

Mobile Application for Warehouse Management System

REALTIME SOLUTIONS

June 16, 2015

Authored by: Yohandi Widjaja

**INTERNSHIP REPORT****FONTYS UNIVERSITY OF APPLIED SCIENCES****HBO-ICT: English Stream**

Data student:	
Family name , initials:	Widjaja, Y
Student number:	2184980
project period: (from – till)	February 2015 - June 2015
Data company:	
Name company/institution:	Realtime Solutions
Department:	
Address:	Torenallee 20 5617BC Eindhoven
Company tutor:	
Family name, initials:	van den Broek, H
Position:	Owner
University tutor:	
Family name , initials:	Zijlmans, J
Final report:	
Title:	Mobile Application for Warehouse Management System
Date:	16-06-2015

Approved and signed by the company tutor:

Date: 16-6-2015

Signature:

A handwritten signature in blue ink, enclosed in a rectangular box. The signature is stylized and appears to be "H. van den Broek".

Preface

This document describes the process and result of my graduation project in Realtime Solutions titled “Mobile Application for Warehouse Management System” which has been conducted from February 2015 until June 2015 by Yohandi Widjaja, a final year student from Fontys University of Applied Sciences.

I chose this assignment because I have an interest in mobile application development and would like to experience mobile development in real life situation. Moreover, I am also interested to specialize myself in this subject for my future career as a developer. I believe this is an ideal subject to specialize in due to the prevalence of smart mobile devices today. Additionally, I also find it quite enjoyable during my time in Mobile Computing class at Fontys.

I would like to express my gratitude to Mr. van den Broek for giving me a chance to have this project in his company. I would like to express my appreciation to all my colleagues at Realtime Solutions for their support, guidance and hospitality. I would also like to thank my friend and family for their unending support and finally I would like to thank my school tutor Mr. Zijlmans for his advice and feedback during the duration of the project.

Yohandi Widjaja

Table of Contents

Summary	7
Glossary.....	8
1. Introduction	10
2. Company Information	11
3. The Project	12
3.1 Initial Situation.....	12
3.2 Desired Situation	13
3.3 The Assignment.....	14
3.4 Project Organization.....	15
4. Research	16
4.1 How to develop a mobile application?	16
4.1.1 Web Apps	16
4.1.2 Native Apps.....	17
4.1.3 Hybrid Apps	18
4.1.4 Research Conclusion	19
4.2 Which cross platform tools to use?	19
4.2.1 Cordova/Phonegap	19
4.2.2 Xamarin	20
4.2.3 Appcelerator Titanium	20
4.2.4 Research Conclusion	20
4.3 How to scan a barcode?.....	21
4.4 Which framework to use?	21
4.4.1 Sencha Touch	21
4.4.2 Famo.us	22
4.4.3 Angular JS	22
4.4.4 Ionic.....	22
4.4.5 Onsen UI	23
4.4.6 jQuery Mobile	23
4.4.7 Kendo UI.....	23
4.4.8 Research Conclusion	24
4.5 What is the best way to test the application?	24
5. Methodology	25
5.1 Prioritizing Functionalities	25
5.2 Requirement Gathering and Research	25
5.3 Design.....	25
5.4 Implementation and Testing	26

5.5 Weekly Scrum Meeting.....	26
5.6 Potentially Finished Product	26
6. Project Implementation Phase	27
6.1 Adding More Framework.....	27
6.1.1 Backbone.js or Knockout.js	28
6.1.2 Result of Previous Comparison.....	30
6.2 Using Require.js	30
6.3 Using External Barcode Scanner.....	32
7. Final Product User Interface	35
7.1 Index Page	35
7.1.1 'Splash' Screen.....	35
7.1.2 'Login' Screen.....	35
7.1.3 'Menu' Screen.....	36
7.2 'Add Stock' Page	37
7.2.1 'Scan location/container' Screen.....	37
7.2.2 'Scan product' Screen	39
7.2.3 'Add product' Screen	40
7.2.4 'Scan location only' Screen.....	41
7.3 'Remove Stock' Page	42
7.4 'Order Picking' Page.....	43
7.4.1 'Next picking task' Screen	43
7.4.2 'Next picking lines' Screen	44
7.4.3 'Print document' Screen	46
7.5 'Inventory' Page.....	47
7.5.1 'Scan location' Screen.....	47
7.5.2 'Confirm stock' Screen	48
7.6 'Settings' Page.....	49
8. Final Product Main Features	50
8.1 Server Connection	50
8.2 Account and Settings.....	50
8.3 Barcode Scanning.....	51
9. Deployment	52
10. Conclusions	53
11. Recommendations	54
Evaluations	55
Reference.....	56
Appendix A: Graduation Project Survey	Error! Bookmark not defined.

Appendix B: Project Plan	V
1. Introduction	VIII
2. Project Statement.....	VIII
2.1 Product Owner.....	VIII
2.2 Project Leader	VIII
2.3 Lead Developer	VIII
2.4 Supporting Developer	IX
2.5 Current Situation	IX
2.6 Project Justification	IX
2.7 Project Product	X
2.8 Project deliverables and non-deliverables	X
2.8.1 Project deliverables	X
2.8.2 Project non-deliverables	X
2.9 Research	XI
2.10 Project risk.....	XI
3. Project Phasing	XII
Phase 1: Start up	XII
Phase 2: Version 2	XII
Phase 3: Version 3	XIII
Phase 4: Version 4	XIII
Phase 5: Version 5	XIII
Phase 6: Version 6	XIV
Phase 7: Version 7	XIV
Phase 8: Version 8	XIV
Phase 9: Presentable Version	XV
Phase 10: Finalization	XV
4. Management plan	XVI
4.1 Money.....	XVI
4.2 Skill	XVI
4.3 Quality	XVI
4.4 Information.....	XVII
4.5 Time	XVII
4.6 Organization	XVIII
4.6.1 Project Organization.....	XVIII
4.6.2 Contacts.....	XVIII
5. Communication plan	XIX
5.1 Interested parties in the project	XIX

5.2 Means of Communication	XIX
----------------------------------	-----

Summary

This document is made to give a detailed information about the process of a graduation project that was accomplished by Yohandi Widjaja, a 4th year ICT student at Fontys University of Applied Science. The graduation project took place at Realtime Solutions, a logistic software service provider based in Eindhoven, The Netherlands. The project is to make a mobile application for Warehouse Management System (WMS) for approximately 100 working days from February 2015 to June 2015.

Realtime WMS is the primary service offered by Realtime Solutions. Realtime WMS support warehouse operation by helping workers with their task (e.g. inventory management, order picking, etc.), and keeps track on all changes and actions within the warehouse. The system relies on barcodes as a unique identifier for all items stored within the warehouse. To scan barcodes, workers used a hand held terminal that is connected to a centralized WMS database via wireless networks installed within the client's warehouse.

Although the system works well, it has a steep upfront cost that makes it unreachable for smaller companies. The reason of the high initial cost are primarily caused by the steep price of the hand held terminal. Every warehouse employee needs it to work with the Realtime WMS while it costs around 1000 € per unit.

As a result, a lower cost alternative to the hand held terminal is needed to reach these potential clients. Realtime Solutions concludes that the best alternative would be to develop an app for smart mobile devices that can replicate some important features of the hand held terminal. A working prototype of the app is expected at 19th of June 2015 for public demonstration at Realtime Solutions 10th anniversary celebration.

To ease development process for multiple mobile device platforms and make use of web development skills already present in the company, the app is developed with Phonegap cross platform tool (CPT). Phonegap allows developer to create a multi-platform mobile application with common web development programming language instead of native programming language that is unique to a certain platform (e.g. Java for android, Objective-C for iOS).

The working prototype produced at the end of this project is able to scan barcodes and replicates several crucial use case scenario of the hand held terminal for smaller businesses. It is also compartmentalized into modules to make future development easier.

Glossary

Terms	Meaning
Android	A popular mobile operating system
Application Programming Interface (API)	A set of functions, procedures, methods or classes used by computer programs to request a service from the operating system, software libraries or any other service providers running on the computer [28]
CSS (Cascading Style Sheet)	Computer language used to describes the look and formatting of an HTML file(s).
Cross Platform Tool (CPT)	A framework to create a cross platform mobile application
Framework	Universal, reusable software environment that provides specific functionality as part of a larger platform to facilitates the development of software applications. In other words, a beefier brother of the Library [29]
HTML (Hypertext Markup Language)	Standard language used to create websites
HTTP (Hypertext Transfer Protocol)	A set of standardized rules for transferring data on the web
iOS	A popular mobile operating system used exclusively on Apple branded mobile devices
Java	A general purpose programming language used in different types of computer. Java is the official language used to create Android apps [30]
Javascript	A programming language commonly used as part of web browsers for various purposes, such as: user interaction, server communication and dynamic content
jQuery	A Javascript library designed to simplify Javascript implementation on HTML file(s).
Library	A collection of resources used by computer programs, often to develop software. These may include configuration data, documentation, pre-written code, values/type specifications, etc.
Mobile Application	A software designed to run on mobile devices
Plugin	A software component that adds a specific functionality to an existing program [31].
Radio Frequency Identification	A technology to track and identify an object with radio signals
Representational State Transfer (REST)	An architecture style to connect network based application that use HTTP request to make communication between the machines.

Supply Chain Management	A broad term to describe a set of processes that ensures a steady supplies of goods to the consumer
Warehouse Management System(WMS)	A system that controls movement and storage of materials in the warehouse
Objective-C	A general purpose programming language commonly used to develop apps on iOS operating system [32]

1. Introduction

A WMS is one of the key part of the supply chain management and is primarily responsible for controlling the movement and storage of materials within a warehouse. WMS often utilize data identification and capture technology, such as: barcode scanner, hand held terminal, and possibly radio frequency identification (RFID) to monitor the flow and status of the products within the warehouse. Data collected through these technologies are connected to a central database to provide useful reports regarding the status of goods in the warehouse.

Like other WMS provider, Realtime Solutions mainly handles a sizeable warehouse common to medium and large companies. The origin of this project begins several years ago. In an effort to gather more profits and expand itself, Realtime Solutions conducted a market research and discovered that there are significant profits that can be made in a new market segment of small scale and low-cost WMS implementation for smaller companies [1]. In order to reach this market, Realtime Solutions needs to lower the start-up cost of their Realtime WMS. Through some research [1], Realtime Solutions has determined that the best way to cut down the cost is to find a much cheaper alternative to the hand held terminal that is used today.

In the past, Realtime Solutions was not able to find any affordable mobile devices that has enough computing power to effectively replace the hand held terminal. Lack of suitable platform forced Realtime Solutions to postpone its plan for entering this new market segment. Meanwhile, recent development of mobile devices have significantly increased its processing power while lowering its cost in the process. Thus, Realtime Solutions recently decided to execute its postponed plan this year and initiated this project in February 2015.

This report is organized as follows: Chapter two is about the company. Chapter three will explain about the project. Chapter four will be about the research conducted and its result in the beginning of the project. The fifth chapter will talk about the methodology used to implement the project. The sixth chapter will talk about interesting activities committed during the project implementation phase. Chapter seven will describes the user interface of the final product, while chapter eight explains the main features of the final product. Chapter nine describes the deployment scenario of the final product. Chapter 10 and 11 will talk about the conclusions and recommendations of this project respectively.

Finally, my personal evaluation of my experience in this project and the reference for resources used in this report will be presented after chapter 11.

2. Company Information

Realtime Solutions is a software service company which focuses on automating logistic processes with an emphasis on storage and transport processes. Realtime Solutions developed its own line of software product: Realtime WMS, Realtime Manifesting and Realtime Tracking. With its products and services, Realtime Solutions continuously improves the efficiency and effectiveness of the customer's logistics operation [2].



Figure 2.1 Realtime Solutions logo.

Realtime Solutions is established in 2005 by Harm van den Broek, an IT specialist with extensive logistics experience. Initially, it focused on inventory management, tracking and tracing. Later on, it also supports the document flow associated with these logistics operation (manifesting) [2]. Currently, Realtime Solutions is experimenting on improving logistics operation further with wearable smart devices and augmented reality.

3. The Project

3.1 Initial Situation

Realtime WMS is one of the main service offered by Realtime Solutions. It supports daily logistics operation in the customer's warehouse and keeps accurate record of any changes or actions within the warehouse. This normally revolves around scanning barcodes placed on the product/shelves and update its location, stock level or status in the database.

Realtime WMS consists of two main parts: **hand held terminal**, and **web console**.



Figure 3.1.1 Example of a typical hand held terminal used in the warehouse.

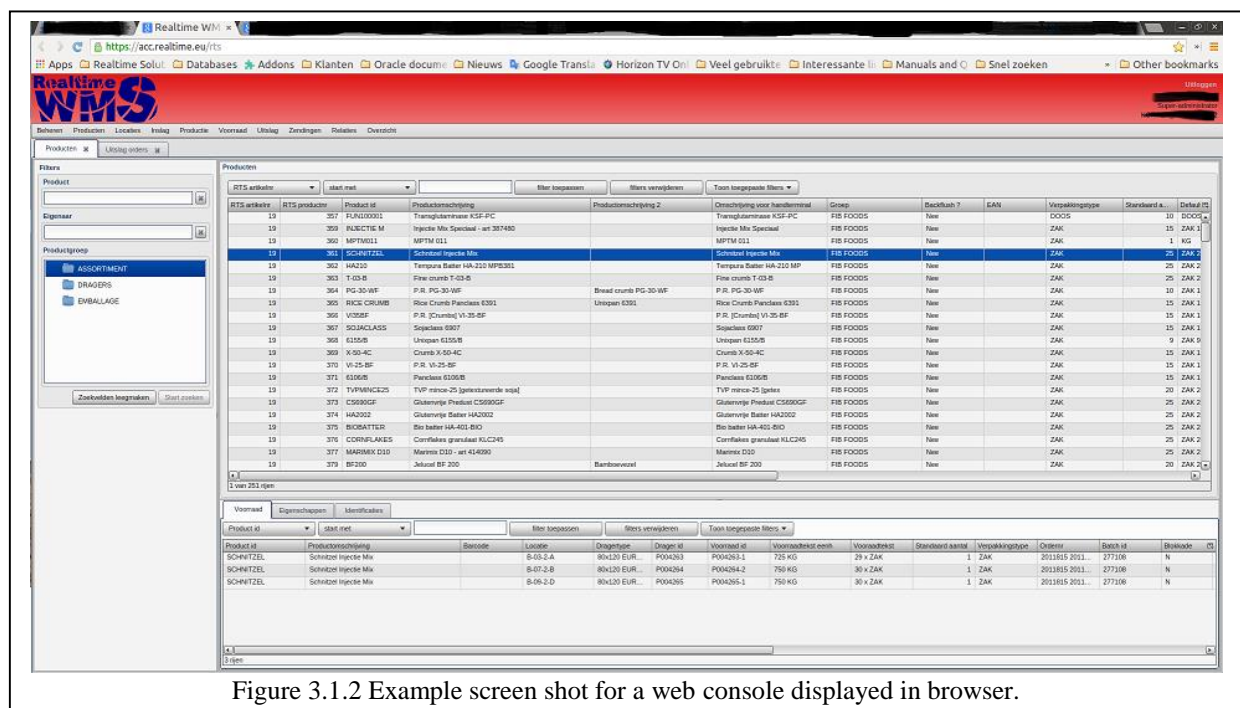
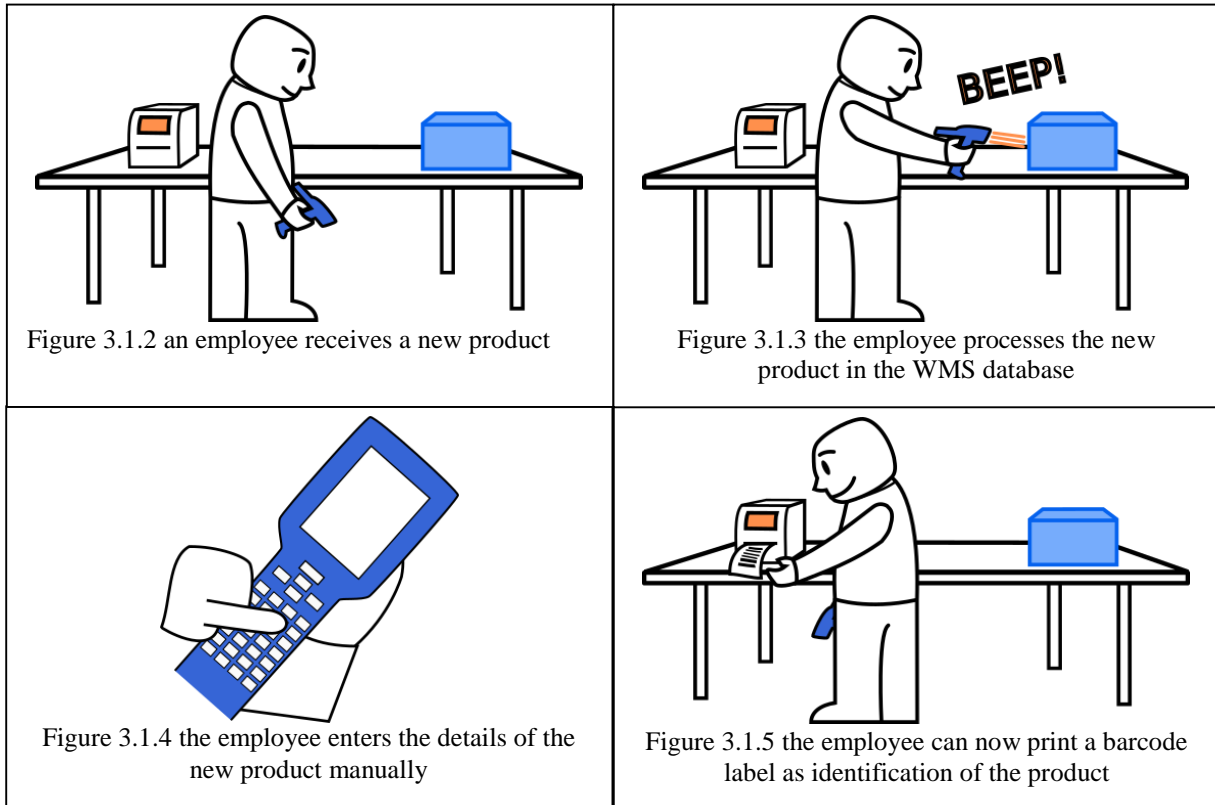


Figure 3.1.2 Example screen shot for a web console displayed in browser.

Hand held terminals are issued to each warehouse employee for daily warehouse operation. It is connected to the WMS database via Telnet wireless network installed in client's warehouse.

Consider the following illustration for adding a new stock in the database as an example:



Webconsole is accessed through normal internet browser and gives oversight, history and extra functionalities reserves for higher ranking employees within the client's warehouse.

The system works well, however it has a relatively high upfront cost as hand held terminals are expensive (1000 € or above per unit) and only works inside the specialized telnet wireless network installed on client's warehouse. As a result, only medium and large companies can afford Realtime WMS.

3.2 Desired Situation

To reach these smaller customers, a cheaper alternative to the hand held terminal is required. Realtime Solutions concludes that the best alternative platform to the hand held terminal are smart mobile devices that are commonly used by the general public today [1].

With the proper application, these mobile devices are capable of replicating many if not all of the hand held terminal's functionalities. Apart from lowering the initial cost for new small scale customers, it will also provide additional benefit to current customers that used hand held terminal. For example, a senior ranking employee that wants to do a quick check of the inventory will no longer need to bring the bulky hand held terminal with him.

3.3 The Assignment

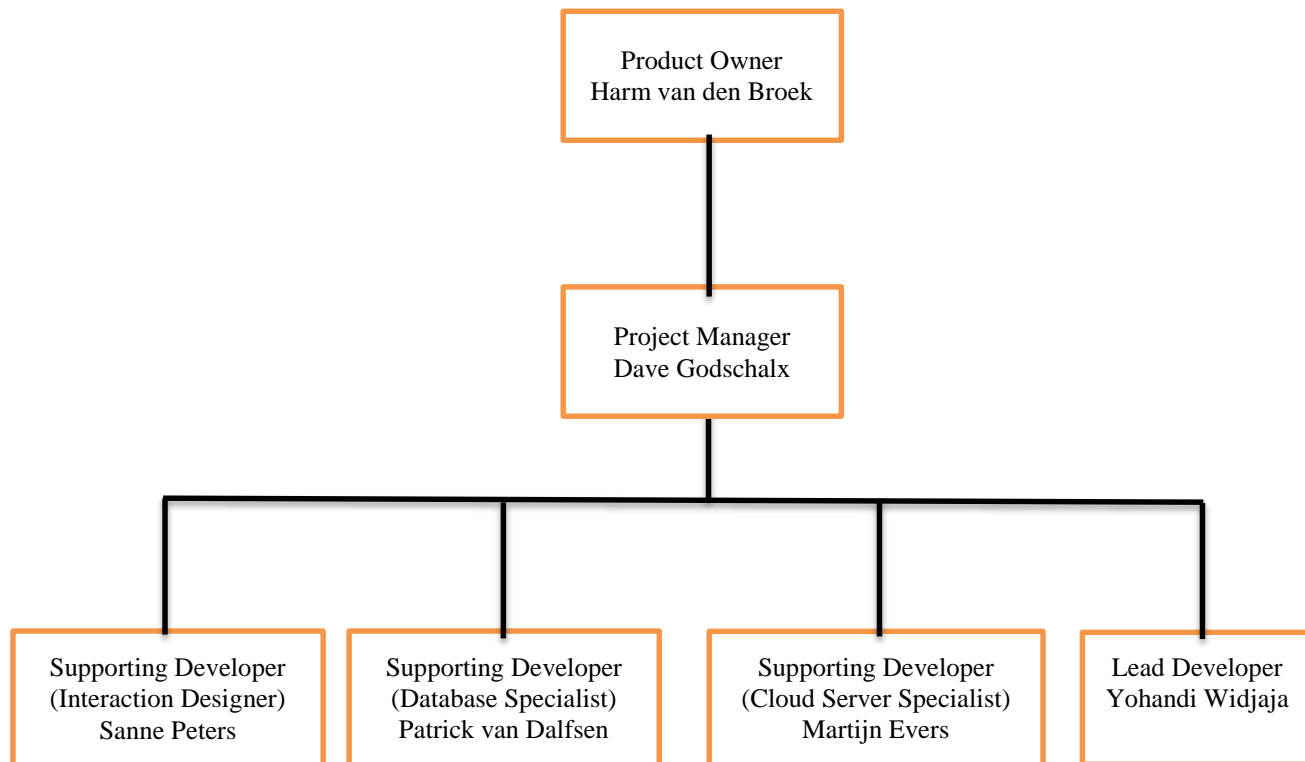
My assignment is to develop a mobile application that will complement the current hand held terminal used in Realtime WMS. The application must have the following basic functionalities [1]:

1. It can read a barcode – this is the most important mobile functionality of the hand held terminal that needs to be replicated.
2. Communicate with the Realtime WMS database via Realtime Solutions cloud server.
3. Use the functionalities mentioned earlier to implement important use case scenario of the hand held terminal in the app.
4. The application must be customizable/extendable so that it is easy for other developer to continue its development in the future.

The application is going to work together with the Realtime Solutions cloud server and the Realtime WMS database server. The cloud server will be the communication bridge between the application and the Realtime WMS database server. All of the necessary calculations and/or database operations are done within the Realtime WMS database server. A workable application is expected at the middle of June to be presented during Realtime Solutions 10th anniversary at 19th of June 2015. The application will then be used by Realtime Solutions for further testing to determine its market desirability.

3.4 Project Organization

To efficiently use the specialization of the other employees, the project is organized as follows:



Product owner provided information about the desired implementation of the functionalities specified on chapter 3.3. Project manager was responsible for the non-technical aspects of the project, such as: marketing, project scheduling, user interface design and testing the application. As a lead developer, my tasks were to research, define the technical requirement, and implement the desired user interface design and functionalities. Supporting developers helped the other project member as necessary according to their specialization.

4. Research

This chapter gives an insight on the research needed for building a mobile application. These research are conducted at the beginning of the project. On this project, Research is important as the initial requirements of the app is quite vague. Realtime Solutions mentioned which functionalities and use case are required but it didn't say anything specific on how to achieve those requirements. Some important research questions which emerged during this phase are listed below:

1. How to develop a mobile application?
2. Which cross platform tools to use?
3. How to scan a barcode?
4. Which framework to use?
5. What is the best way to test the application?

After some research over the internet and consultation with other project members, I found the answers for research questions mentioned earlier. The following chapters will explain what crucial information are produced from each research question.

4.1 How to develop a mobile application?

There are three approaches to make a mobile application: Web Apps, Hybrid Apps, and Native Apps.

4.1.1 Web Apps

Web Apps are programs that run in a web browser. Some web apps are designed to fit in both desktop and mobile web browsers. They are typically accessed by typing a URL in the web browser. There are multiple frameworks that can make your Web app look like a native app. Alternatively, it can be built with media queries to automatically resize the elements size to fits in smaller screen on mobile devices [3][4].

Functional and Technical Specification

Pros	Cons
<ul style="list-style-type: none">• Compatibility <p>Properly designed Web Apps will run on most standard-compliant Web Browsers on any mobile device and desktop.</p> <ul style="list-style-type: none">• Simplicity <p>No installation and updates are needed for the users. Like accessing a webpage, Users only need the URL and internet connection to use the Application.</p>	<ul style="list-style-type: none">• Hardware Interfacing <p>Currently, there is no standard way for a web application to access the device's hardware. The lacks of standard method and tools makes it difficult for Web Application to utilize the device's hardware.</p> <ul style="list-style-type: none">• Discoverability <p>Web application is not listed in the app store.</p>

4.1.2 Native Apps

Native Apps are apps that are written using a platform specific programming language. Native apps have full access to the hardware and the user interface is rendered by the platform SDK.

Functional and Technical Specification

Pros	Cons
<ul style="list-style-type: none">• Performance <p>By using the platform SDK, the app will make full use of the specific platform's capabilities.</p>	<ul style="list-style-type: none">• Compatibility <p>A native application will only work on the platform it is written for.</p> <ul style="list-style-type: none">• Cost <p>Developing for multiple platform will require either an expert that are familiar with several programming languages or several experts that specializes in each platform's programming language.</p>

4.1.3 Hybrid Apps

Hybrid Apps tries to combine the best features of the other two approaches. There are 2 approaches to create a hybrid apps:

1. Use a tool to make a Web apps that are enclosed in native application layer/shell which resemble a native mobile app with extensions/plugins that gives access to device's hardware.
2. Use a tool that can creates native apps from a non-native programming language.

Functional and Technical Specification

Pros	Cons
<ul style="list-style-type: none"> • Compatibility <p>A single codebase can be used to create multiple app for each platform</p> <ul style="list-style-type: none"> • Discoverability <p>Hybrid apps can be added to the app store and installed like any other native apps</p>	<ul style="list-style-type: none"> • Performance <p>Native apps often beat the hybrid solution in term of performance, however recent improvements have narrowed the gap [5].</p> <ul style="list-style-type: none"> • Hardware Interfacing Compatibility <p>Although hybrid apps can access device's hardware with plugins/extensions, not all of them offer 100% compatibility. Sometimes, the plugin for a particular mobile platform is outdated and no longer work or simply doesn't exist yet.</p> <ul style="list-style-type: none"> • Platform-specific Build <p>A hybrid app may needs to be compiled and built to specific mobile platform. Therefore, just like the native app, it needs the target platform's SDK installed on the machine that is going to build them- one SDK for each target platform. This process is rather tedious and complicated as certain SDK are exclusive to specific operating system. (E.g. Windows Phone SDK only works on Windows machine and iOS SDK on OS X machine).</p>

4.1.4 Research Conclusion

After considering the three approaches above, I've come to a conclusion that the Hybrid solutions is the best for my current situation because:

- Cross platform requirement
Realtime Solutions declares cross platform compatibility an important requirement during the research in the initial phase.
- Code Reuse
With hybrid apps, Realtime Solutions can utilize their existing knowledge and codebase in HTML/CSS/Javascript to support the app.
- The WMS application is not considered 'heavy'.
With this consideration, the performance difference between hybrid and native apps will not be noticeable.
- Cloud based builds
Cloud based builds is online service that compiles and builds the core of hybrid application for specific target platform. Theoretically, no additional SDK needs to be installed when such service is used.

4.2 Which cross platform tools to use?

Based on research across the web, I found three Cross Platform Tools (CPT) that I consider worth trying based on their popularity, user base, and the programming language it is based on [6].

4.2.1 Cordova/Phonegap

This is arguably one of the most famous CPT tools currently in the market. Phonegap is an open source framework to create a cross platform mobile application with HTML, CSS and javascript.

Initially developed by a start-up company Nitobi Software, Phonegap quickly gains attention and wins the people's choice award at O'Reilly Media's 2009 Web 2.0 Conference. The framework has been used to develop many apps since. On 2011, Adobe entered an agreement to acquire Nitobi and both company jointly donates the then current PhoneGap project to the Apache Software Foundation. The acquisition also makes Adobe the owner of the name PhoneGap. To avoid confusion and trademark issues, Apache renamed their PhoneGap project as Apache Cordova.

It creates a hybrid apps by covering a base program that use HTML/CSS/Javascript with a native application layer. This is made possible with the 'Web View' web browser widget commonly found on mobile operating system.

PhoneGap Architecture

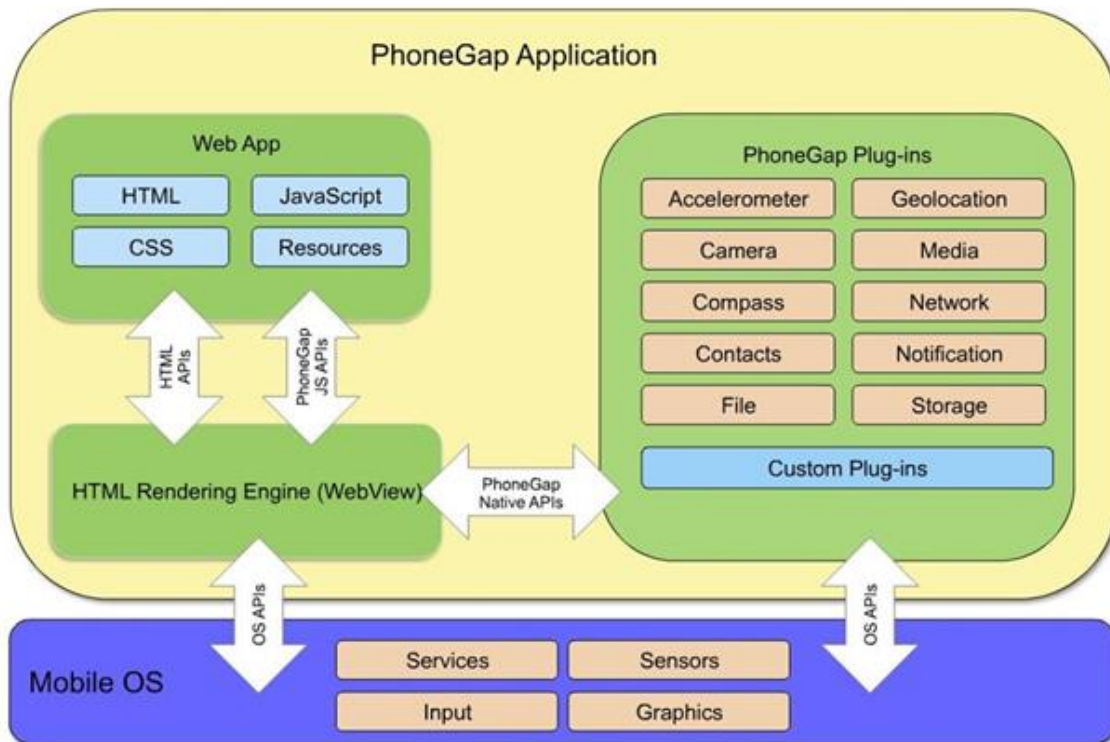


Figure 4.2.1.1 Phonegap architecture diagram [7]

On native application development, 'Web View' is used to display HTML content either from a remote server, or local HTML bundled together with the native application codes. The native application layer generated by PhoneGap creates a 'Web View' that matches the device's screen size, load the web app components and presents it to the user. When the web app needs to access device's hardware, it calls a plugin which then pass the call to the mobile OS just like any other native apps.

4.2.2 Xamarin

Xamarin is a comprehensive mobile development platform to create native apps with C# programming language. It compiles the C# source code into the appropriate native code for a selected target platform [8].

4.2.3 Appcelerator Titanium

Titanium works with similar concept of Xamarin, but instead of C# it uses javascript as the main programming language [9].

4.2.4 Research Conclusion

Considering all of the options above, Cordova/Phonegap is the best CPT for this project for the following reasons:

- It is one of the most famous CPT around with a large developer community and support material.
- Considerably lower learning curve than the other alternatives.
- Realtime Solutions has previous experience in developing web application.

4.3 How to scan a barcode?

Fortunately there are several ready to use plugin found over the internet. Some possible candidates are listed as follows:

1. PhoneGap BarcodeScanner Plugin (free, available in the official phonegap plugin repository)
2. ZBar SDK (free, but iOS only)
3. Scandit Barcode Scanner (fast, not free)

Technically they all work in the same principle, it adds barcode scanning functionalities to an existing Phonegap implementation. They are, however differs in price and performance (particularly speed of scanning and reliability) as shown in the list above. After communicating these findings with the other project members, we have decided to use the PhoneGap BarcodeScanner plugin because it's free and the performance are deemed enough for the time being.

4.4 Which framework to use?

Working with Phonegap is quite similar as working with web applications, it needs a suitable framework to structure its code and/or user interface. There are several frameworks considered for use in this project. The following are the list of framework that is free to use commercially [10].

4.4.1 Sencha Touch

Sencha touch is a framework created from a combined popular Javascript library projects (Ext JS, jQTouch and Raphaël). One of the fastest, most comprehensive mobile web framework [11] [12].

Functional and Technical Specification

Pros	Cons
<ul style="list-style-type: none">• good performance• Large library of UI components• good documentation and user base	<ul style="list-style-type: none">• High learning curve (pure javascript programming).• Paid support and training• Paid supporting development program.• Lack of 3rd party support (because it's a commercial product)

4.4.2 Famo.us

A relatively new open source Javascript framework based on Angular JS to create complex mobile application. This is the only framework that includes an open source 3D layout engine with a 3D Physics animation engine [13].

Functional and Technical Specification

Pros	Cons
<ul style="list-style-type: none">• Offers the best performance	<ul style="list-style-type: none">• Bad documentation• Lacks template, UI needs to be designed from scratch• High learning curve (pure javascript programming).

4.4.3 Angular JS

Angular JS, commonly referred to as Angular is an open source web application framework maintained by Google and a community of individual developers and corporations [14]. It is used for both web and mobile application development. It comes with a Model View Controller (MVC) architecture as a standard. This architecture separates your app into three main component dubbed 'Model', 'View', and 'Controller'. 'Model' represents the data/information used by the app, 'View' takes care of the data presentation on screen and 'Controller' respond to user input and interacts with 'View' and 'Model' to update the screen and application data as necessary.

Functional and Technical Specification

Pros	Cons
<ul style="list-style-type: none">• Good documentation/tutorial• MVC support out of the box• Quite popular	<ul style="list-style-type: none">• Moderate learning curve.

4.4.4 Ionic

Also a newcomer, it's a hybrid mobile framework based on Angular JS. It offers good performance with a nice looking UI [15]. Simply said, it is an Angular.js framework with additional ready to use user interface components which are specialized for mobile application.

Functional and Technical Specification

Pros	Cons
<ul style="list-style-type: none">• Good performance• Comes with a Nice UI straight out of the box	<ul style="list-style-type: none">• Still in beta as it is quite new• need previous experience with Angular JS

4.4.5 Onsen UI

Very similar to Ionic, it used Angular JS on its core [16].

Functional and Technical Specification

Pros	Cons
<ul style="list-style-type: none">• Good performance• Can be used without Angular JS framework.	<ul style="list-style-type: none">• Still in beta as it is quite new• small user base

4.4.6 jQuery Mobile

An extension of jQuery commonly used for web development. It has a wide platform compatibility for mobile and desktop devices [17].

Functional and Technical Specification

Pros	Cons
<ul style="list-style-type: none">• The easiest to learn from the rest of the framework – same syntax as classic JQuery• Largest user base, many working online example• Excellent 3rd party support(especially Stack Overflow)	<ul style="list-style-type: none">• Slower than other framework• Average looking user interface• Does not come with MVC architecture support (can be added with other Javascript framework)

4.4.7Kendo UI

Like jQuery Mobile, it is a jQuery based mobile framework [18].

Functional and Technical Specification

Pros	Cons
<ul style="list-style-type: none">• easy to learn with similar syntax to classic jQuery• Better performance than jQuery Mobile.• comes with MVC architecture support	<ul style="list-style-type: none">• Bad documentation• Bad 3rd party support• Commercial product

4.4.8 Research Conclusion

In general, the list of framework above can be classified into three groups based on their core technology.

They are:

1. Pure Javascript (Sencha Touch, Famo.us)
2. Angular JS (Ionic, Onsen UI)
3. jQuery (jQuery Mobile, Kendo UI)

Pure javascript framework here means that everything needs to be made with javascript. For example, to create a page user must create it by typing some javascript code and let the framework add it instead of creating an HTML file on your own. Considering my experience and skills, pure javascript framework is not something that can be learned quickly to be used on this project. Small userbase, less stellar documentation and high learning curve is not a good combination for a junior developer. This left me with Angular JS-based framework and jQuery-based framework.

Angular JS and other mobile frameworks based on it offers good performance and program structure with its MVC architecture. Learning Angular JS would take some time as they are not particularly beginner-friendly. Moreover, the smaller user base of the Angular JS-based mobile frameworks would also makes it harder to find solutions online. On the other hand, I have experience with jQuery from my previous internship project.

Therefore, I had narrowed the options into two possible choices: jQuery Mobile or Kendo UI. Both are built with the same technology and generally are similar. Performance wise some reviewer considers Kendo UI is better, however JQuery mobile has better documentation, 3rd party plugin support and user base. The jQuery Mobile user base are huge, even larger than the total combination of other mobile frameworks mentioned before [19]. This would make it easier to find solutions online that help me implement the required functionalities and minimize jQuery Mobile's weakness.

I personally believe that Angular.js would be the best choice for this project. However, I chose not to use it because I had a tight deadline and I did not want to spend extra time learning something new while there is an easier alternative right next to it (jQuery Mobile). All things considered, I decided to start this project on jQuery mobile not because it is the best framework out there, but simply because I want to have a working product as soon as possible.

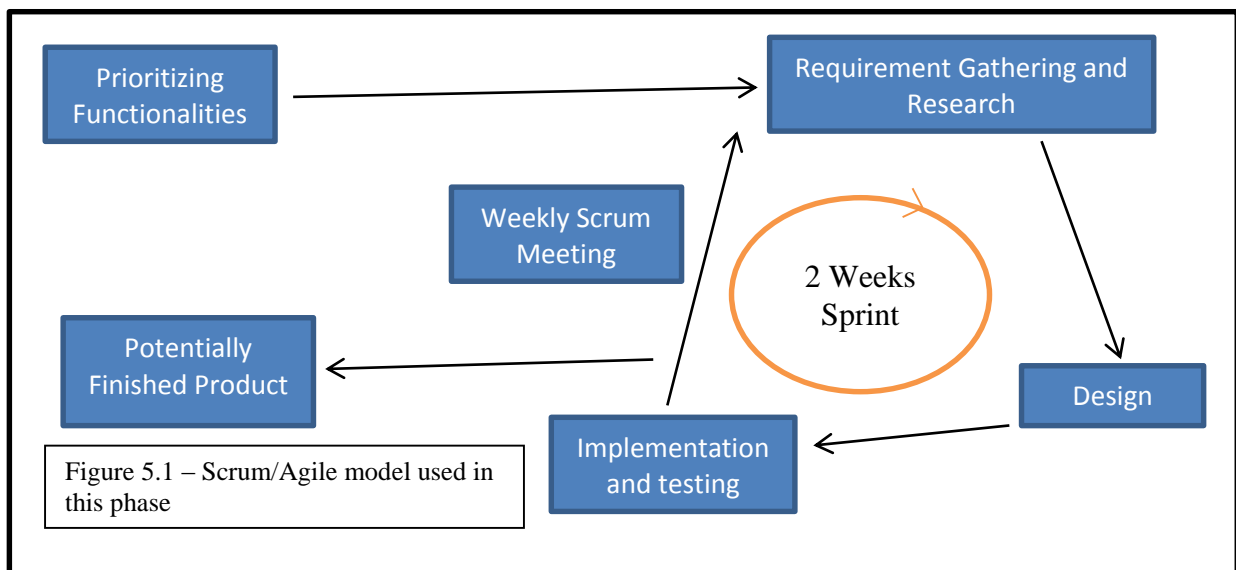
4.5 What is the best way to test the application?

Prior to this project, I used Genymotion Android emulator. This emulator, while faster than the default emulator packed inside the Android SDK is still quite slow when compared to real android device. Moreover, none of the emulators are able to provide 100% proper emulation of a phone's camera which make barcode scanning very difficult. As a result of these issues, I decided to use a real android device for testing the application. Since I don't have any android device, the company provides several older devices donated by several other employees.

5. Methodology

This chapter gives explanation over the methodology used during Project Implementation phase. Project Implementation phase begins right after completing the Research discussed earlier in Chapter 4. The main purpose of this phase is to produce a working end product as soon as possible. Since Realtime Solutions was not sure how much functionalities of the handheld terminal can be replicated by the app, we decided to use agile methodology for the implementation of the project. Using agile, we could develop the app in small increments to determine the feasibility of the mobile devices to replicate certain functionalities of the handheld terminal.

As shown on 'Figure 5.1' below, the project was developed in 'sprints' with each sprint last for 2 weeks. The rest of this chapter will explain each activities listed in the diagram below.



5.1 Prioritizing Functionalities

At the beginning of this phase, a short list of prioritized functionalities are constructed from existing information on the project plan and meetings with the product owner. This is a MOSCOW-like list that determines the order and priority of the functionalities that needs to be implemented in the following sprint.

5.2 Requirement Gathering and Research

After receiving lists of required functionalities, the next step would be to gather information about what is needed to achieve such functionalities followed closely with a research about how to implement it. All of these activities are done with a combination of internet search and consultation with the supporting developers.

5.3 Design

As mentioned earlier in this document, the project leader and the qualified supporting developer were responsible for specifying user interface design and interaction of the app. The 'design' activities conducted here strictly relates to the program structure and logic which are needed to implement the specified user interface design.

5.4 Implementation and Testing

The design is implemented during this activities. After implementation, the program is compiled and installed on mobile devices supplied by the Realtime Solutions for testing. It is also installed on the project leader's phone so that he can test it further while I'm busy working on the other activities. Any bugs or errors encountered were documented and initialized as 'issues' in the company's issue and project tracking software.

5.5 Weekly Scrum Meeting

At the beginning of each week, a scrum meeting with the other project member was held to track the current status of the project. First, each project member communicates every activities related to the project that were done in the previous week. Then, we had discussions about the current issues which are not fixed yet. Finally, we decided on what to do for the rest of the week and update the list of prioritized functionalities as necessary.

5.6 Potentially Finished Product

Finally, at the end of the project a potentially finished product would be released for demonstration to Realtime Solutions client as a proof of concept.

6. Project Implementation Phase

This final phase of the project was aimed to produce a working end product which is suitable for public demonstration during Realtime Solutions 10th year anniversary on June 2015. This chapter of the report was created to give a better understanding over some important activities conducted during the Project Implementation phase.

6.1 Adding More Framework

In order to make the page dynamic in jQuery mobile, the elements that need to be changed must have a unique id. Then an event that is going to modify the element(s) on certain condition must be added to the Javascript file included in the HTML component.

```
<div data-role="collapsible" data-content-theme="e" id="collapsePlace">
<h3 id='h3Text'>Place:</h3>
<!--things...-->
```

Figure 6.1.1 (HTML snippet) changing a HTML element with jQuery Mobile

```
$('#collapsePlace #h3Text').text('new text');
```

Figure 6.1.2 (Javascript snippet) changing a HTML element with jQuery Mobile

It doesn't look that bad, however you will soon run out of idea to uniquely name your elements and your Javascript code will become a tangled mess of jQuery events real quick. Code snippets below demonstrate such problem.

```
<div data-role="main" class="ui-content">
  <a href="#" id="addProduct" data-role="button">Add New Product</a> <br>
  <a href="#" id="addLocation" data-role="button">Add Location</a> <br>
  <a href="#" id="newStock" data-role="button">New Stock</a><br>
  <a href="#" id="editStock" data-role="button">Edit Stock</a><br>
  <a href="#" id="moveCarrier" data-role="button">Move Carrier</a><br>
  <a href="#" id="moveStock" data-role="button">Move Stock</a><br>
  <a href="#" id="pickStock" data-role="button">Pick Stock</a><br>
</div>
```

Figure 6.1.3 (HTML Snippet) clickable menu list with jQuery Mobile

```
document.getElementById('addProduct').addEventListener('click', , function() {
    $.mobile.changePage("./pages/add_product.html");
    }, false);
document.getElementById('addLocation').addEventListener('click', , function() {
    $.mobile.changePage("./pages/add_location.html");
    }, false);
document.getElementById('newStock').addEventListener('click', , function() {
    $.mobile.changePage("./pages/new_stock.html");
    }, false);
document.getElementById('editStock').addEventListener('click', , function() {
    $.mobile.changePage("./pages/edit_stock.html");
    }, false);
document.getElementById('moveCarrier').addEventListener('click', , function() {
    $.mobile.changePage("./pages/move_carrier.html");
    }, false);
document.getElementById('moveStock').addEventListener('click', , function() {
    $.mobile.changePage("./pages/move_stock.html");
    }, false);
document.getElementById('pickStock').addEventListener('click', , function() {
    $.mobile.changePage("./pages/pick_stock.html");
    }, false);
```

Figure 6.1.4 (Javascript Snippet) clickable menu list with jQuery Mobile

The solution to this problem is to use another framework that will add more object oriented structure and supports data binding alongside jQuery mobile. Data binding is the process that establishes connection between the HTML user interface and its Javascript business logic that will determine what is shown in the user interface.

6.1.1 Backbone.js or Knockout.js

There are two known Javascript frameworks that can give proper structure to jQuery Mobile, they are: Backbone.js and Knockout.js. Both are designed to structure Javascript intensive web application to give better control and synchronization of the Javascript functions with the HTML element of the application [20].

Backbone.js

As its name suggests, backbone gives structure to web applications by providing Models, Views and Collections architecture (not to be confused with Model View Controller or MVC mentioned in the earlier chapter). According to the official documentation, 'Models' and 'View' has the following functions [21]:

Models	Views
<ul style="list-style-type: none">• Orchestrates data and business logic.• Loads and saves from the server.• Emits events when data changes.	<ul style="list-style-type: none">• Listens for changes and renders UI.• Handles user input and interactivity.• Sends captured input to the model.

Meanwhile, 'Collections' deal with a group of related models, handling the loading and saving of new models to the server and providing helper functions for performing aggregations or computations against a list of models.

Using Backbone.js would require an understanding of this architecture and other syntax associated with this framework.

Pros	Cons
<ul style="list-style-type: none">• Stiffer/stronger structure via Model-View-Collections architecture• Many built in functionalities	<ul style="list-style-type: none">• Complex with quite steep learning curve• Documentation is not very clear

Knockout.js

Knockout.js is a javascript framework based on Model View Viewmodel (MVVM) architecture. According to its official documentation, 'Model' is the application stored data, 'View' is the visible user interface (the HTML file) and 'Viewmodel' is a pure code representation of the app's data and user interface [22].

In practice, there is little distinction between 'Model' and 'Viewmodel' knockout.js is not concerned over where the 'Model'/'Viewmodel' comes from, it focuses mainly on maintaining the data binding between the object value stated in the 'Model'/'Viewmodel' with the one in the HTML document('View').

Pros	Cons
<ul style="list-style-type: none"> • Simple to use • Great documentation • More flexible 	<ul style="list-style-type: none"> • Focus mainly on data binding.

The following screenshot compares the implementation of backbone.js and knockout.js for a simple HTML document that shows a text ‘Hello Jane!’

```

<script id="myTemplate" type="text/template">
  <input type="text" id="textInput">
  Hello <span id="name"><%=foo.get('name') %></span>!
</script>

<div id="myView"></div>

var myModel = new Backbone.Model({name: 'Jane'});
var MyView = Backbone.View.extend({
  el: '#myView',
  model: myModel,
  events: {
    'input #textInput': 'inputUpdate'
  },
  inputUpdate: function () {
    // update the model when the value of the input element has changed
    myModel.set('name', $('#textInput').val());
  },
  initialize: function () {
    // update the input element when the model value has changed
    myModel.on('change:name', function () {
      $('#name').html($('#textInput').val());
    });
    this.render();
  },
  render: function () {
    var template = _.template($('#myTemplate').html(), {foo: this.model});
    this.$el.html(template);
  }
});

new MyView();

```

Figure 6.1.1.1 Backbone.js example snippet for ‘Hello Jane!’

```

<input type="text" data-bind="value: myName">
<div>Hello <span data-bind="text: myName"></span>!</div>

<script>
  // define the model
  var model = {
    myName: ko.observable('Jane')
  };

  // start Knockout
  ko.applyBindings(model);
</script>

```

Figure 6.1.1.2 Knockout.js example snippet for ‘Hello Jane!’

6.1.2 Result of Previous Comparison

Backbone.js offers better structure and is potentially more powerful. However it has steeper learning curve and mediocre documentation. Since I was under the pressure to produce a working application in a couple of months, I chose knockout.js instead of the backbone.js due to its low learning curve and great documentation.

6.2 Using Require.js

Nowadays, javascript is used to create increasingly complex web application. Large amount of javascript files that are dependent on one another quickly creates ‘script hell’ inside one’s web application.

```
<!-- Include the jQuery library -->
<script src="js/vendor/jquery-1.11.2.min.js"></script>
<script type="text/javascript" src="js/stopDefaultPage.js"></script>
<!-- Include the jQuery Mobile library -->
<script src="js/vendor/jquery.mobile-1.4.5.min.js"></script>
<script src="js/vendor/jquery.mobile.custom.min.js"></script>
<link type="text/css" href="css/vendor/jquery.mobile-1.4.5.min.css" rel="stylesheet"/>
<link type="text/css" href="css/vendor/jquery.mobile.custom.structure.min.css" rel="stylesheet"/>
<link type="text/css" href="css/vendor/jquery.mobile.custom.theme.min.css" rel="stylesheet"/>
<link rel="stylesheet" type="text/css" href="css/vendor/jquery.mobile.iscrollview.css" />
<link rel="stylesheet" type="text/css" href="css/vendor/jquery.mobile.iscrollview-pull.css" />
<link rel="stylesheet" type="text/css" href="css/index.css" />
<link rel="stylesheet" href="lib/StylingRTS/css/main.css">
<script type="text/javascript" src="js/vendor/iscroll.js"></script>
<script type="text/javascript" src="js/vendor/jquery.mobile.iscrollview.js"></script>
<script type="text/javascript" src="js/vendor/knockout-3.2.0.js"></script>
<script type="text/javascript" src="js/vendor/knockout-postbox.min.js"></script>
<script type="text/javascript" src="js/vendor/knockout.validation.min.js"></script>
<script type="text/javascript" src="js/vendor/knockout.mapping.js"></script>
<!-- Include the moment.js for date formatting-->
<script type="text/javascript" src="js/vendor/moment.js"></script>
<script type="text/javascript" src="js/vendor/i18next-1.7.7.min.js"></script>
<script type="text/javascript" src="cordova.js"></script>
<!-- Include the insomnia.js to stop sleep mode-->
<script type="text/javascript" src="js/vendor/Insomnia.js"></script>
<script type="text/javascript" src="js/index.js"></script>
<script type="text/javascript" src="js/scanBarcode.js"></script>
</head>
<body class="app"> |
```

Figure 6.2.1 ‘script hell’ inside an HTML file

Require.js is a javascript file and module loader that uses javascript Asynchronous Module Definition (AMD) API. AMD API specifies mechanism for defining script as ‘module’ and handles their dependency. Consider the following code snippets for clearer explanation:

```
<script data-main="js/common" src="js/vendor/require.js"></script>
</head>
<body class="app">
```

Figure 6.2.2 Snippets from the previous example with require.js

```

requirejs.config({
  baseUrl: 'js',
  paths: {
    "jquery": 'vendor/jquery-1.11.2.min',
    "jquery-mobile": 'vendor/jquery.mobile.custom.min',
    "knockout": 'vendor/knockout-3.2.0',
    "knockout-validation": 'vendor/knockout.validation.min',
    "knockout-postbox": 'vendor/knockout-postbox.min',
    "i18next": 'vendor/i18next.amd.withJQuery-1.8.0.min',
    "addRemoveStockModel": 'viewmodels/AddRemoveStockModel',
    "headerModel": 'viewmodels/HeaderModel',
    "loginModel": 'viewmodels/LoginModel',
    "menuModel": 'viewmodels/MenuModel',
    "navigationModel": 'viewmodels/NavigationModel',
    "orderPickingModel": 'viewmodels/OrderPickingModel',
    "sendRequestModel": 'viewmodels/SendRequestModel',
    "undirectedInventoryModel": 'viewmodels/UndirectedInventoryModel',
    "settingsModel": 'viewmodels/SettingsModel'
  }
});

```

Figure 6.2.3 inside common.js: Declares all javascript files in the requires.config to create short alias for a javascript files. The javascript files could be a third party library or your own javascript files

```

define(["knockout", "knockout-postbox", "knockout-validation", "jquery", "jquery-mobile", "i18next", "helper"], function
(ko, validation, jquery, jqm, i18n, shared) {
  return function() {
    //codes to do something here
  };
});

```

Figure 6.2.4 example snippets to create a module. All required javascript libraries/custom files are called and loaded through its short alias declared earlier in Figure 5.2.2.

```

require(["knockout", "knockout-postbox", "knockout-validation", "jquery", "jquery-mobile", "i18next", "helper"], function
(ko, validation, jquery, jqm, i18n, shared) {
  //do something
});

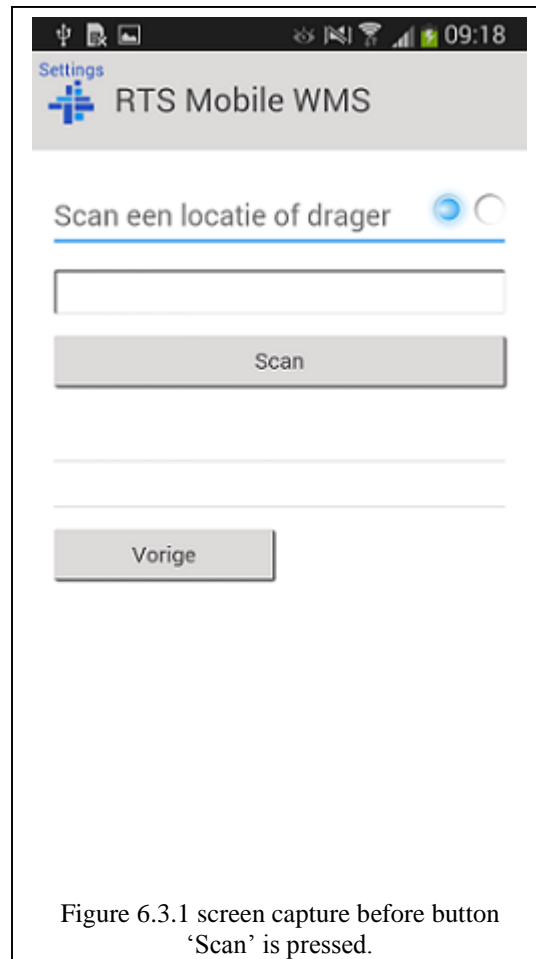
```

Figure 6.2.5 example snippets to call modules. Load all modules listed within the bracket (["modulename"]), then execute the code inside the function.

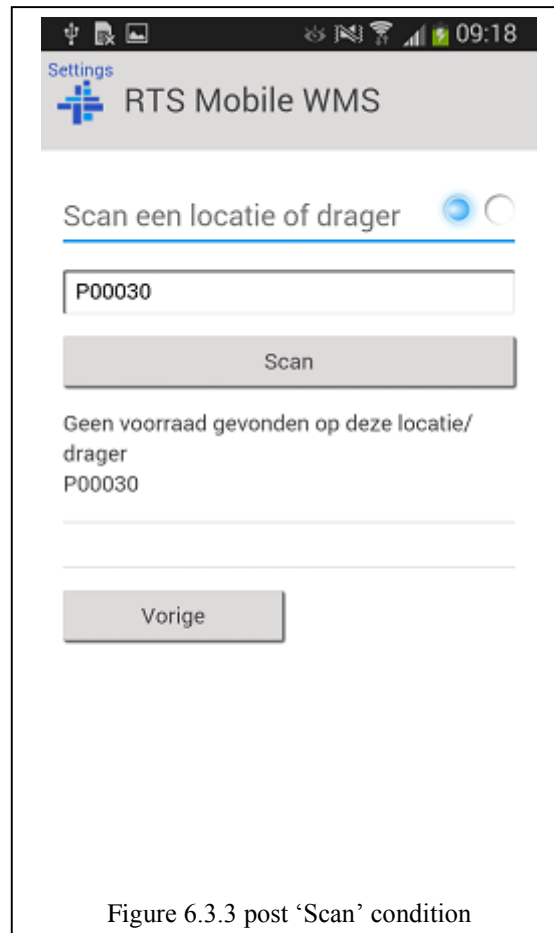
Before using require.js, I wrote all my javascript codes in a single javascript file. The file grows rapidly and the number of lines reach around 2000. The large amount of lines makes it hard to work with the code, especially to detect and fix bugs within the code. With require.js, I managed to separate that single javascript file into 12 modules with an average of 170 lines of code per module.

6.3 Using External Barcode Scanner

By default, the mobile application is started in 'Portrait' orientation as this is the standard and most logical way to show user interfaces on mobile phones application that have a lot of form elements.

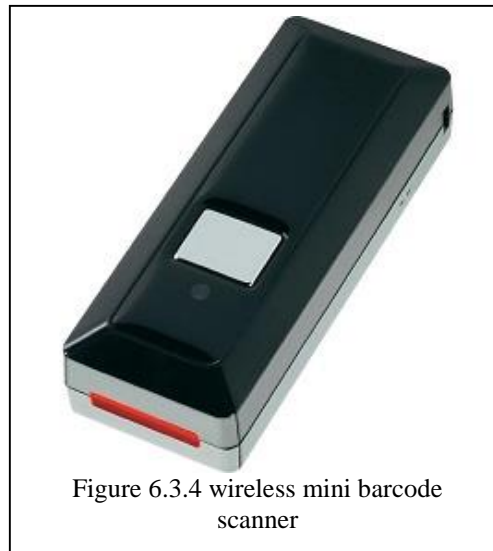


When the function to scan barcode is called, it automatically shows the scanner viewfinder with 'Landscape' orientation. After the barcode is scanned or if scan is cancelled, it closes the scanner viewfinder and shows the user interface back with 'Portrait' orientation.



This behaviour, while manageable for a single barcode scan becomes the source of annoyance to the user that has to scan barcodes several times during a single use case scenario.

After multiple tests during the weekly meeting, it becomes clear that there is no other way to disable/adjust this behaviour within the web component of the app. Therefore, the Product Owner decided to integrate an external mini barcode scanner into the application.



These mini barcode scanner communicates with the phone via Bluetooth connection. To make the integration of this external hardware easier, I set this device to connect with Human Interface Device (HID) Standard. HID is standard protocol used to connect input devices (mouse, keyboard, etc.) to a computer.



On this mode, the phone will treats data received from the scanner like it would when user presses the keyboard. In other words, when the scanner successfully scans a barcode, it returns the text values of the barcode which are then handled in the same way when a user types these text values from his/her keyboard.

7. Final Product User Interface

This chapter describes all of the pages available in the app. Each page is an HTML file supported with CSS and javascript. A set of screenshot and explanation will elaborate the functionalities of each page and its response on expected user interaction.

7.1 Index Page

This will be the first HTML page loaded by the application. It consists of 3 screens: 'Splash', 'Login', and 'Menu'.

7.1.1 'Splash' Screen

Based on the user data stored in the phone, the app will determine which screen to show first. If no data exists, 'Login' screen will be displayed. When valid data are found, the app will skip 'Login' and loads 'Menu' screen. If something goes wrong, it will show the empty 'splash' screen instead.

7.1.2 'Login' Screen

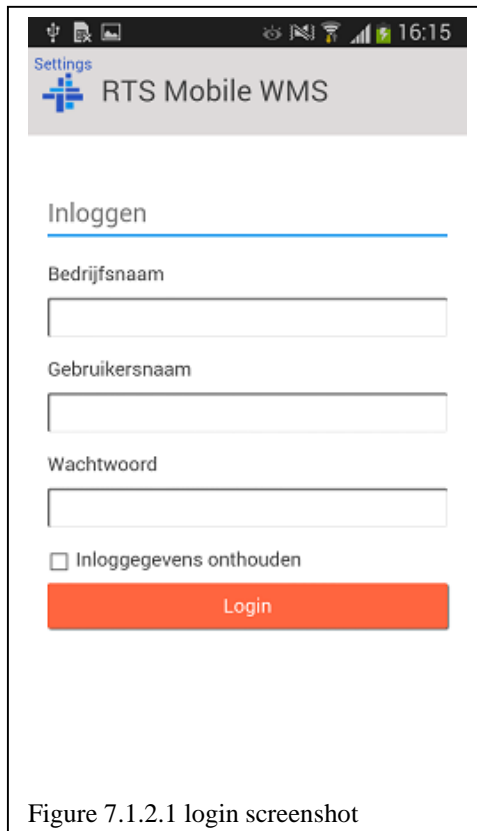
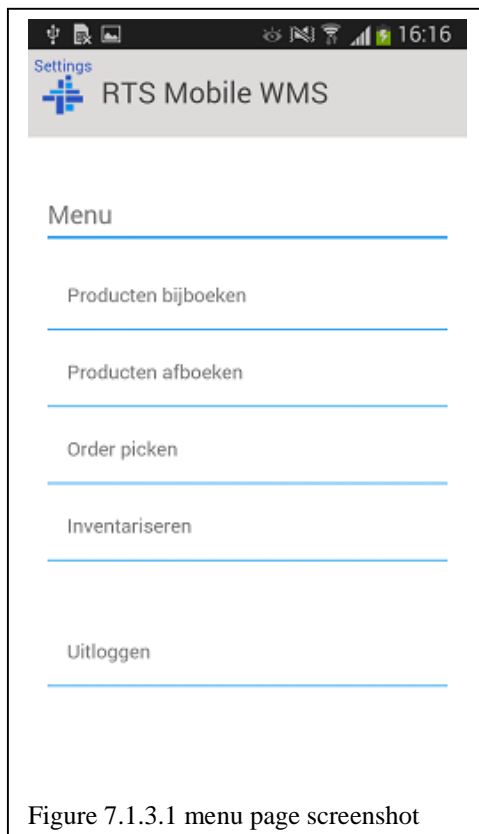


Figure 7.1.2.1 login screenshot

This is the first page that is shown when user starts the app for the first time. Like many other forms, user will enter his login information in the provided fields, optionally check the checkbox 'Inloggegevens onthouden' to keep his/her login information, and press the 'Login' button to check the credentials. When correct credentials are entered, user can access the main menu and starts using the app.

When the user decides to keep his/her login information, this screen will not appear again until he/she logs out of the application.

7.1.3 'Menu' Screen



This is the main menu of the application. All available functionalities are listed here. Pressing the function name in the list will bring user to the appropriate screen. Pressing 'Uitloggen' will logs user out and show the login screen. To exit the app, user simply presses the 'back' or 'menu' button on his/her phone.

Figure 7.1.3.1 menu page screenshot

7.2 'Add Stock' Page

Clicking the 'Product bijboeken' in the 'Menu' screen will bring user to the 'Add Stock' page. This page consists of 4 screens, they are: 'Scan location/container', 'Scan product', 'Add product', and 'Scan location only'.

7.2.1 'Scan location/container' Screen

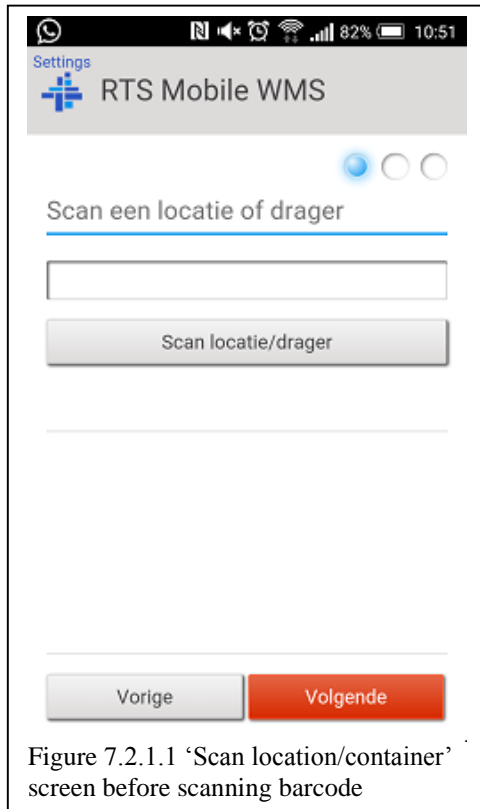


Figure 7.2.1.1 'Scan location/container' screen before scanning barcode

'Scan location/ container' is the first screen that is loaded when user clicks 'Product bijboeken'. An indicator at the top right corner of the screen's body inform the user of the amount of steps he/she has to perform. To proceed further, user must scan a location or container barcode. This can be achieved manually by typing down the barcode alphanumerical value or location ID in the input field and press the 'go' button in the keyboard afterwards. To scan a barcode using external barcode scanner, click on the input field so that the keyboard appears. Then scan the barcode with the external barcode scanner.

Alternatively, user can also scan the barcode with the phone's camera by pressing the 'Scan locatie/drager' button. This action will bring the scanner viewfinder as shown below:



Figure 7.2.1.2 scanner viewfinder

Ensure that the whole barcode is positioned inside the bright rectangle and the red line in the middle of the barcode. When a barcode is scanned, the screen returns to 'Scan location/container'.

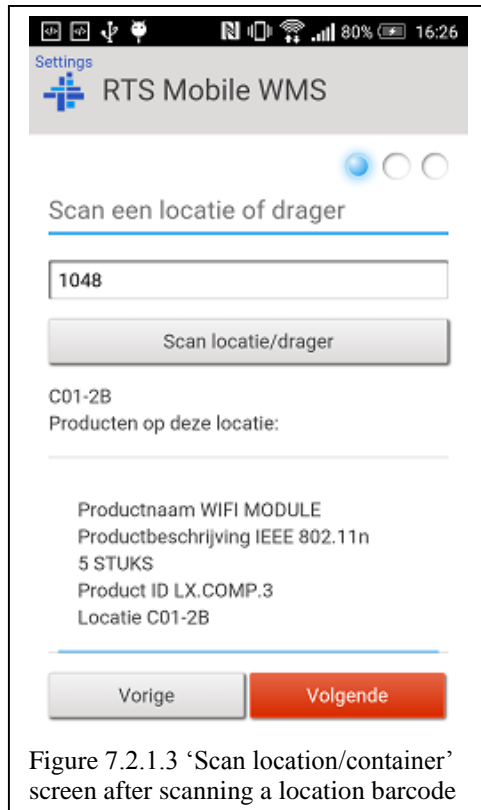


Figure 7.2.1.3 'Scan location/container' screen after scanning a location barcode

Regardless of the scan method, the 'Scan location/container' screen will display information about the barcode/location ID afterwards. If a location/container contains product, a scrollable product list will appear on screen as shown in the screenshot below.

Due to the restriction imposed by the database, a container must have a location information. If user scans an empty container, he will need to do an additional step 'Scan location only' to get the required location information. A container with product may skip the additional steps because a container is only allowed to hold 1 product and product always have a location information.

Pressing the button 'Vorige' will bring user back to main menu, while pressing the button 'Volgende' will bring user to 'Scan product screen'. Alternatively, user can select the product from the list and goes directly to 'Add product' screen.

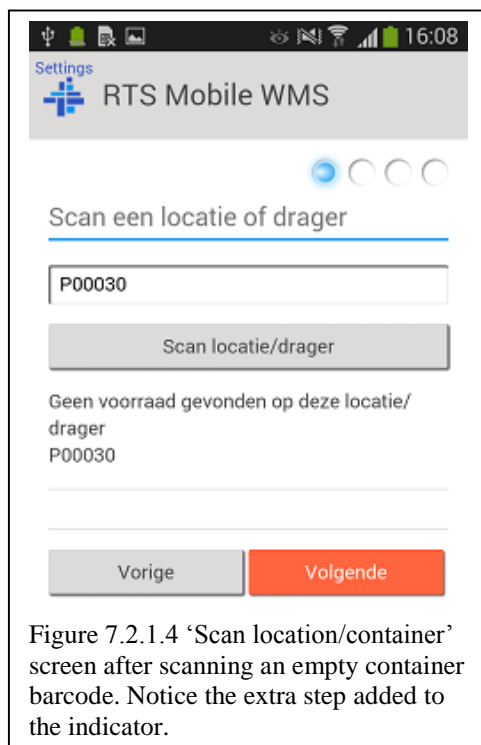
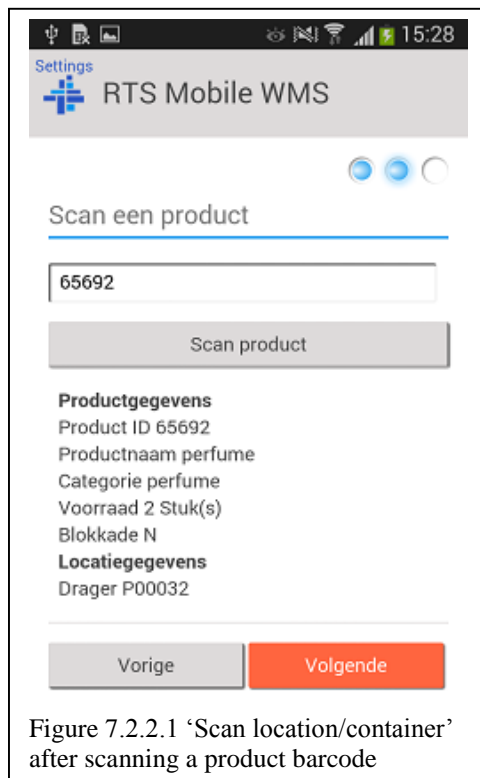


Figure 7.2.1.4 'Scan location/container' screen after scanning an empty container barcode. Notice the extra step added to the indicator.

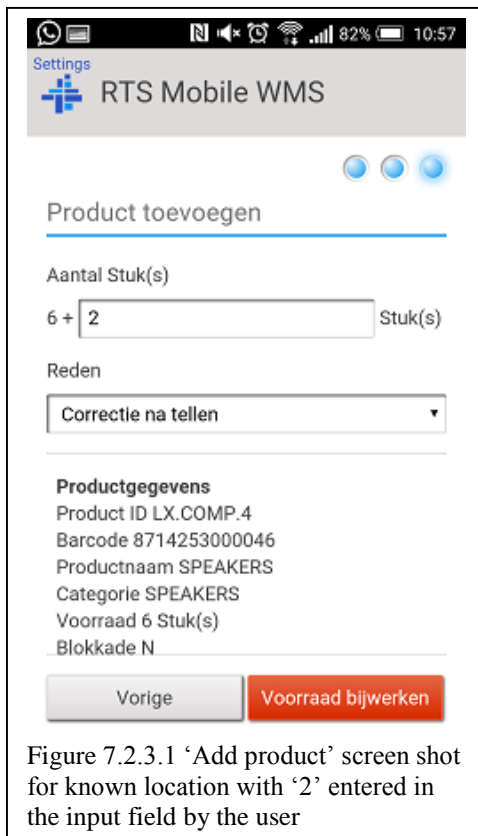
7.2.2 'Scan product' Screen



This is the screen where user scan a product barcode. Product is scanned with the same method as the 'Scan location/container' described previously on chapter 6.2.1. Pressing 'Vorige' on the screen or 'back' button on the user's phone will bring him/her back to the previous page mentioned in chapter 6.2.1. Pressing 'Volgende' will bring user to the 'Add product' screen.

Figure 7.2.2.1 'Scan location/container' after scanning a product barcode

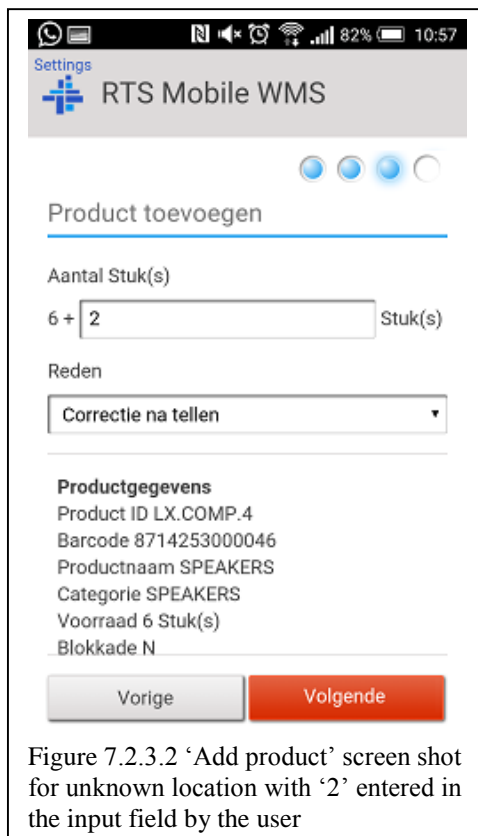
7.2.3 'Add product' Screen



As mentioned earlier, this would be the last step for location and container with known location. Pressing 'Vorige' on the screen or the 'back' button on user's phone will bring the user to the previous screen. If user skips 'Scan product' by selecting a product from the list of scanned product described in chapter 6.2.1, then he/she will be returned to the 'Scan location/container' screen. Otherwise, user will be returned to the 'Scan product' screen.

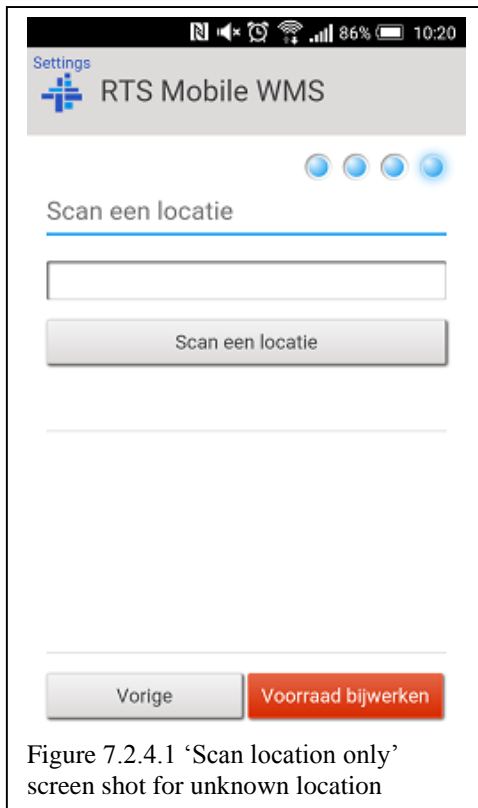
Pressing the 'Voorraad bijwerken' on the screen or 'go' button in the keyboard while the input field is active will update the amount of current product in the database, reset all screen in this page and bring user back to 'Scan location/container' screen.

For container with unknown location, the 'Add product' screen will look and response differently. The figure below shows the slight appearance difference for container with unknown location.



Pressing the 'Volgende' on the screen or 'go' button in the keyboard while the input field is active will bring user to the next screen 'Scan location only'. The rest of the screen will behave the same way as mentioned earlier.

7.2.4 'Scan location only' Screen



This screen looks and behaves similarly to the 'Scan location/container' screen described in chapter 6.2.1. The main difference are:

- It only accepts location barcode/ID as valid input
- Pressing 'Vorige' on the screen or the 'back' button on user's phone will bring the user to the 'Add product' screen
- Pressing the 'Voorraad bijwerken' on the screen after scanning/entering valid location, the system will update the amount of current product in the database, reset all screen in this page and bring user back to 'Scan location/container' screen.

7.3 'Remove Stock' Page

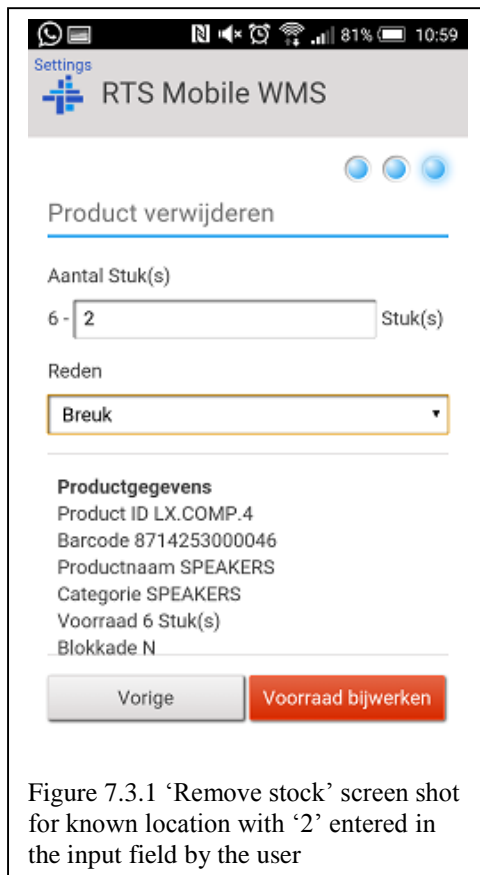


Figure 7.3.1 'Remove stock' screen shot for known location with '2' entered in the input field by the user

Clicking the 'Product afboeken' will bring user to the 'Remove Stock' page. This page consists of 4 screens, they are: 'Scan location/container', 'Scan product', 'Remove stock', and 'Scan location only'.

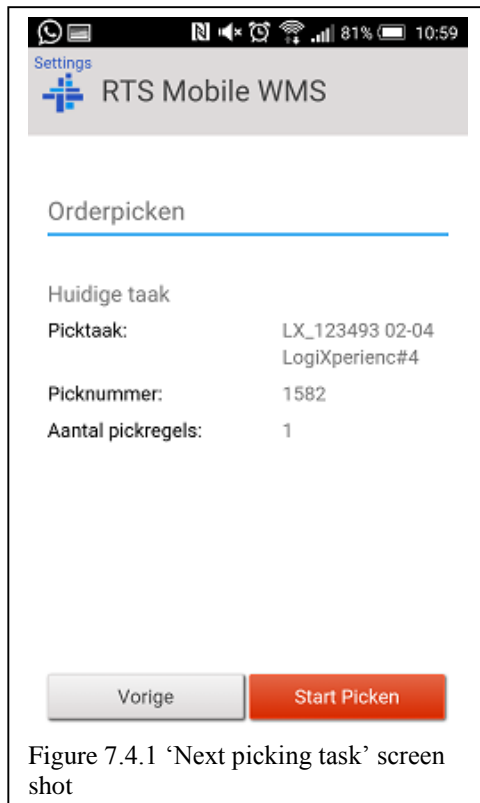
This page is highly similar to the page described in chapter 6.2 and for the most part uses the same screen and user interaction. The only difference is it used a 'Remove stock' screen instead of 'Add product' screen used by the previous page.

The 'Remove stock' screen is also just a copy of 'Add product' screen. As long as user interface and interaction is concerned, the only difference is that it has '-' operator near the input field instead of '+' and different sets of 'Reden' ('Reasons') of reducing the amount of product in the database.

7.4 'Order Picking' Page

This is the page that will be loaded when user clicks 'Order picken' in the 'Menu' screen. This page consists of 3 screens, they are: 'Next picking task', 'Next picking lines' and 'Print document'.

7.4.1 'Next picking task' Screen

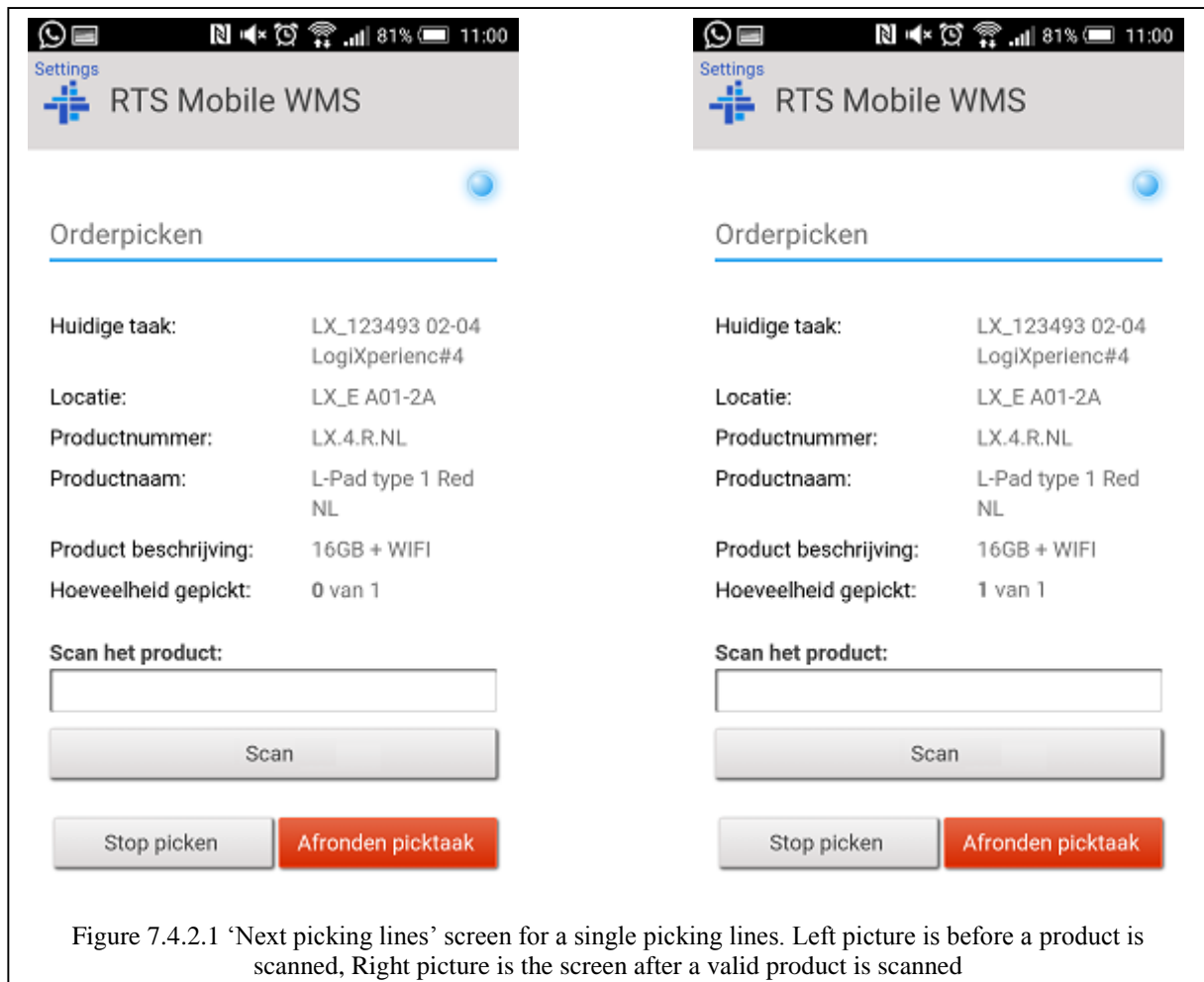


As soon as the page is loaded, this will be the first screen displayed to the user. The screen shows the current picking task available for the user. Pressing 'Vorige' on the screen or 'back' button on user's phone will bring him/her back to the 'Menu' screen. Pressing 'Start picken' will take the user to the 'Next picking lines' screen to start the picking activity.

Figure 7.4.1 'Next picking task' screen shot

7.4.2 'Next picking lines' Screen

Picking task can have more than one picking line. The following screen will appear if a picking task only contains a single picking line.



Product is scanned by the phone's camera or entered through the input field either manually or with external barcode scanner. The scanning procedure is the same as explained in chapter 6.2.1. Each time a valid product is scanned, the number of 'Hoeveelheid gepickt' ('Amount picked') will increase until the required level is reached. If invalid product is scanned or a valid product is scanned when the 'Amount picked' has reached its required level, an appropriate error message will be shown.

Regardless of the 'Amount picked', pressing 'Afronden picktaak' will bring the user to the 'Print document' screen. On the other hand, pressing 'Stop picken' on the screen or 'back' button on the phone while the user is on this page will return the user to the 'Menu' screen. For multiple picking lines, consider the following screen shot and explanation.

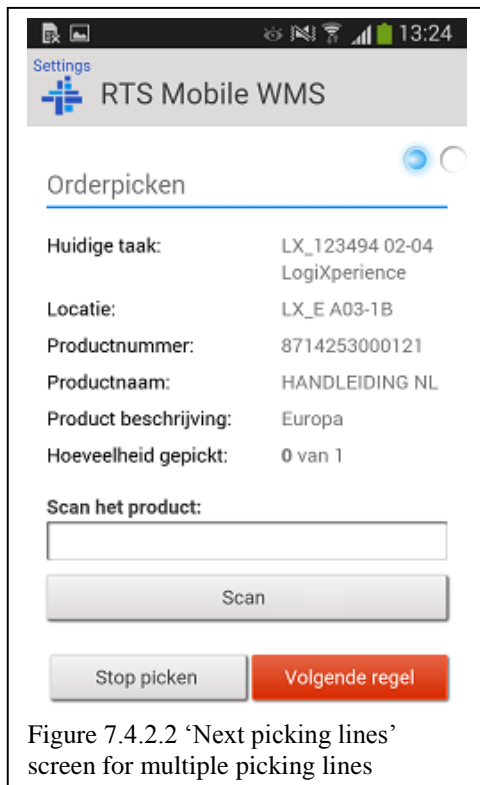


Figure 7.4.2.2 'Next picking lines' screen for multiple picking lines

There is no change in 'Stop picken' and 'back' button behaviour. The main difference is the 'Volgende regel' button on the screen. Pressing it will update the 'Next picking lines' screen with the next picking line as demonstrated by the screenshot below.

When the last picking line is reached like shown on figure 6.4.2.3 above, the 'Volgende regel' button will change into 'Afronden picktaak'. As explained earlier, pressing this button will bring user to the 'Print document' screen.

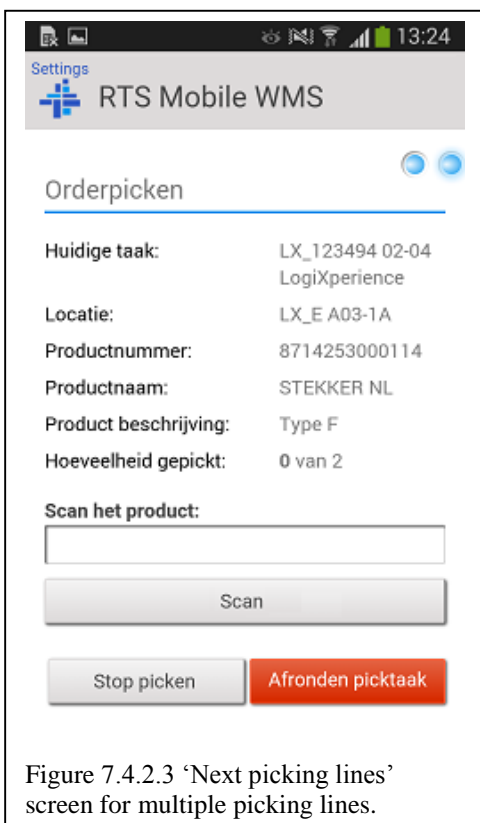
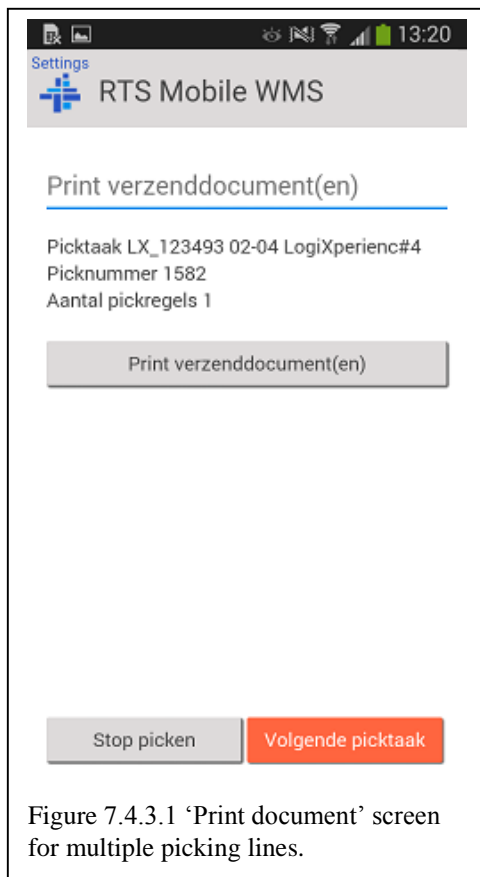


Figure 7.4.2.3 'Next picking lines' screen for multiple picking lines.

7.4.3 'Print document' Screen



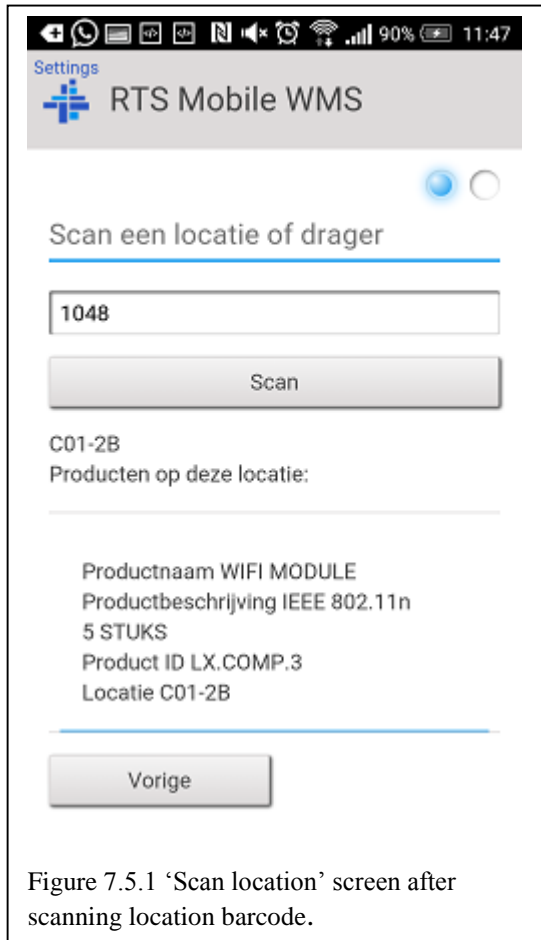
This is the last screen in the 'Order picking' page. Pressing the 'Print verzenddocument(en)' will send a print job to the nearest printer that will print the shipping document related to the picking task. Pressing 'Stop picken' on the screen or 'back' button on the phone will return user to the 'Menu' screen. Pressing 'Volgende picktaak' will bring user back to the 'Next picking task' screen for further picking task.

Figure 7.4.3.1 'Print document' screen for multiple picking lines.

7.5 'Inventory' Page

Pressing the 'Inventariseren' button in the 'Menu' screen will bring user to this page. This page consists of two screens, 'Scan location' and 'Confirm stock'.

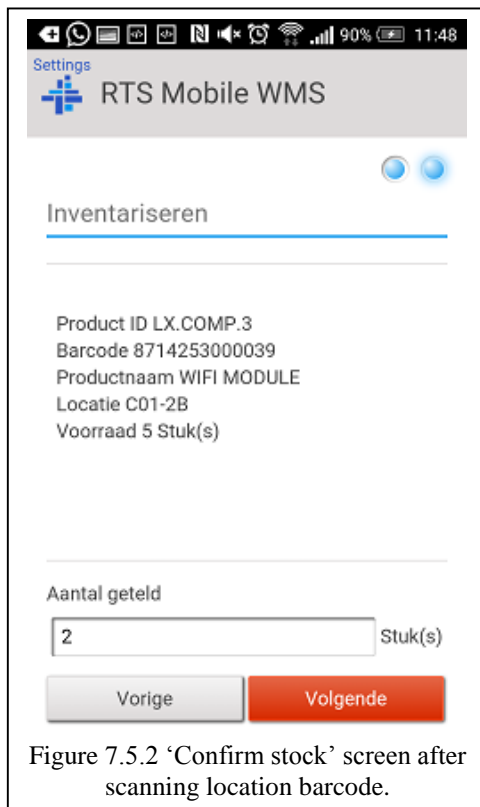
7.5.1 'Scan location' Screen



This screen use the same method to scan barcode that were explained in detail on chapter 6.2.1. To proceed to the next screen, user must select a product from the list of product found on a location/container. Pressing 'Vorige' on the screen or 'back' button on the phone will bring user back to the 'Main' screen. When a product from the list is selected, user will be brought to the next screen 'Confirm stock'.

Figure 7.5.1 'Scan location' screen after scanning location barcode.

7.5.2 'Confirm stock' Screen



Pressing 'Volgende' on the screen or 'go' button in the keyboard while the input field is active will change the stock level of the current product according to the value that user entered in the input field 'Aantal geteld'. Moreover, the app would also bring user back to the 'Scan location' screen with the updated information of the previously scanned location.

Pressing the 'Vorige' on the screen or 'back' button will return the user to the 'Scan location' screen without any change in the data displayed previously.

7.6 'Settings' Page

At the corner of each page, there is a small link/button called 'Settings'. Clicking this link brings user to the 'Settings' page.

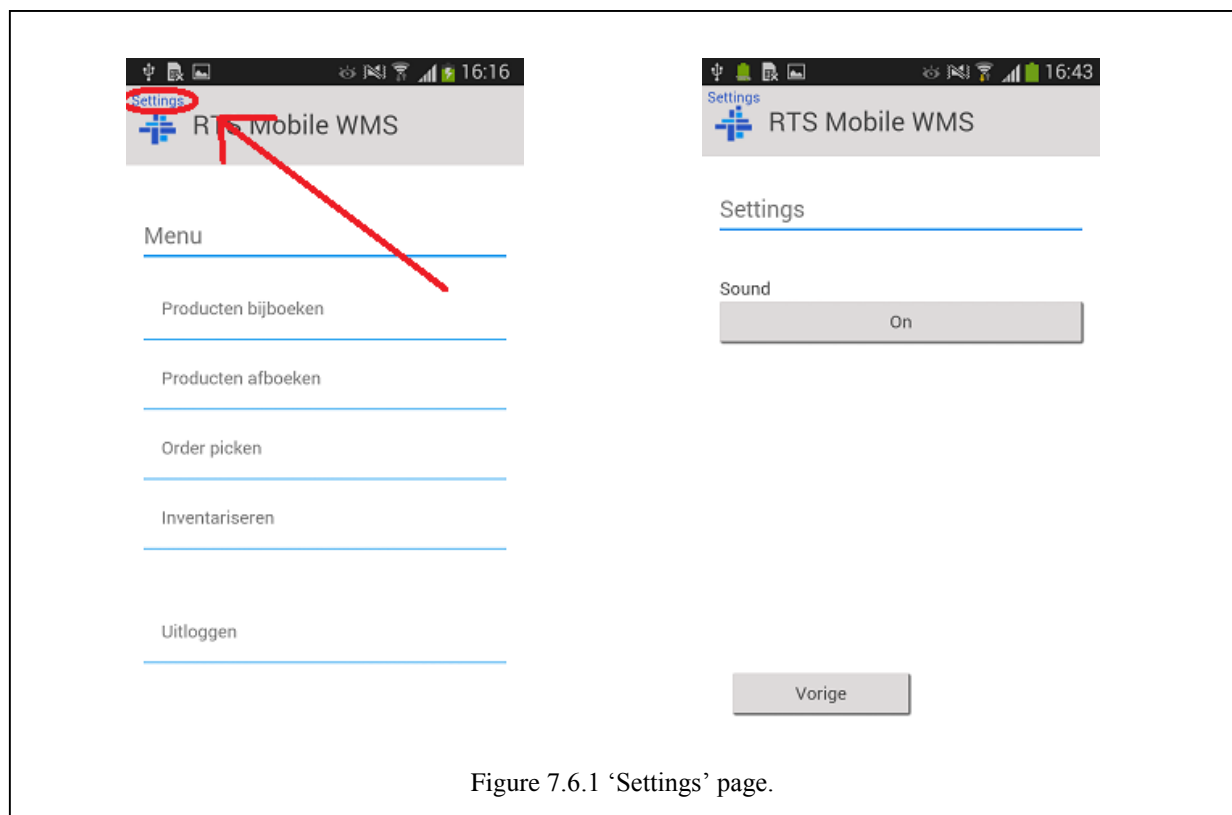


Figure 7.6.1 'Settings' page.

At the moment, only 'Sound' settings are available. Pressing the button located just underneath the 'Sound' text will toggle sound in the app on or off. By default, this value is set to 'on'. Setting the value to 'off' will prevent the app from sounding a tone each time it displays error/notification message.

Pressing 'Vorige' on the screen, or 'back' button on the phone will bring user back to the page where he/she at before opening the 'Settings' page. The same behaviour can also be triggered by pressing the 'Settings' button/link while the user is already at the 'Settings' page.

8. Final Product Main Features

This chapter explain in details about the main features of the app and the way it is implemented at the moment this report is written.

8.1 Server Connection

As mentioned in the beginning of this document, the app will leave most of the business logic/calculation to the WMS database. The app communicates itself to the WMS database with HTTP through the Realtime Solutions cloud server.

The HTTP request is send with the 'ajax()' method implemented within jQuery mobile framework. AJAX (Asynchronous Javascript and XML) is a web development technique to create a better web application by asynchronously (in the background) send and retrieve data from the server. The 'ajax()' method within jQuery mobile handles both sending the data and receiving the reply from the server.

A request is sent each time user wants to use the app functionalities. For example, when a location barcode is scanned, user authorization data and the text value of the barcode is send to the server. The server will check the validity of the user authorization data, if authorization data is valid, server will send the appropriate information to the WMS database. WMS database will check the location and returned a reply to the app through the server.

Request sent always contains a user authorization data, timestamp and correlation ID. User authorization data is created by combining the user's company ID, username and password. Timestamp that shows exact time and date of the request is created with a moment.js library while correlation ID are hardcoded for each functionalities.

The reply that user receives will vary depending on the data he/she sends. This reply consists of 2 parts, a message and optionally a data. The message could be an error message or a success message. If expected data is returned, it will be displayed on the app.

8.2 Account and Settings

As mentioned earlier, a user account is used to create a unique identifier which will be included in the request send to the cloud server. User only need to enter his login credentials once because it is stored in a special object from knockout.js library that will publish its value when needed elsewhere in the app.

The same goes for settings, another special object holds Boolean value to indicate the status of a functionality. Before the related function is executed, it will check this value and acts accordingly. However, these information will be lost when user closes (exit) the app completely. The constant need to login each time the app starts is a big source of annoyance to some user.

The solution to this problem is to store the information locally in the user's phone. When user starts the app again, the data stored from the user's phone will be used to initialize these special object values.

The data are stored locally via phonegap's webStorage API. This API implements the web standard API of the same name. webStorage stores data inside the app's webview cache. There are two variation of webStorage, sessionStorage and localStorage. sessionStorage is destroyed when the browser used to display the page is closed, this means the data will be destroyed when user exits the app. On the other hand, localStorage is persistent and will only loses its value when programmatically ordered to do so. As a result, user authorization data and settings are stored within the localStorage object.

localStorage implementation provides some security to the data stored with:

- Private webview cache – each application that uses localStorage stores data on its own webview cache. This means, other app should not be able to access this cache through normal means [23].
- Same origin policy – a URL was issued to all pages available in the app. Only a page with the same URL address origin may access the localStorage data [24].

8.3 Barcode Scanning

Barcodes can be scanned in three way: scan barcode with the phone's camera, manual user input or with the external barcode scanner. Scanning barcode with the phone's camera is achieved by using the default Phonegap barcode scanner plugin. Calling the method 'scan()' will bring the scanner viewfinder while pressing the 'back' button on the phone will cancel the scan and close the scanner viewfinder. Each time the method 'scan()' is called, a callback function will catch the result. If a barcode is successfully scanned a barcode, it will returns a success message, the text value of the barcode and the barcode type.

When manual user input or external barcode scanner is used, the text value entered in the input field will be sent directly to the server. Thus, the barcode scanner plugin is not used at all in this mode.

9. Deployment

The final result will be demonstrated in ‘Innovatie Middag’ on 19th of June 2015 as part of the company’s 10th anniversary. At the time this report was written, a set of test database, mock products to physically represent the test database and test plan is being meticulously prepared by the project manager. Apart from the public demonstration at the event, Realtime Solutions also planned to test the app for real world usage on a number of selected client’s business environment. If the test results prove to be positive, Realtime Solutions will continue the development of this project further.

10. Conclusions

WMS is an important part of modern supply chain management. It is used mainly by medium and large companies to manage their warehouse operation. Realtime WMS, the main service offered by Realtime Solutions is one example of such system. Due to the high cost associated with its implementation, only medium and large companies can afford the Realtime WMS [1]. Meanwhile, a market research conducted within Realtime Solutions suggests that there is also a sizeable profit in a new market segment of small scale WMS implementation for smaller companies [1].

Realtime Solutions realized that replacing the hand held terminal used by workers in the clients' warehouse with something cheaper is the best way to minimize the cost of Realtime WMS implementation [1]. Thus, this project is launched to create a prototype of the mobile application that will substitutes hand held terminal for small scale Realtime WMS implementation. The result of this project will be demonstrated on the company's 10th anniversary celebration in June 2015.

During researches conducted within the project, I discovered that there are many ways to create a mobile application. I chose to create a hybrid mobile application based on HTML, CSS and javascript to make use of the company's experience in these sectors and fulfil the requirement for a platform agnostic mobile application. I also faced many framework options to achieve the desired functionalities and on each occasion I chose the most popular/easiest of the bunch. I made such decision because I find the deadline to be quite tight. By using the most popular/easiest frameworks, I can reduce the time required to study a framework before using it and get more time to implement the required functionalities.

11. Recommendations

This project was aimed to produce a proof of concept that can be demonstrated in time for the company's 10th anniversary event. As such, it is not developed with the best program design structure. Should Realtime Solutions decides to continue this project in the future, I would like to recommend two possible options:

- A. Continue using the current set up (jQuery mobile + knockout.js + require.js) and improve it by separating the code further into smaller reusable components (widgets) and restructure the code to form an MVC-like structure.
- B. Refactor the app to use Angular.JS. This way, we rely on the MVC structure and widgets standard that are included within the framework to ensures a good program structure.

Regardless of the options chosen, improving the structure of the app will improve its maintainability and expandability.

Furthermore, I would also recommend to change the information stored in Account and Settings features. As mentioned earlier, user credentials are currently stored in the user's device via localStorage implementation. It will be a good idea to replace the credentials with a session token generated by the server instead. This way, user will not need to change his credentials if there is a security breach in the user's device.

After improving the structure and security of the app, the next step would be to release the app to the store by following the official guidelines available on the internet [25] [26] [27].

Evaluations

Working at Realtime Solutions has been a pleasant experience for me. I got a chance to work on an interesting mobile development project in a supportive, constructive and relatively stress-free work environment and culture. At the beginning of this project, I had absolutely no idea where to start because this is the first real project that I initiated from the very beginning and actually has a say on it.

With the support from other project member, I slowly pieced together the technical requirements and boundaries of this project. Then, I did many other research related to these requirements and finally I had enough information and started working on the project.

If the deadline of this project wasn't so tight, I would build the project from the beginning with Angular.js framework. Alas, I had to make do with the fastest and easiest solution to produce a working application as soon as possible because I was not willing to risk missing the deadline. Nor did anyone at Realtime Solutions, as they had released news articles about the upcoming anniversary celebration on 19th of June and promised the demo of the mobile app there.

The program structure of the final product is not perfect, however I am glad that I was able to finish the app with an acceptable code quality in time for demonstration on the anniversary celebration. I have no doubt that the experience and skills that I got from this project will go for a long way in supporting my future career as a software developer.

Reference

- [1] Statement from Harm van den Broek
Founder and current owner of Realtime Solutions
- [2] <http://www.realtimesolutions.nl/nl/10-over/32-ons-bedrijf>
Official Realtime Solutions company website.
- [3] http://www.lionbridge.com/files/2012/11/Lionbridge-WP_MobileApps2.pdf
A slideshow from Lionbridge, a company that provides translation, online marketing and content managements.
- [4] <http://mashable.com/2012/09/12/web-vs-native-apps/>
An article posted on Mashable.com, a tech news information site.
- [5] <http://developer.telerik.com/featured/the-state-of-hybrid-mobile-development/>
Opinion from active member posted on Telerik site for developers. Telerik is the company behind Kendo UI
- [6] <http://www.developereconomics.com/pros-cons-top-5-cross-platform-tools/>
A website that publish information related to mobile development.
- [7] <http://stackoverflow.com/questions/26648378/phonegap-basic-architecture>
- [8] http://developer.xamarin.com/guides/cross-platform/application_fundamentals/building_cross_platform_applications/part_1_-_understanding_the_xamarin_mobile_platform/
Official Xamarin documentation
- [9] <http://www.appcelerator.com/blog/2012/05/comparing-titanium-and-phonegap/>
A blog entry in the official Appcelerator forum. Appcelerator is the maker of the Titanium framework
- [10] <http://www.gajotres.net/top-7-mobile-application-html5-frameworks/>
A blog entry from a software developer that is quite active in mobile development forums
- [11] <http://www.gajotres.net/sencha-touch-vs-jquery-mobile/>
A blog entry that compares Sencha Touch with jQuery Mobile from a mobile application developer with good online presence.
- [12] <http://www.gajotres.net/best-html5-mobile-app-frameworks-sencha-touch/>
Another entry from the same author about sencha touch
- [13] <http://www.gajotres.net/best-html5-mobile-app-frameworks-famo-us-review/>
Another entry from the same author about famo.us framework
- [14] <https://en.wikipedia.org/wiki/AngularJS>
Wikipedia article about AngularJS.

- [15] <http://www.gajotres.net/top-7-mobile-application-html5-frameworks/2/>
Another entry from the same author about Ionic framework.
- [16] <http://www.gajotres.net/top-7-mobile-application-html5-frameworks/5/>
Another entry from the same author about Onsen UI framework.
- [17] <http://www.gajotres.net/top-7-mobile-application-html5-frameworks/3/>
Another entry from the same author about jQuery Mobile framework.
- [18] <http://www.gajotres.net/top-7-mobile-application-html5-frameworks/4/>
Another entry from the same author about Kendo UI framework.
- [19] <http://stackoverflow.com/tags>
Famous Q&A sites for programmer/developer. From this link, I made a personal research to determine which framework has the highest userbase by calculating the amount of question about it at Stack Overflow. The results at the time this report is written are:
1. jQuery Mobile – 21296 questions.
 2. Kendo UI – 9143 questions.
 3. Ionic – 1544 questions
 4. Onsen UI – 487 questions
 5. Famo.us – 508 questions
- Angular JS has around 99516 questions, however it is not included for comparison above because it is not a specialized mobile framework. Even when it is included anyway, the amount of jQuery tagged questions (620061 questions) still beats it by a significant margin.
- [20] <http://speckyboy.com/2014/10/14/backbone-vs-knockout/>
An article from Speckyboy online magazine for web designer.
- [21] <http://backbonejs.org/#Getting-started>
Official backbone.js documentation.
- [22] http://knockoutjs.com/documentation/observables.html#mvvm_and_view_models
Official knockout.js documentation.
- [23] <http://developer.android.com/reference/android/webkit/WebView.html>
Official android developer guide about webView.
- [24] <http://www.w3.org/TR/webstorage/>
Official W3C publications about webStorage. W3C is international standards organization for the web.
- [25] <http://developer.android.com/distribute/tools/launch-checklist.html>
Official android documentation for placing an app in the Google Play.
- [26] <https://developer.apple.com/library/ios/documentation/IDEs/Conceptual/AppDistributionGuide/Introduction/Introduction.html>
Official android documentation for placing an app in the App Store.

- [27] <https://dev.windows.com/en-us/publish>
Official Windows documentation for placing an app in the Windows Store.
- [28] https://en.wikipedia.org/wiki/Application_programming_interface
Wikipedia page about application programming interface.
- [29] https://en.wikipedia.org/wiki/Software_framework
Wikipedia page about software framework.
- [30] https://en.wikipedia.org/wiki/Android_software_development
Wikipedia page about Android software development.
- [31] https://en.wikipedia.org/wiki/Plug-in_%28computing%29
Wikipedia article about plugin.
- [32] <https://developer.apple.com/library/mac/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>
Official Apple developer manual

Appendix A: Project Survey



University of Applied Sciences

Specialisation:

Data student:

Name student

ICT & : Software engineering / Business /Technology , BMT, other:

: Initials: Y.W.

First name: Yohandi

Name: Yohandi Widjaja

Studentnumber:2184980

Telephone: 064-7607822

E-mail.: y.widjaja@student.fontys.nl

Data company:

Name company/organisation : Real Time Solutions B.V.

Visiting adress : Torenallee 20, 5617BC Eindhoven

Company mentor :

Initials: H.B.

Telephone: 065-1341918

Name: Harm van den Broek

E-mail.: info@realtimesolutions.nl

Department/ position: Owner

Startdate Internship : 02-02-2015

Duo Internship : No

If duo name of buddy: _____

Accepted by student: date:

signature:

Accepted by company: date:

signature:

Hand in date Graduation Project Survey:

Approved by graduation project coordinator: yes/no

date:

signature:

Remarks : _____

Description of the graduation project:**1. Describe the problem analysis:**

(what is the reason for the internship company to initiate this assignment? What is for the company the added value of this assignment? Can you describe the starting situation and starting points: introduction and problem definition.

A mobile application for warehouse management system is demanded by some customer to replace the aging mobile terminal they currently use. Also creating a mobile application will open up the possibility for more functionality that mobile terminal could not provide(e.g. Features that use image, gps location data, etc.).

2. Describe the graduation assignment.

(especially the objectives, results to be delivered and final products to be realized. Also indicate what you want to achieve for the internship company. Give a clear description of the graduation assignment).

Create a mobile application for the company's client (a dashboard) which are connected to the company's cloud server. The application should implement all functionalities currently available in the legacy system along with additional features that previously was impossible to implement in the legacy system(e.g. Features that use Image, GPS location data, etc.).

Since clients have different requirements, The application should also be easy to customize and extend to fulfil client's specification/needs.

3. What is the research component of this assignment?

(What are your research topics? If necessary, draw up a research plan).

There are several think needs to be researched:

1. How to develop mobile application? (Especially what kind of tools are required to do so)
2. How to integrate the mobile application to the company 's cloud server?
3. How to design a customizable/extendable mobile application?(which design pattern works best)
4. How to design a secure mobile application?

4. What are the methods and tools?

(What operating procedure and means can you made use of during the internship period at the company? What facilities will be made available by the company?

No operating procedure/method is specified.

Company will provide a smart phone/tablet to test the application and a Linux workstation may be provided if necessary.

The assignment will be developed on my own laptop since I'm more familiar with Windows than Linux. A cloud server is also available for use with mobile application. Further tools will be added as part of the research component of this project(find a suitable tools to develop a mobile apps)

5. How and by whom will you be guided by the company, which other 'stakeholders' are involved?

Mr. Harm van den Broek will tutor me for the general part of the assignment. The other programmers will help me on more technical questions.

6. What fields of Study play an important factor in realizing the graduation assignment?
(for example information analysis, design, realization, monitoring and security).

Research: Required to determine the best way to integrate the mobile application to the company's cloud server and creates an expendable/maintainable design.

Information gathering/analysis : Required to find out how to implement the application

Programming: Required to realize the assignment.

Security: Required for secure connection between the mobile application and the server

OTHER DETAILS:

Preference university tutor:	1. Bert van Gestel
	2. Jack Zijlmans

PLEASE SEND THIS FORM BY EMAIL TO THE GRADUATION COORDINATOR IMMEDIATELY AFTER THE GRADUATION INTERVIEW HAS TAKEN PLACE.

Appendix B: Project Plan

Project Plan

FONTYS UNIVERSITY OF APPLIED SCIENCES

HBO-ICT: English Stream

Data student:	
Family name , initials:	Widjaja, Y
Student number:	2184980
project period: (from – till)	February 2015- June 2015
Data company:	
Name company/institution:	Real Time Solutions
Address:	Torenallee 20 5617BC Eindhoven
Company tutor:	
Family name, initials:	van den Broek, H
Position:	Owner
University tutor:	
Family name , initials:	Zijlmans, J

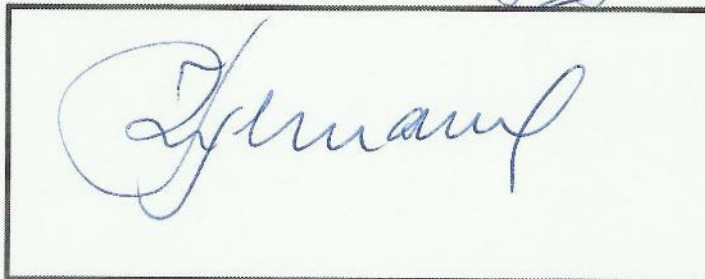
Approved and signed by the company tutor:

Date: 6-3-2015

Signature:

H. VAN DEN BROEK


School tutor:
ir. JHA. Zijlmans



Contents

1. Introduction.....	VIII
2. Project Statement	VIII
2.1 Product Owner	VIII
2.2 Project Leader	VIII
2.3 Lead Developer.....	VIII
2.4 Supporting Developer	IX
2.5 Current Situation.....	IX
2.6 Project Justification.....	IX
2.7 Project Product.....	X
2.8 Project deliverables and non-deliverables	X
2.8.1 Project deliverables	X
2.8.2 Project non-deliverables.....	X
2.9 Research.....	XI
2.10 Project risk	XI
3. Project Phasing.....	XII
Phase 1: Start up	XII
Phase 2: Version 2.....	XII
Phase 3: Version 3.....	XIII
Phase 4: Version 4.....	XIII
Phase 5: Version 5.....	XIII
Phase 6: Version 6.....	XIV
Phase 7: Version 7.....	XIV
Phase 8: Version 8.....	XIV
Phase 9: Presentable Version	XV
Phase 10: Finalization	XV
4. Management plan.....	XVI
4.1 Money	XVI
4.2 Skill	XVI
4.3 Quality.....	XVI
4.4 Information	XVII
4.5 Time	XVII
4.6 Organization.....	XVIII

4.6.1 Project Organization	XVIII
4.6.2 Contacts.....	XVIII
5. Communication plan.....	XIX
5.1 Interested parties in the project.....	XIX
5.2 Means of Communication.....	XIX

1. Introduction

This document describes the detailed plan of my internship project at Real Time Solutions.

The project is to create a mobile application for Warehouse Management System (WMS) dashboard. This project is the first mobile application project at Real Time Solutions and will be created from scratch. The mobile app will use the same cloud server that is currently used for the Realtime WMS web console to communicate with the WMS-database. The results of this project will be used to complement the current WMS which use a web console for standard browsers and a telnet application for hand held terminals. During the project, I will be guided by a project manager – who is a Fontys student as well – and supported by 3 other developers of Realtime Solutions.

My internship project will last for approximately 100 working days. I will spend at least 8 hours per day on this project to make sure that my project stays on track. To give the overview about this project, everything regarding the project continuation such as the project schedule, possible risks, deliverables, and communication plan are described in this project plan.

2. Project Statement

2.1 Product Owner

The product owner is Harm van den Broek from Realtime Solutions. He specifies the requirement and goal of this project. He will also give feedback during the project so that this project will produce a product that suits his needs.

2.2 Project Leader

Dave Godschalx, Another student from Fontys will take the official role of ‘Project Leader’ to manage the non-technical aspect of this project. The project leader has to organize weekly meeting, understand the overview of the project status, and make technical decision together with the Lead Developer.

2.3 Lead Developer

I, Yohandi Widjaja will be the main developer for this project. The lead developer has to determine the technical requirement to achieve the requested functionalities, conduct research on how to design and implement the functionalities, make technical decisions together with project leader and realize the project.

2.4 Supporting Developer

Three experienced developers are assigned by Realtime solutions to help me during the project. Each of them are specialized on a subject that is important to this project. Their name and specialization are listed as follows:

- Sanne Peters – Interaction Designer
- Patrick van Dalfsen – Database Specialist
- Martijn Evers – Cloud Server Specialist

They will provide help and guidance as necessary during the project.

2.5 Current Situation

Currently, Realtime Solutions uses a telnet application on hand held terminals to provide logistical functionality to front line users in the warehouse. The application offers great flexibility, but requires rather expensive hand held terminals in a WIFI network. The prevalence of mobile devices like smart phones, tablet computers or even smart glasses, coupled with the desire to have a low cost logistical functionality that works outside the warehouse and beyond the range of the company's WIFI network, Realtime Solutions has decided to start building applications for these mobile devices.

2.6 Project Justification

Not all company can afford the aforementioned hand held terminal, therefore this project is created to provide a low cost WMS implementation for smaller company/business. Realtime Solutions deemed mobile device is the perfect platform to deliver their low cost WMS platform for this new target group for several reasons:

- Mobile devices are quite common today.
- Mobile devices are small and light enough to operate while working in the warehouse.
- Installing app on mobile device is cheaper and faster than buying/installing hand held terminals and its support systems.
- Cameras present on most mobile devices can be set to read barcode – one of the most important features on a hand held terminal.
- Mobile devices may not offer the flexibility and ergonomics of the hand held terminal, however such issues are irrelevant for small scale WMS application.

In addition, the mobile app will also offer extra functionality to existing customers that use hand held terminal. These new functionalities are made possible due to the hardware that is commonly used on mobile devices. Examples of this extra functionalities are: adding pictures and videos to the data, adding GPS coordinates to data, incorporate smart glass functionality, etc.

2.7 Project Product

An operational version of the WMS Mobile application for Android platform is expected at the end of May. The following use case should be covered by this version:

- Add product – scan barcode of product, add product details, type, and quantity, send data to WMS-database.
- Add location - scan barcode of product, show product details, add current location, add previous location, send data to WMS-database.
- New stock - scan barcode of product, show product details, type quantity, optionally, scan carrier (box or pallet) and/or location, send data to WMS-database.
- Edit stock - scan carrier or location, show stock information, choose edit, type new quantity.
- Move carrier - scan carrier, show stock information, scan new location, send data to WMS database.
- Move stock - scan carrier or location, type quantity, scan new carrier or location, send data to WMS database.
- Pick stock - get next order from WMS database, process line by line: show location, scan product, type quantity, and send data to WMS database.

2.8 Project deliverables and non-deliverables

2.8.1 Project deliverables

Project Deliverables	Expected Date
Project plan	06/03/2015 (Final Version)
Small Prototypes	Every 2 week from the end of February
Presentable Version	19/06/2015
Activity Report	Monthly
Final Report	Monthly, Starting at the beginning of April

2.8.2 Project non-deliverables

Project non-deliverable	Expected Date
Research result	-

2.9 Research

Research is a crucial part of this project as this is the first mobile application project for Realtime Solutions. The following are the research questions I deemed important when I received this project for the first time:

- How to develop mobile application? (Especially what kind of tools are required to do so)
- How to integrate the mobile application to the company's cloud server?
- How to design a customizable/extendable mobile application?(which design pattern works best)
- How to design a secure mobile application?

2.10 Project risk

Risk	Affected area	Impact	Probability	Prevented by	Solution if it occurs
Missing Deadline	All phases	Medium	Medium	Realistic Scheduling	Deal with the consequences
Communication problem with other project member	All phases	Medium	Low	Language skills	Google Translate
Unforeseeable Problem when Implementing functionality	All phases except phase 1	Medium	Medium	Google search, asking other project member	Skip the troublesome functionality

3. Project Phasing

The following section is a rough estimated schedule for my project at Realtime Solutions. This project starts on February 2015 and I expect it to be finished at the end of June 2015.

Phase 1: Start up

#February 2 – February 20, 2015

Estimated time for this phase : 15 days

Total project time spent : 15 days

Deliverable(s) : -

Tasks :

- Get an overview of project in general
Find out some general information that I need for project plan.
- Start working on project plan
Project plan is important to keep the project on the right track. A good project plan will lead to better planning that will affect the whole project success.
- Start working on the project
Discuss with other team member what to do next and in which order, Research to find out what is needed and how to implement it.
- Create first prototype
Create the first version of the application

Phase 2: Version 2

February 23 – March 06, 2015

Estimated time for this phase: 10 days

Total project time spent : 25 days

Deliverable(s) : Second version of the application, Final version of Project Plan

Tasks :

- Testing/evaluation
Check the previous version for bugs
- Improvement
fix bugs and add more functionality
- Research
Do research as necessary to fix bugs/add more functionality

Phase 3: Version 3

March 9-March 20, 2015

Estimated time for this phase: 10 days

Total project time spent : 35 days

Deliverable(s) : Third version of the application

Tasks :

- Testing/evaluation
Check the previous version for bugs
- Improvement
fix bugs and add more functionality
- Research
Do research as necessary to fix bugs/add more functionality

Phase 4: Version 4

March 23-April 3, 2015

Estimated time for this phase: 10 days

Total project time spent : 45 days

Deliverable(s) : 4th version of the application, First version of Final

Report

Tasks :

- Testing/evaluation
Check the previous version for bugs
- Improvement
fix bugs and add more functionality
- Research
Do research as necessary to fix bugs/add more functionality

Phase 5: Version 5

April 6-April 17, 2015

Estimated time for this phase: 10 days

Total project time spent : 55 days

Deliverable(s) : 5th version of the application

Tasks :

- Testing/evaluation
Check the previous version for bugs
- Improvement
fix bugs and add more functionality
- Research
Do research as necessary to fix bugs/add more functionality

Phase 6: Version 6

April 20-May 1, 2015

Estimated time for this phase: 10 days

Total project time spent : 65 days

Deliverable(s) : 6th version of the application, second version of Final

Report

Tasks :

- Testing/evaluation
Check the previous version for bugs
- Improvement
fix bugs and add more functionality
- Research
Do research as necessary to fix bugs/add more functionality

Phase 7: Version 7

May 4-May 15, 2015

Estimated time for this phase: 10 days

Total project time spent : 75 days

Deliverable(s) : 7th version of the application

Tasks :

- Testing/evaluation
Check the previous version for bugs
- Improvement
fix bugs and add more functionality
- Research
Do research as necessary to fix bugs/add more functionality

Phase 8: Version 8

May 18-June 5, 2015

Estimated time for this phase: 15 days

Total project time spent : 90 days

Deliverable(s) : 8th version of the application, third version of Final

Report

Tasks :

- Testing/evaluation
Check the previous version for bugs
- Improvement
fix bugs and add more functionality
- Research
Do research as necessary to fix bugs/add more functionality

Phase 9: Presentable Version

June 8 - June 19, 2015

Estimated time for this phase: 10 days

Total project time spent : 100 days

Deliverable(s) : Presentable version of the application, latest version of Final Report

Tasks :

- Testing/evaluation
Check the previous version for bugs
- Improvement
fix bugs and add more functionality
- Research
Do research as necessary to fix bugs/add more functionality

Phase 10: Finalization

June 22 - June 30, 2015

Estimated time for this phase: 7 days

Total project time spent : 107 days

Deliverable(s) : Latest version of the application

Tasks :

- Testing/evaluation
Check the previous version for bugs
- Improvement
fix bugs and add more functionality
- Research
Do research as necessary to fix bugs/add more functionality

4. Management plan

4.1 Money

No money is involved.

4.2 Skill

Phase name	Skill required
All Phases	Planning, Analysing, Research, Java, XML, HTML, CSS, Javascript

4.3 Quality

The overall quality of the end product can be measured in several aspects. The details of these aspects are listed below.

- Features

A good quality software will cover all feature listed on the requirement document.

- Ease of Use

This is a very subjective factor, thus it's hard to be quantified. A good software should have a good user documentation, clear error messages, good exceptions management and provide some recovery after a failure.

- Maintainability

This is one of the most important aspects. A good quality application will be under version management. In this way we can easily rolling back to the previous version as needed.

- Readability

A good application will has a unified coding style, thus another developer can easily read the code written by another developer.

4.4 Information

	Participant	Project plan	Activity Report	Working Product	Final application
Lead Developer	Yohandi Widjaja	Cr,Ar,S,Di	Cr,Ar,S	Cr, Ar, S,Di	Cr, Ar, S,Di
Product Owner	Harm van den Broek	Di,R,Ar,A	-	Di, R, Ar, A	Di, R, Ar, A
Fontys tutor	Jack Zijlmans	Di,R,Ar,A	Di, R	Di	Di

Indexes:

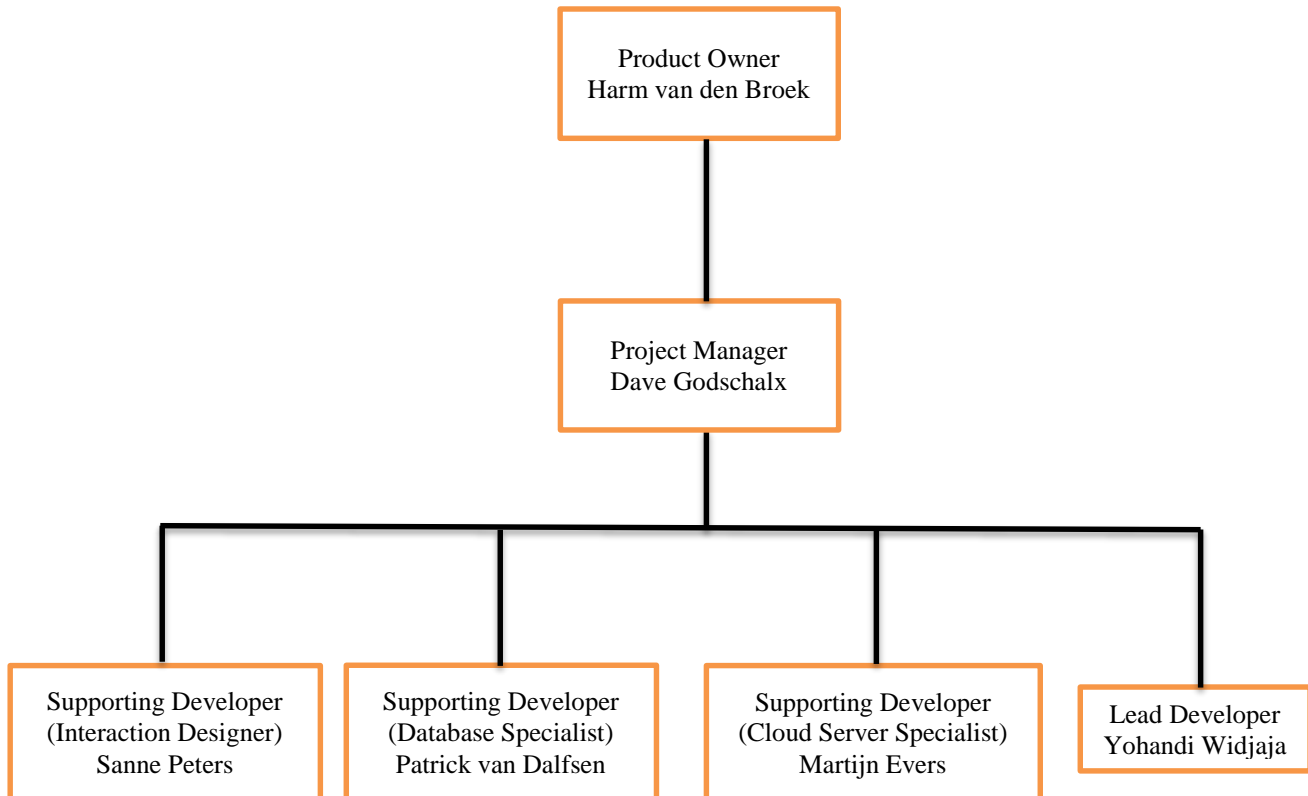
A : Approve
 Ar : Archive
 Cr : Create
 Di : Discuss
 S : Send
 R : Receive

4.5 Time

Period	Phase name	Time	Description
February 2 – February 20, 2015	Start-up	15 days	Preliminary research, First prototype
February 23 – March 06, 2015	Second Version	10 days	Second prototype
March 9-March 20, 2015	Third Version	10 days	Third prototype
March 23-April 3, 2015	4 th Version	10 days	4 th prototype
April 6-April 17, 2015	5 th Version	10 days	5 th prototype
April 20-May 1, 2015	6 th Version	10 days	6 th prototype
May 4-May 15, 2015	7 th Version	10 days	7 th prototype
May 18-June 5, 2015	8 th Version	10 days	8 th prototype
June 8 - June 19, 2015	Presentable Version	15 days	Presentable prototype
June 22 - June 30, 2015	Finalization	7 days	Final version
Total		107 days	

4.6 Organization

4.6.1 Project Organization



4.6.2 Contacts

Lead Developer

Name : Yohandi Widjaja
Telephone : 0647607822
Email : y.widjaja@student.fontys.nl
Address : Hagenkampweg Zuid 143, 5615DC, Eindhoven

Product Owner

Name : Harm van den Broek
Telephone : 0651341918
Email : info@realtimesolutions.nl
Address : Torenallee 20, 5617BC, Eindhoven

School tutor

Name : Jack Zijlmans
Telephone : 0885071207
Email : j.zijlmans@fontys.nl
Address : Fontys Hogescholen, Rachelsmolen 1, Eindhoven

5. Communication plan

5.1 Interested parties in the project

Who	On behalf of	Interest	Means of Communication (*)
Harm van den Broek	Realtime Solutions	Mobile application for WMS Dashboard	Email, Meeting
Jack Zijlmans	Fontys	Project progress	Phone, Email, Meeting
Yohandi Widjaja	Student	Finish the project	-

5.2 Means of Communication

From	To	Information	Medium	Frequency of data
Yohandi Widjaja	Jack Zijlmans	Activity Report	Email	At the end of each month
Yohandi Widjaja	Jack Zijlmans	Final Report Draft	Email	Per month starting on April
Yohandi Widjaja	Jack Zijlmans	Small meeting	Conversation	As needed
Yohandi Widjaja	Harm van den Broek	progress meeting	Conversation	As needed
Yohandi Widjaja	Harm van den Broek, Jack Zijlmans	Final Presentation	Presentation	Once at the end of the internship