Login window of the demo release

The Login window FIGURE of the demo release is different from the initial design, but has the main features, like username and password fields and Login button for user login into system. The design change for Login window is decided after stakeholder discussions on web application deployment approval occasion.



**Task management
(version 0.2)**

Username :

Password :

Login

FIGURE. The Login window of the demo release

Appendix 2. The Requirement shell

The following TABLE explains the information to be presented on the requirement shell. The official requirement shell is found from Volere Requirement Specification Template (Volere Requirements Specification Template, Requirements Shell).

TABLE. Requirement shell content specification

Requirement id	Unique id
Requirement type	The type of the requirement
Use case	Id of events / use cases, which needs this requirement
Description	One sentence statement of the requirement intention
Rationale	A justification of the requirement
Originator	The stakeholder who raised the requirement
Fit criterion	A measurement of the requirement such that it is possible to test is the solution matches the original requirement
Customer satisfaction	Degree of stakeholder happiness if this requirement is successfully implemented. 1=uninterested, 5=extremely happy
Customer dissatisfaction	Measure of stakeholder unhappiness if this requirement is not part of the final product. 1=hardly matter, 5=extremely displeased
Priority	The relative importance
Conflicts	Other requirements that cannot be implemented this on is.
Supporting material	Pointer to documents to illustrate and explain this requirement
History	Creation, changes

Appendix 3. The Stakeholder definitions

The TABLE explains the definitions of the Onion model stakeholder roles (Onion model, The Proposed Taxonomy).

TABLE. The Onion model stakeholder role definitions

Stakeholder role	Definition
Normal operator	Role that involve giving routine commands and monitoring outputs from the product, whether these are via a human-computer interface or not.
Maintenance operator	Role that involve maintaining the product, such as servicing hardware, and diagnosing and fixing faults.
Operational support	Role that involve advising Normal Operators of a Product about how to operate it.
Functional beneficiary	Role that benefit from the results or outputs created by the Product.
Interfacing system	Role responsible for neighboring systems that have electronic or other interfaces to/from the Product.
Purchaser	Role responsible for having the Product developed.
Sponsor	Role responsible for initiating development of the Product, for obtaining funding for it, and for protecting the development from ‘political’ pressures and funding cuts.
Negative stakeholder	Any role that could be harmed by the product physically, financially or in any other way that might be found justifiable by the authorities (e.g. a court of law, a Regulator), or conversely that could attempt to harm the product.
Political beneficiary	Any role in public office or private business that can benefit in terms of power, influence and prestige through the success of the Product.
Financial beneficiary	Any role that can benefit financially from the success of a Product.
Regulator	Any role responsible for regulating the quality, safety, cost or other aspects of the Product.
Developer	Any of the many roles (requirements engineer, analyst, designer, programmer, tester, safety engineer, security engineer, electronics engineer, metallurgist, human factors engineer, project manager, etc.) involved directly in Product development.
Consultant	Any of the many roles (marketing expert, software expert, business analyst, management specialist, etc.) involved in supporting some aspect of Product development, characteristically from outside the development organization.