

# **Technology Evaluation and Draft Implementation of a RIA Smartphone or Tablet WEB-Browser Solution for a “Mobile Attendant” in Automated Checkout**

Xuejie YU

25<sup>th</sup>, June, 2012

Diploma Thesis

Supervisors:

Prof. Dr.-Ing. Holger Hinrichs  
Department of Electrical Engineering and Computer Science  
Lübeck University of Applied Sciences

Dipl.-Informatiker. Hendrik Franco da Cruz  
Retail Division  
Wincor Nixdorf International GmbH

Department: **Electrical Engineering and Computer Science**

Major: **Information Technology**

Subject: **Technology Evaluation and Draft Implementation of a RIA Smartphone or Tablet WEB-Browser Solution for a “Mobile Attendant” in Automated Checkout**

## **Abstract**

The purpose of this thesis is to evaluate different technologies and find out the suitable ones which can be used to implement a platform-independent Web-Browser-based application for a “Mobile Attendant” (MA) and build a prototype for this application. It mainly concerns three parts: 1. User Interface (UI) of the MA; 2. Monitor the status change of self-checkout machines; 3. Communication between MAs and self-checkout machines.

After comparing different state-of-the-art technologies, Javascript, jQuery, HyperText Markup Language (HTML) 5 and Cascading Style Sheets (CSS) 3 are selected as the main technologies to implement the logical control of different components on a web page and the functions of UI. Experiments shows that UDP[1] multicast cannot be consumed by browsers directly, so a proxy architecture has to be set up which is in order to monitor the status change and receive the information packages which are created when the state of self-checkout machines are changed. Then it forwards the packages to the MA by a novel technology named WebSocket. Besides, the communication between MAs and self-checkout machines is realized by the technology AJAX to consume the Webservice which is provided by the self-checkout machines.

Afterwards, the thesis presents a prototype with an elegant user interface and main functions of the existing fixed attendant station. Finally, it analyzes the results, evaluates the prototype and gives recommendations for the further research.

Keywords: Web application, Smartphone, Tablet, WebSocket, Webservice, AJAX, jQuery, HTML5

# Contents

<b>1. Introduction</b>	<b>5</b>
1.1. Background.....	5
1.2. Objective.....	6
1.3. Structure of the thesis.....	6
<b>2. Foundation</b>	<b>8</b>
2.1. Definition Web Application.....	8
2.2. System Introduction.....	9
2.3. Basic Technologies .....	10
2.4. Tool Selection.....	12
2.4.1. Development Tool.....	12
2.4.2. Debug Tool.....	14
<b>3. Requirement Analysis</b>	<b>15</b>
3.1. Introduction of existing Attendant.....	15
3.1.1. User Interface .....	15
3.1.2. Communication of Mobile Attendant .....	16
3.2. Application Requirement .....	16
3.2.1. Functional Requirement .....	17
3.2.2. Nonfunctional Requirement .....	17
3.2.3. Hardware Requirement .....	17
<b>4. Technical Design</b>	<b>19</b>
4.1. General Architecture.....	19
4.2. Technologies Evaluation .....	20
4.2.1. User Interface Design .....	20
4.2.2. UDP Transmission .....	23
4.2.3. Webservice .....	26
4.2.4. Camera and Barcode .....	27
4.3. Complete Architecture .....	27
4.3.1. The Structure of the System .....	27
4.3.2. User Case Diagram .....	29
4.3.3. Sequence Diagram .....	33
4.3.4. Activity Diagram .....	34
4.3.5. Class Diagram .....	35
<b>5. Implementation</b>	<b>37</b>
5.1. User Interface Design .....	37

5.1.1. Component Introduction .....	37
5.1.2. Key Function Description .....	38
5.2. UDP Communication .....	41
5.2.1. Component Introduction .....	42
5.2.2. Key Function Description .....	42
5.3. Webservice Consumption .....	43
5.3.1. Component Introduction .....	43
5.3.2. Key Function Description .....	44
5.4. Camera and Barcode .....	45
5.4.1. Component Introduction.....	45
5.4.2. Key Function Description.....	45
<b>6. Evaluation and Discussion</b>	<b>47</b>
6.1. Result.....	47
6.1.1. Function Implementation.....	47
6.1.2. Application Test.....	49
6.1.3. Result Analysis.....	50
6.2. Application Evaluation.....	50
6.3. Discussion.....	53
6.3.1. Comparison of the PC-based application and the Web-based application.....	53
6.3.2. Problem Discussion.....	54
<b>7. Summary</b>	<b>55</b>
7.1. Brief view on the Application.....	56
7.2. Recommendation.....	56
<b>List of Figure.....</b>	<b>57</b>
<b>List of Table.....</b>	<b>58</b>
<b>Bibliography.....</b>	<b>59</b>
<b>Glossary .....</b>	<b>61</b>
<b>Appendix A.....</b>	<b>63</b>

# Chapter 1

## Introduction

This chapter will provide the background knowledge of the research. Then it presents the objective of this research and illustrates the structure of the thesis.

### 1.1 Background

After the appearance of first portable handset in 1973, the mobile devices thoroughly influence people's lives. The development of plenty mobile devices, like Pageboy, mobile phone and iPad help people throw off the restriction of wires and push people to the mobile life. As the great success of Apple and Google, the iPhone, iPad and Android devices are spreading all over the world. On the train, at the office, in the street, even in bed, using mobile devices is becoming more and more common scenario.

On the other hand, the striking development of the mobile communication and Internet brings the unprecedented opportunities for the popularization of the mobile life. It is unimaginable that there are already one third people in the world are involved in the business. Furthermore, with the release of HTML5 and CSS3, the web applications enjoy a new spring as they laid a solid foundation for the development of the web applications by the convenient APIs those can access the local resources. As the remarkable merits of the web application like the absence of installation and widely adaptation to different platforms, nearly all the professionals agree with that within next decades, the web applications will have the edge over the local applications. It seems that the transforming in web and mobile is widely acknowledged and unstoppable.

Wincor Nixdorf's software solution for Automated Checkout in Retail, named TPiSCAN, is widely used in retail area like supermarkets and drugstores. The self-service terminals are usually accompanied by a supervisor terminal where store associates can provide help to customers. A mobile version exists that is based on mobile Java. With new Tablet PCs and Smartphone and their new software stacks hitting the consumer market, Wincor Nixdorf seeks for a new platform-independent solution which is based on web browser and acquires the necessary functions of the mobile attendant. Also, the solution shall make heavy use of

available client logic in order to avoid the typical “sluggish” usability of fat-server-thin-client WEB application. Furthermore, the solution should allow for the invocation of local camera devices to read 1D and 2D barcodes on the mobile attendant devices.

## 1.2 Objective

The purpose of this thesis is to evaluate different technologies and find out the suitable ones those can be used to implement a platform-independent Web-Browser-based application for a “Mobile Attendant” and build a prototype for this application. It concerns about the programming in web page and its background functions. It should use the HTML, CSS and JavaScript as the main programming languages. The HTML and CSS will mainly support the User Interface of the application. The JavaScript will support the functions of the application. The design of the prototype should refer to the fixed existing attendant stations and realize most of the functions of the fixed existing Attendant with some fashionable concepts. The research should provide the solutions for the challenges listed below:

- Find a suitable mechanism feasible with Javascript to inform an MA that the status of a monitored self-checkout machine has changed and needs to be re-pulled.
- Listen to the status changes and visualizing the status of several self-checkout systems to realize the parallelism.
- Communicate with the self-checkout machines via existing WebService interface.
- Find a way to analyze received XML files by SAX or DOM in Javascript.
- Find a way to implement responses to attendant intervention status on self-checkout machines which makes the MA active and reactive.
- Scan the barcodes by the camera in Javascript and evaluate the platform independence.
- Find out whether there is a price to pay if the application shall also run well in a traditional WEB browser on a PC.
- Find out the suitable tools which support the development effectively.

## 1.3 Structure of the thesis

Figure 1.3.1 shows the structure of the thesis.

Chapter 1 introduces the background and objective of the research.

Chapter 2 introduces the web application, the self-checkout system, basic technologies and basic tools which will be used in this research.

Chapter 3 analyses the requirements from UI and communication aspects.

Chapter 4 designs the general architecture to guide the technology evaluation. Then it builds a complete architecture according to the evaluation result.

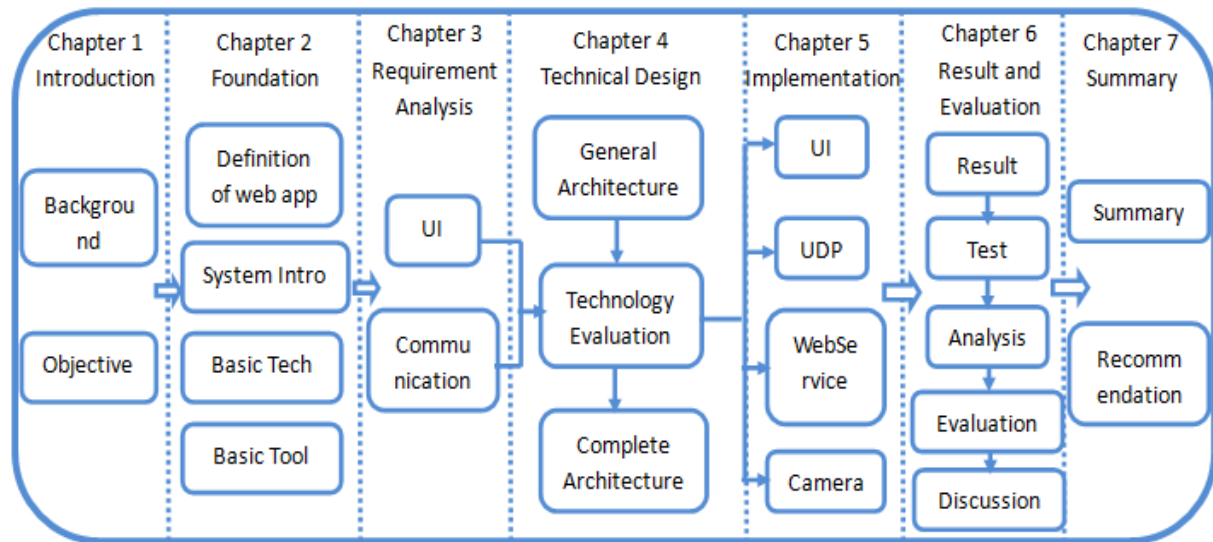


Figure 1.3.1 Structure of the Thesis

Chapter 5 implements the functions of UI, UDP communication, Webservice and camera.

Chapter 6 presents the result. Then it tests the application and analyzes it. It also evaluates the application and makes some discussions on it.

Chapter 7 summarizes the research and gives some recommendations on further research.

# Chapter 2

## Foundation

This chapter will include the foundational knowledge about the research. “What is web application” will be answered first. Then it will introduce the system to give a brief view on the functions of each component. It will introduce the basic technologies which are the base of the research and select the proper tools which facilitate the research.

### 2.1 Definition of Web Application

“Web applications use the web browser as a way of connecting to networked tools and systems. They are stored on web servers, and use tools like databases, JavaScript (or Ajax or Silverlight), and PHP (or ASP.Net) to deliver experiences beyond the standard web page or web form.” [JK1].

Compared with the local application, the web application has several distinct advantages:

- ① The absence of installation. A web application can be accessed by a URL. The users just need to type in the address of the application and then they can enjoy the functions of the application.
- ② The ubiquity of web browsers. For the modern personal computers, Smartphones and other mobile devices like Tablet, the browser is one of the most fundamental components. The omnipresence of the browser endues the web applications with the strong viability. When there is a web browser, the web applications can be run.
- ③ The convenience of update and maintenance. A web application is deployed to the web server only once, instead of several times to each user’s devices. In a distributed system, a same application may distribute on different devices. When the application is updated, all the application on these devices should be updated. However, for the web application, once the application on the server is updated or changed, the web application that the users visit will be changed. There is only one application shared by all users. As a result, the maintenance and update of the web application will be quite convenient.



- ④ The cross-platform independence. For a web application the browser is the platform. As long as the devices have a browser with the same rendering engine [2], they can support the web application identically.

Also, the web application has its own disadvantages:

- ① The difficulty of complex user interface implementation. Compared with the swing framework in JAVA, the HTML only provides the basic elements for the UI design. In the swing framework, the components can be rendered by interpolation between two elements to realize the gradual change effect. In the HTML, it is difficult to realize such functions in a convenient way.
- ② The difference in APIs of different browsers. Even though all the browsers support the Javascript and most of the APIs are same, different browsers still have some its own APIs. If a web application wants to be cross-browser, it should embed different codes for different browsers. It increases the difficulty of programming and brings about some compatibility problems.
- ③ The problem of local devices invocation. Out of consideration for the security, the web application is forbidden to access the local resources, includes files and devices. Though the HTML5 public some APIs to enable the access, it cannot work as well as the local application. The browser is an application of a system. So the web application is an application runs over another application rather than on the system directly. If the web application wants to invoke local devices, it needs to rely on browsers. It is an indirect invocation and may cause delay.

## 2.2 System Introduction

Wincor Nixdorf's software solution for Automated Checkout in Retail is a distributed Java application. The software, called TPiSCAN, consists of front-office parts (self-checkout machines, optional attendant stations) and back-office parts (server component and WEB-based maintenance components). It runs on Windows and Linux platforms.

Attendant stations visualize the agile and rich status (including the electronic receipt) of a number of self-checkout machines monitored in parallel, and offer remote reaction on problems requiring attendant interaction. For attendant stations, there are, today, two available solutions: (1) A sophisticated "fat" Java solution for stationary touch-screen Retail PCs based on Java Swing and communicating via RMI. And (2) a "functional" solution for mobile devices based on J2ME on Windows CE or Windows Mobile and communicating via Web Services and raw sockets (IP-Multicast). The structure is shown in Figure 2.2.1.

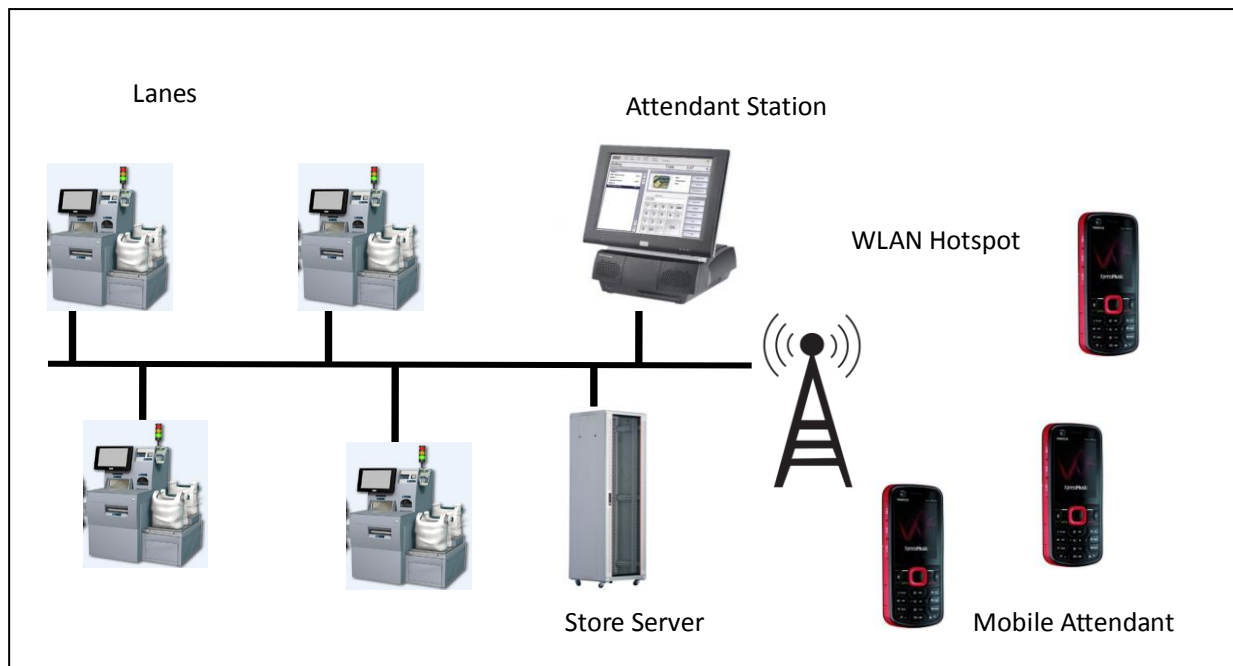


Figure 2.2.1 the Basic Structure of the System

## Component in the system

**Lanes:** self-checkout machines which replace traditional operated tills in the exit area of a store. The lanes function as cashiers. Customers can scan the products and pay the bills themselves with the help of lanes.

**Attendant Station:** an attendant station is situated in proximity to the lanes. The attendant station is used by the staffs of the shop or store. The staffs can view the status of different lanes and manipulate lanes, like void entry, void transaction by sending request to the lanes. The attendant station can also make a response when a lane sends some requests like input price or confirm the age of the customer. It is used to monitor the lanes and realize the high propriety functions those the lanes do not have. It is the supervisor of the lanes.

**Server:** The server is placed in the office. It functions as the bridge between the lanes and mobile attendant. It provides the Web Service for the mobile attendant to obtain the list of lanes and authenticate the store staff.

## 2.3 Basic Technologies

### 1) Hyper Text Markup Language (HTML) 5

“HTML is the lingua franca for publishing hypertext on the World Wide Web. It is a non-proprietary format based upon SGML [3], and can be created and processed by a wide range of tools, from simple plain text editors to sophisticate WYSIWYG [4] authoring tools.” [W3C1] It is written in a pair of tags like <html></html> and the content is placed between two

tags. The audio, video, picture or text can be embedded into a web page by using HTML. HTML can be rendered and executed by any browser. Besides, HTML can embed script languages, like Javascript which can affect the behaviors of a web page. HTML can also link with the CSS file which is used to define the appearance of a web page.

HTML5 is the newest version of HTML. It introduces some new markup elements, like <nav> used for navigation, <video> used for adding a video piece. In addition to the markup elements, HTML5 specifies some application programming interfaces (APIs) which can be used by JavaScript, like the canvas element for immediate mode 2D drawing, the File API for handling file uploads and file manipulation. Among them, the WebSocket is one of the most exciting features for the web application. It establishes the connection between client and server based on TCP and works like a normal socket. It belongs to the PUSH technologies and can be blocked in the stage of "Receive" and make a response when new message comes in. The WebSocket will be one of the most suitable technologies for the communication in web application.

## **2) Cascading Style Sheets (CSS) 3**

"Cascading Style Sheets (CSS) is a simple mechanism for adding style (e.g., fonts, colors, spacing) to Web documents" [W3C2]. The purpose of CSS is to separate the document content from document appearance, including the layout and colors. The separation facilitates the programming in both content and appearance. It also makes the specification of appearance very flexible and convenient. What is more, it also enables several pages to share the same formatting to keep the consistency.

CSS3 is the newest version of CSS. It introduces many new functions which bring the unprecedented convenience to the UI design. In the past, for example, the border radius should be realized by the pictures and the text shadow cannot be realized at all, even by the pictures. In CSS3, one line of codes can realize such functions. Also, with the introduction of the animation/ transitions and many other new functions, the gorgeous UI can be built in a very efficient way.

## **3) JavaScript**

JavaScript is a scripting language. It is designed to add interactivity to HTML pages. And it is a lightweight programming language. JavaScript is usually embedded directly into HTML pages. It is an interpreted language (means that scripts execute without preliminary compilation) and is free to use. [W3S1].

JavaScript is one of the most important client-side programming languages in web technology. Since it is weakly typed, it is an object-oriented, imperative and functional programming language. Its DOM selector helps the programmer manipulate the element of webpage and execute proper functionalities.

## 2.4 Tool Selection

### 2.4.1 Development Tool

As Eclipse is the default software for Wincor Nixdorf and already used for more than ten years, the tool selection mainly focus on the plug-in of Eclipse.

#### Introduction of Eclipse

Eclipse is an open source tool introduced by the Eclipse Foundation. It is an open development platform which comprises of extensible frameworks, tools and runtimes for building deploying and managing software across the lifecycle. It is supported by the members of the foundation and has a community that helps complement the products and services [ECL1].

Eclipse is based on Java and comprises a set of standard plug-in, includes Java Development Tools (JDT). It is a famous cross-platform Integrated Development Environment (IDE). It supports different programming languages like Java, C++, and PHP by integrating corresponding plug-ins.

#### 1) JSEclipse

JSEclipse is one the most popular plug-ins for the Javascript edit. It enhances the experience of edit by including many useful and convenient features. It includes the library dojo and prototype which can be used in the way like dojo.string. It provides the code template, code completion and even has an understanding of the code.

#### Code Template

The code template is a template that is used for simplifying coding. The user can set the name and the content of a template by this plug-in. When users type in the name, the whole template will be inserted in the code. This feature saves a lot of time for the programmer in the frequently used codes.

#### Understanding of code

The plug-in updates the understanding of the codes in real time. It can record the functions or the objects those are defined and support them for later programming. Suppose the programmer defines an object with several attributes, when the programmer create a new instance of this object, the plug-in will provides all the attributes as candidates for the programmer to eliminate the type in and recall time.

#### 2) Aptana

Aptana plug-in is a powerful plug-in for the web development. Its goal is to build the web applications quickly and easily. It provides a lot of amazing and powerful functions which really make the development convenience and efficient. It mainly includes three important

editors: HTML editor, CSS source editor and Javascript source editor.

### **HTML Editor**

HTML Editor is included in aptana plug-in for the development of a web page. It provides several functions to facilitate the programming in HTML.

- ① It sets different components in different colors. The tag is dark red, the attribute is light red, the value is blue and the content is black. The differences among these components make the programmer much more vigilant and simplify the check of codes.
- ② It provides the error reminder. Programming is a sensitive and error-prone task. The tool provides the error reminder with red underscores to indicate the place of vital errors and yellow underscores to indicate the place of warnings.
- ③ It provides the automatic completion of tag. Writing markup language always embedded one tag name into another and it may create several levels of nesting. The tool will complete the end tag automatically only if the programmer provides the formal point bracket. It facilitates the programming and avoids some fundamental errors.

### **CSS Source Editor**

CSS Editor is included in aptana plug-in for the development of Cascading Style Sheets. It provides the basic functions for the programming in CSS.

- ① Like the HTML Editor, It also sets different components in different colors to help the programmer in programming and checking.
- ② It provides the error reminder. Since the grammar of the CSS is based on the curly brackets, if the brackets are missing, red underscores will appear to indicate position of the errors.

### **JavaScript Source Editor**

JavaScript Source Editor is included in aptana plug-in for the Eclipse environment. It is designed to help web developers edit JavaScript files in a much easier and safer way.

- ① It sets different components in different colors. Not only the different type of components are in different color, but also the different type of variables are in different colors, like int is blue and string is green. It makes the different contents quite distinguishable and easy to check the codes.
- ② It provides the error reminder. The tool provides the error reminder with red underscores and yellow underscores to remind the programmer.
- ③ It provides a code assistant. The code assistant provides two major functions. The first one is code completion. It provides the candidates of codes to shorten the time of input and avoid input mistakes. The second one is the description of the method and the hint for the platform support. Even though the JavaScript is a public language for all browsers,

different browsers still have their own APIs for some functions. Therefore, consideration for the difference among browsers is one of the most important features for the web application programming. The description of the method will save the time of checking APIs and the hint can avoid some unconscious errors or mistakes to make the programming much more effective and efficient.

### **3) Comparison and Result**

JSEclipse provides many useful features in Javascript edit. However, it does not support the HTML and CSS edit. Also, the main purpose of this research is to build a browser-based application, the hint for the browser support which is provided by aptana is quite meaningful and useful. With the well support to HTML, CSS and Javascript, the aptana is selected as the main tool for the research.

## **2.4.2 Debug Tool**

### **1) Aptana**

Aptana plug-in provides also the debug tool with the development tools. The debug tool provides all the basic debug functions like breakpoint, variable monitor and step by step execution. For web debug, it provides the functions that can execute the debug action on different browser by setting the debug configuration. Then the application will be executed on the target browser and variables can be viewed in the Eclipse window.

### **2) Firebug**

Firebug is an add-one for the Firefox browser. It supports the debug function for the web application. The user can view the structure of the HTML, the CSS file that relates to the HTML, the DOM, the console window and the source code of Javascript. The user can set the breakpoints on the code and execute the application step by step. While the debug is executing, the user can view the change of all variables. Any errors or warnings will be presented in the console window. It also provides a command input area for user input.

### **3) Comparison and result**

The functions of the Aptana and Firebug are nearly the same. The debug of web application should be executed on the browser. In Aptana, the execution of the application and the monitor of variables are separated into two windows, it is not a convenient feature. In Firebug, they are displayed in the same window.

As a result, the Firebug is more convenient for debug.

# Chapter 3

## Requirement Analysis

This chapter will analyze the requirements for the implementation of the web application via analyzing the user interface and the communication mechanism of the existing attendant station.

### 3.1 Introduction of existing Attendant

The attendant station and self-checkout machines (called “lanes” in reference to the traditional operated conveyor belt-equipped checkout lanes) are already used for several years. The applications are modifying and updating all the time. The purpose of the research is to find a suitable solution which can be regarded as web version of the existing attendant. As a popular and matured business application, consistency is one of the most important rules for any update and change. Therefore, keep the consistency with the existing attendant and take full advantage of existing resources are the most fundamental rule.

#### 3.1.1 User Interface

The attendant station provides an elegant user interface on PCs [Figure 3.1.1.1]. It consists of different divisions with different functions. In the upper part (division 1), the application provides the function like “logout” for the user to log off and “change language” to change the language of the whole application. On the left side, the application displays two main divisions, division 2 is the list display, and division 3 is function panel. The list display area can display the updates in real-time with status on lanes. The function panel provides several operation choices, like “voidEntry” and “void Transaction” for the user to manipulate the self-checkout machines. It provides a tab menu for the user to change the different classification of operations. On the right side (division 4), the application displays four lane buttons to choose different lanes to monitor. Both the chosen lanes and the selected items in the list will be displayed in highlighted color to make it much distinguishable.

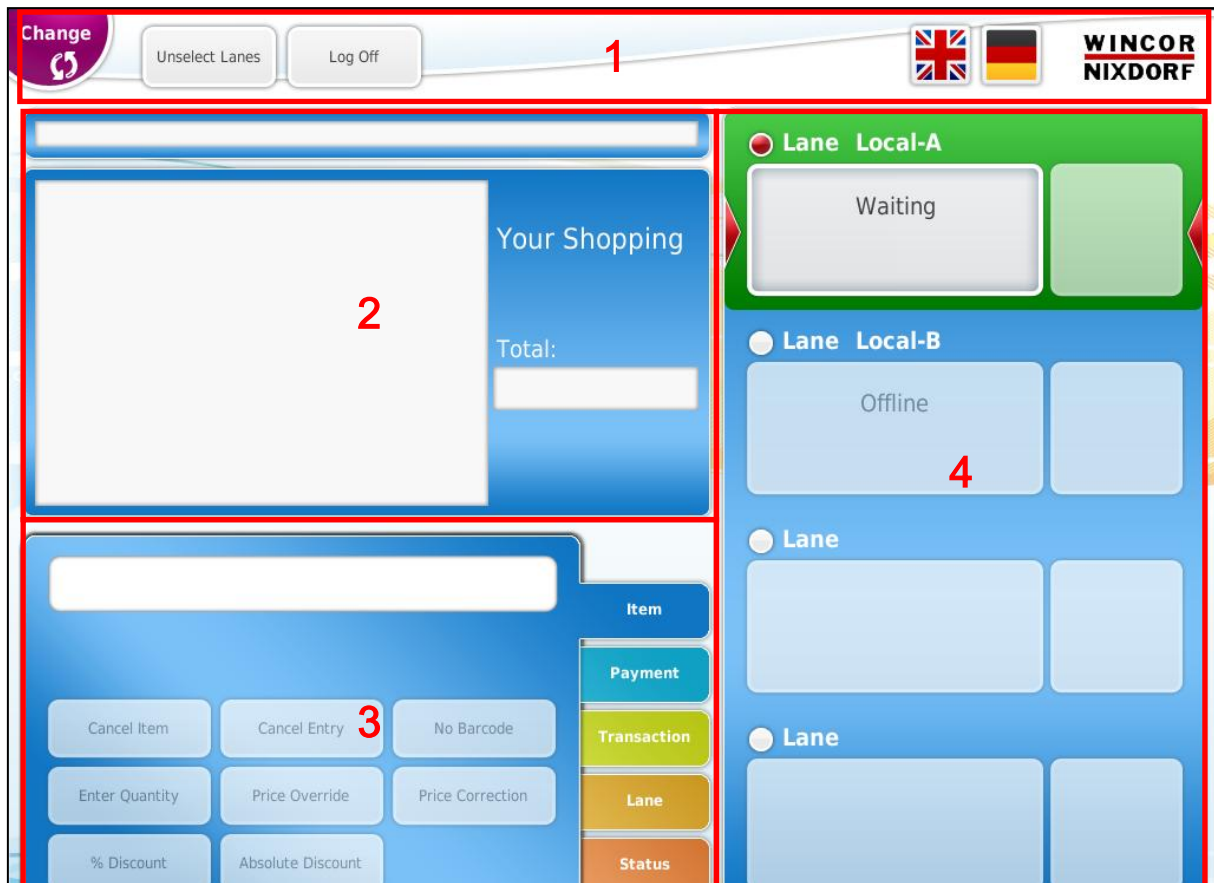


Figure 3.1.1.1 User Interface of Attendant Station

### 3.1.2 Communication of Mobile Attendant

The communication between the existing mobile attendant and lanes is based on UDP multicast and WebService. When the application starts, it requires the user to verify his/her identity. When the status of a lanes changed, including the update of list or any inquiries, the lane will multicast the message in the subnet by UDP multicast. Once the mobile attendant receives the message, it will invoke the WebService "statusFor" which is provided by the lane to get the recent state of the lane. If a lane is in the inquiry status like "confirm" or "input", the MA will make a proper response, like change panel and display the alert labels to these inquiries.

## 3.2 Application Requirement

Based on the consistency rule, combined the knowledge of the existing attendant station and the requirement of the new features like camera, the functional requirement of the new web solution can be figured out. Besides, because of the limitation of the screen size of mobile devices, the user interface should also be modified by rearranging the positions of some divisions. Consider about the user experience and the efficiency of the web application, the nonfunctional requirements and hardware requirements can be concluded.



### 3.2.1 Functional Requirement

- The application should verify the user's name before initialization.
- The UI should display the list of entities [5] and their corresponding prices.
- The entities should be selectable and cancelable.
- The selected entity should be highlighted.
- Function panel and Lane panel can be switched by slide.
- Subpanels of Function panel with different functions should be switchable.
- Different lanes on the Lane panel should be selectable.
- The selected lane should be highlighted.
- Status changed and alarm information should be displayed in a pop up window.
- The client should listen to the communication channel of UDP multicast all the time.
- The client should receive the UDP packet and extract the information in it.
- The MA should invoke the WebService to obtain the status of the corresponding lane and update the list in the UI.
- The MA should change the Function Panel into Confirm Panel when the lanes inquire a confirm action.
- The MA should change the Function Panel into Input Panel when the lanes inquire input action.
- The MA should change the input type of the Input Panel according to the type of the inquiry, like digit input, password input or alphabet input.

### 3.2.2 Nonfunctional Requirement

- The UI of web application should also comprise four divisions those contain same functions with the attend station
- The layout should be displayed identically on different mobile devices
- The update of the list should be processed within 0.5s
- The update of the new status should be processed within 0.5s
- The delay of the action on the application should be less than 0.5s
- The system should be placed on the store server

### 3.2.3 Hardware Requirement

- The mobile devices should have a screen of 4.3 inches at least
- The mobile devices should be touch based
- The mobile devices should support WLAN
- The mobile devices should have a browser and support the HTML5 and CSS3
- The mobile devices should have a comparatively high processing speed

- The mobile devices should connect to the same net with the store server
- The net should have high uplink and downlink speed

# Chapter 4

## Technical Design

This chapter will introduce the basic architecture of the system as the guideline of technologies evaluation. Then it evaluates different technologies for UI and communication in detail and chooses the suitable ones. Based on the chosen technologies, it modifies the structure and analyzes the application via user case diagram, sequence diagram, activity diagram and class diagram.

### 4.1 General Architecture

According to the requirements, the Mobile Attendant should load the page from the store server, receive the multicast and consume the WebService to communicate with the lanes.

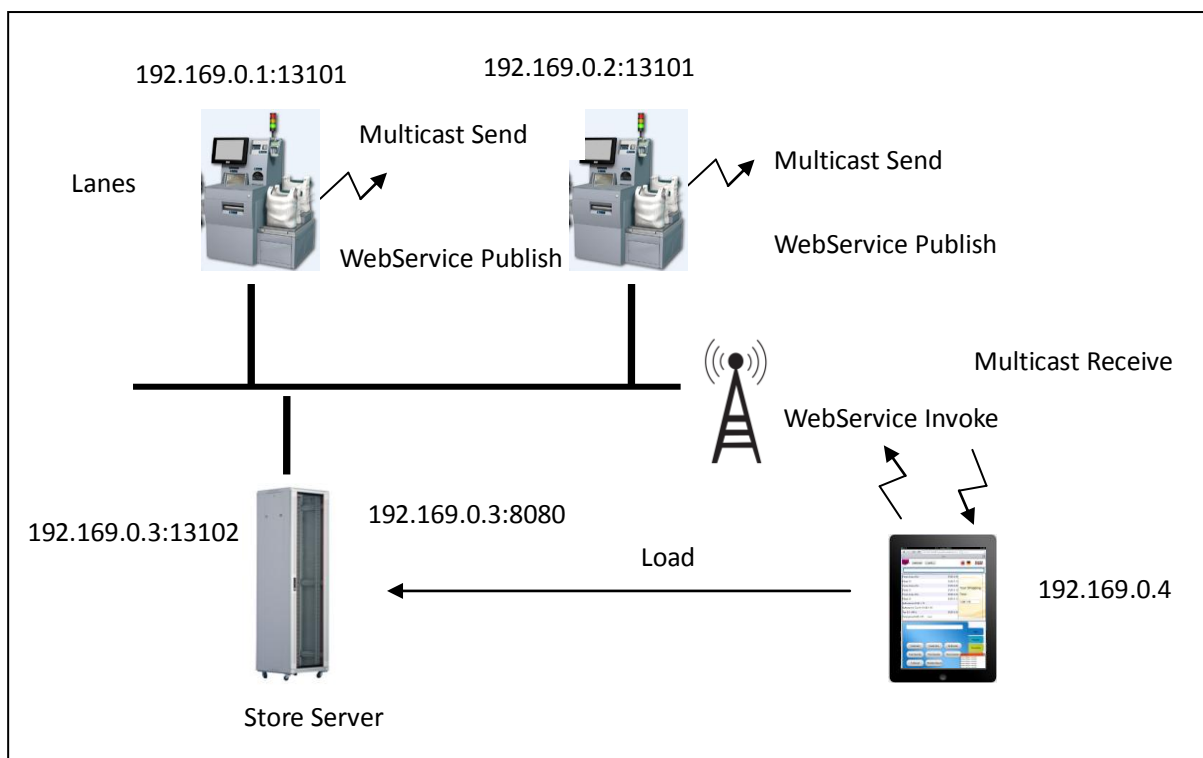


Figure 4.1.1 General Architecture

The whole system: lanes, store server and MAs are connected in the same Local Area Network (LAN) [6] and assigned with a specific IP address. The lanes, store server and the WLAN hotspot are connected by wires. The MAs connect to the system via the WLAN.

Lanes: Once a lane starts, it publishes the WebService for the MAs. When the state of the lane is changed, it will send a multicast to inform the MAs that the state is changed.

Store Server: The web application is deployed in the store server. The application can be obtained by URL.

MAs: Once a MA starts, it will load the web application from the store server by a correct URL like <http://192.168.0.1/WebApp/MA.html>. By this application, the MA opens a UDP socket to receive the multicast. Once it receives the information, it will try to consume the WebService to get the new state of the lanes. Also it can send some request to change the state of a lane actively.

## 4.2 Technology Evaluation

As mentioned in former chapters, the application comprises three main parts: 1, the user interface design; 2, the UDP transmission; 3, the WebService Invocation. For each part, there are a lot of candidate technologies. The technologies evaluation will compare different technologies and consider different aspects like support, market share and so on, to find out the most suitable technologies and design the architecture of the application.

### 4.2.1 User Interface Design

For the web-based programming, there are many kinds of programming language which have their own characteristic and widely used in different areas. Normally, they are divided into two parts, client-side programming language like Javascript and server-side programming language like JSP, PHP. In order to get rid of the dependence of the container to make the application much more flexible, the application is decided to be written in the client-side programming language to support the deployment on the client directly and reduce interactions between server and client as many as possible.

In order to simplify the programming to realize more powerful functionalities in a more convenient way, more and more JavaScript Frameworks are developed and introduced to the public. Among them, there are some popular candidates, jQuery, MooTools, Prototype, and YUI.

#### 1) jQuery

“jQuery is a fast and concise JavaScript Library that simplifies HTML document traversing, event handling, animating, and Ajax interactions for rapid web development” [JQU1]. One of the most significant values of jQuery is the more efficient writing way. It simply the complex command of JavaScript and replace it in a refined and direct-viewing way. For example, it simplifies the “document” command, which is frequently used all over the JavaScript files, to a dollar “\$”. In JavaScript, change font size need the code “element.style.fontSize=size”; In

jQuery, the code is `element.css ('font-size', 20)`. It is much more understandable.

## 2) MooTools

“MooTools is a compact, modular, Object-Oriented JavaScript framework designed for the intermediate to advanced JavaScript developer. It allows the programmers to write powerful, flexible, and cross-browser code with its elegant, well documented, and coherent API” [MOO1].

MooTools is more like an object-oriented language and support the inheritance of class. It is compatible and well supported Safari, Internet Explorer 6+, Firefox, Opera, and Chrome. It is used in some famous web application, like Bing and the official website of Formula 1.

## 3) Prototype

“Prototype is a JavaScript Framework that aims to ease development of dynamic web applications. Featuring a unique, easy-to-use toolkit for class-driven development and the nicest Ajax library around, Prototype is quickly becoming the code base of choice for web application developers everywhere” [PRO1].

It can also simplify the code in an elegant way, like transfer `document.getElementById ("id_of_element").style.color = "#ffffff"` to `$ ("id_of_element").setStyle ({color: '#ffffff'})`;

## 4) YUI

“YUI is a free, open source JavaScript and CSS framework for building richly interactive web applications”. It is published by Yahoo. It has some characteristics: 1, It has a lightweight core and model architecture [7]; 2, It provides basic DOM handler to perform and maintain the

Application	Websites	User	Page views
jQuery	359,641	133,610	4,922,701
MooTools	54,641	45,945	612,195
jQuery UI	51,980	57,731	642,170
Prototype	39,340	41,863	433,159
YUI	30,285	39,244	531,932
Script.aculo.us	9,343	13,854	108,249
Modernizr	5,102	8,513	72,575
Dojo	4,301	5,922	61,455
Xajax	1,522	2,357	20,021
ExtJS	1,350	1,416	13,193
MochiKit	256	639	4,195

Table 4.2.1.1 the Number of Each Category

application; 3, It is supported by a community and its programming tool; 4, It is open to public and free for all users [YUI1].

### Comparison of the candidates and decision

According to the statistics from [www.webappers.com](http://www.webappers.com), a popularity and market share of different web technologies can be illustrated clearly. (See Table 4.2.1.1, Figure 4.2.1.1, Figure 4.2.1.2)



Figure 4.2.1.1 the Percentage of Popularity by User

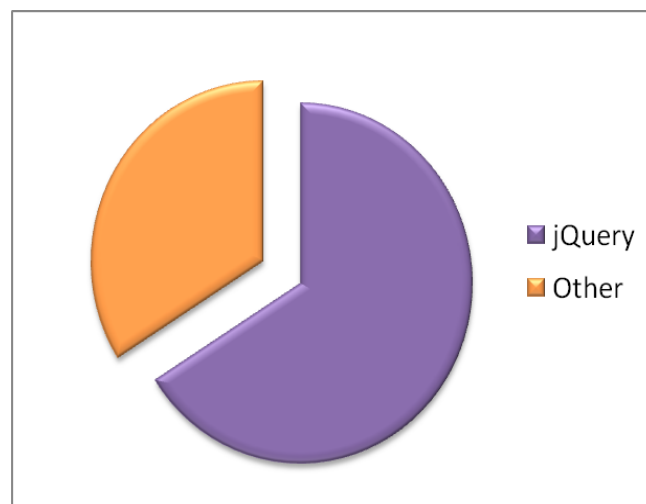


Figure 4.2.1.2 the Percentage of Page Visitors

As the statistics listed in the table and shown in the figures, nearly two thirds of the websites are written in jQuery on the Internet. Also the user of jQuery is more than half in the market. Further more, the number of page visitors of jQuery is two times of that of all other technologies. By this comparison, it is clear that jQuery enjoys the most part of the market and has most users and supporters in the world.

As a result, jQuery is selected as the main programming language for the UI design.

### 4.2.2 UDP Transmission

In Java, the Sun Company provides the API named DatagramSocket to realize the communication via UDP protocol. Because of its fast communication, it is widely used in the area like video transmission. However, in the web area, most of the communications are based on TCP/IP protocol. Therefore, there is no API that supports UDP communication in Javascript. Then it is necessary to find out some alternative solutions. ActiveX, Flash and Proxy are the most common used methods to help the UDP communication.

#### 1) ActiveX

ActiveX is introduced by Microsoft Windows. It is a small program, sometimes called add-one that is widely used on the Internet. It allows the animation or helps the task execution to enhance the user experience. [ACT1].

When a user opens a website, sometimes, there will be an alert pops up and asks the user whether to download the ActiveX plug-in. Also, some functions of a website rely on the ActiveX. The ActiveX can strengthen the interaction between the website the users.

#### 2) Flash

Flash is streaming animation for web pages. It may be a portion of an html web page or an entire web page. Before the user can view the Flash, he/she need to install the plug-in. Flash movies offer two very special web browsing experiences: very fast loading, and vector animation with interactivity [PG1]. Flash is widely used in many areas and welcomed by the users. The support for UDP starts from Flash 10, and there is an AIR DatagramSocket class which is defined for the UDP communication. There are some chat rooms which use Flash and communicate via UDP protocol already created and used on the Internet.

#### 3) Proxy

Proxy architecture is a relay station between the source and destination. It can works as a translator or a forwarder. Nearly all application can have a proxy if needed. Therefore, the proxy architecture is most popular solution to eliminate the gap between two systems. If in this way, the communication between the lanes and MA will be divided into two parts: the communication between lanes and proxy, and the communication between proxy and MA.

### Comparison of the candidates and decision

ActiveX can support the UDP transmission on the web, but only Internet Explore and the browsers which have the same kernel support it. For the mobile device, the popular browsers are Chrome, Safari and Firefox. Their kernel is Gecko. So it is not a good solution.

The Flash is popular, but with the occurring of HTML5, the social status of Flash decreases. And iOS, which enjoys a great share in the mobile devices market, does not support. For these reasons, Flash is not a promising solution.

The proxy has the capability to connect the lanes and MA well. Also, since the existing of store server in the system and it embeds the Web Container “Tomcat”, the introduction of the proxy will not cause any problems.

As a result, the proxy architecture will be a suitable technology for the UDP Transmission.

#### **a) The communication between lanes and proxy**

As the original system is written in Java, it will be better that the alternative technologies used in the proxy architecture are also Java. Therefore, Java Servlet and Java Server Pages are the most suitable candidates.

- **Servlet**

“A servlet is a Java programming language class used to extend the capabilities of servers that host applications accessed via a request-response programming model. Although servlets can respond to any type of request, they are commonly used to extend the applications hosted by Web servers. For such applications, Java Servlet technology defines HTTP-specific servlet classes” [SUN1].

For Servlet, programmers embed the HTML into Java code. The Servlet mainly focuses on the implementation of the functions rather than the web page display.

- **JSP**

“JavaServer Pages (JSP) technology allows you to easily create Web content that has both static and dynamic components. JSP technology projects all the dynamic capabilities of Java Servlet technology but provides a more natural approach to creating static content. The main features of JSP technology are: 1)A language for developing JSP pages, which are text-based documents that describe how to process a request and construct a response; 2) Constructs for accessing server-side objects; 3)Mechanisms for defining extensions to the JSP language. JSP technology also contains an API that is used by developers of Web containers” [SUN2].

For JSP, programmers embed the Java code into HTML. The JSP mainly focuses on the implementation of a web page rather than the functions.

#### **Comparison of the candidates and decision**

The function of the proxy is to build the bridge between the lanes and MA. It focuses on the function and does not need UI at all. The programming in Servlet is much more convenient for the application.

As a result, Java Servlet is a suitable technology for the communication between lanes and proxy.

#### **b) The communication between proxy and MA**

Since the MA is a web page, the communication between proxy and MA can be concluded as



a web request. Therefore, the proxy is a server, the MA is a client. The request from client to server can be realized in many technologies. The popular candidates are polling, long polling and WebSocket.

- **Polling**

“Polling, or polled operation, in computer science, refers to actively sampling the status of an external device by a client program as a synchronous activity” [WIK4]. When the server receives the Http request, it responses immediately no matter whether the required data is ready or not. The browser receives the response and sends the same request to inquire the data in a specific time interval if the server does not response correctly.

- **Long Polling**

Long polling is an improved version of polling technology. It allows an emulation push from a server to a client [WIK5]. When the server receives the Http request, it responses immediately if the required data is ready. If not, it will keep the connection and wait for the data. The connection will be closed if the connection is timeout or the required data is sent back to the client and then it will establish a new connection.

- **WebSocket**

“The WebSocket specification—developed as part of the HTML5 initiative—introduced the WebSocket JavaScript interface, which defines a full-duplex single socket connection over which messages can be sent between client and server. The WebSocket standard simplifies much of the complexity around bi-directional web communication and connection management.” [WEB1]

The fundamental work mechanism of WebSocket is similar to the normal Socket. The connection will be blocked when the connection listens to the channel and waits for the incoming data.

### **Comparison of the candidates and decision**

The polling will waste a lot of resources. If the time interval is 2 seconds, the client will make an Http request every 2 seconds and the server make a response every 2 seconds. The update action depends on the configuration of the time interval. As a result, the quality of real-time will be low.

The long polling is an improvement of the polling. Even though the server will response once the data is ready, but if the data is not prepared for a long time, the keep-live action still wastes a lot of resources, prone to create the server bottleneck. And the long polling does the periodical reconnection that also wastes the resources.

The WebSocket can keep the connection as long as needed and it supports the bi-directional communication. The WebSocket functions like normal socket so that all data can be pushed and shared within process.

As a result, the WebSocket is a suitable technology for the communication between proxy and MA.

### Technologies for WebSocket Evaluation

Since WebSocket is a new concept of HTML5, the support is few and still under development. Only some organizations implement the user interface for public user, among them, the Jetty and jWebSocket are the most popular two.

- **Jetty**

“Jetty provides an HTTP server, HTTP client, and javax.servlet container. Jetty is used in a wide variety of projects and products..... Jetty can be embedded in devices, tools, frameworks, application servers, and clusters” [JET1]. After version 7.0.1, Jetty starts to support WebSocket and publish the API for WebSocket.

- **jWebSocket**

“jWebSocket is provided to users to create innovative HTML5 based streaming and communication applications on the web. HTML5 WebSocket will replace the existing XHR [8] approaches as well as Comet services by a new flexible and ultra high speed bidirectional TCP socket communication technology. jWebSocket is an open source Java and JavaScript implementation of the HTML5 WebSocket protocol with a huge set of extensions.” The jWebSocket package contains: jWebSocket Server, jWebSocket Client, jWebSocket and jWebSocket Flash Bridge [JWE1].

### Comparison of the candidates and decision

The work mechanism of Jetty and jWebSocket are basically same. They all need to create a WebSocket first, then establish the connection, send the message and receive the message. However, the pre-condition of these two technologies are different. The jetty is a web container and supports the Java Servlet, but the jWebSocket need the jWebSocket Server which will be an additional component for this application.

As a result, jetty is a suitable technology for the support of WebSocket.

### 4.2.3 WebService

“WebService can convert the applications into Web-applications and communicate using open protocols. It is published, found, and used through the Web. Web service is self-contained and self-describing. It can be used by other applications and the XML is the basis for Web services.” [W3S2] It is a method of communication between server and client and the communication is based on the HTTP protocol.

For JavaScript, AJAX is a vital method to consume the WebService. The XMLHttpRequest object is widely used and simple. It is quite convenient to consume the WebService via AJAX.

But AJAX has a great disadvantage--- Cross Domain Limitation. The definition of the “same domain” is same hostname, same port and same protocol. Also because the existing WebService are written in JAVA, it will not support JSON [9] format which can support cross domain communication. The only way is to set up a proxy to forward the HTTP request. Therefore, it will also need the proxy architecture and the http request is divided into two parts: the http request between lanes and proxy, and the http request between proxy and MA.

### **1) The Http request between lanes and proxy**

Since the former container Jetty support Java Servlet, the proxy should be written in JAVA. As Java does not provide the API for Http request, so the library called HTTPClient which is introduced by Apache can be used as a Servlet to send an Http Request from proxy to lanes.

### **2) The Http request between client and proxy**

Since the application and proxy are deployed in the store server, they are in the same domain. MA can use AJAX to send an http request directly to the proxy.

## **4.2.4 Camera and Barcode**

As the improvement of camera technology and the need from consumers, nearly all mobile devices integrate one or two cameras. For retail, scanning barcode is an indispensable process. If the camera on the mobile device can be used, it will make the transaction much more convenience.

### **1) Camera**

In HTML5, there is a new API called “getUserMedia” introduced. It can invoke the local camera and take picture via it. It is included in HTML5 and can be used via Javascript. So the Javascript is a suitable technology for the camera use.

### **2) Barcode**

Once the camera capture the picture of barcode, it is necessary to analyze the barcode and extract the information in it. Then it needs an algorithm to analyze the picture. Reading a picture and analyze it can be realized by Javascript and its libraries. So the Javascript is a suitable technology for the barcode analysis.

## **4.3 Complete Architecture**

### **4.3.1 The Structure of the System**

According to the chosen technologies, the application introduces a new component, proxy architecture, into the system. Therefore, the structure of the system needs to be modified. It is shown in the figure [4.3.1.1]. In original system, the MAs communicate with the lanes directly.

However, now the MAs need to communicate with lanes via the proxy. The proxy builds a bridge between lanes and MAs.

The whole system: lanes, store server/proxy and MAs are connected in the same LAN and assigned with a specific IP address. The lanes, store server and the WLAN hotspot are connected by wires. The MAs connect to the system via the WLAN.

**Lanes:** Once a lane starts, it publishes the WebService for the proxy. When the state of the lane is changed, it will send a multicast to inform the proxy that the state is changed.

**Store Server/Proxy:** The web application is deployed in the Store Server. When the proxy starts, it will open a WebSocket for MAs to connect. If there is a multicast in the LAN, the proxy will receive it and forward it to MAs by WebSocket. When the proxy receives an HTTP Request, it will extract the information in it and send a new request to consume the WebService on the lane. Once it gets the response from the lane, it will forward it to the MA by HTTP Response.

**MAs:** Once a MA starts, it will load the web application from the store server by a correct URL like <http://192.168.0.1/WebApp/MA.html>. By this application, the MA tries to connect WebSocket in the proxy. If there is a piece message comes from the WebSocket, it means the status of a lane is changed. It will send a HTTP Request to the proxy and wait for the response. When the HTTP Response arrives, it will make a proper reaction according to the information in this response.

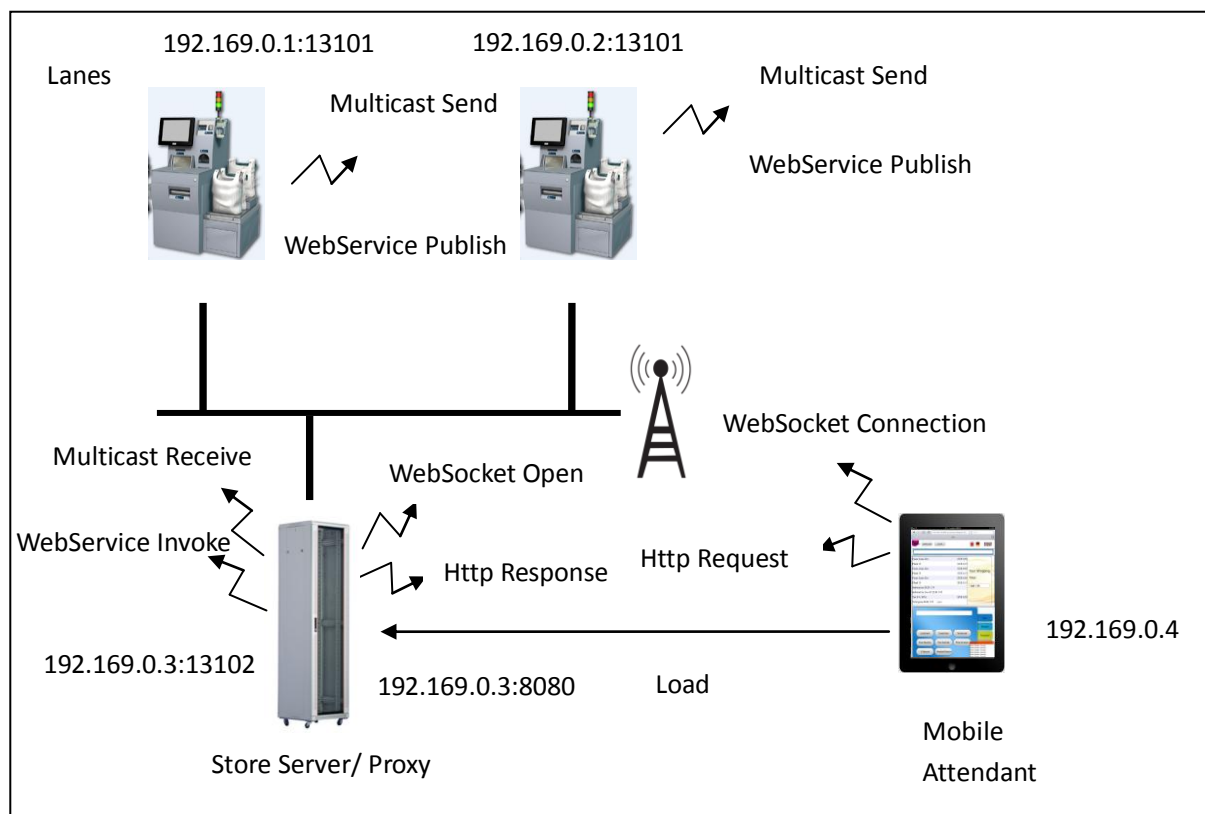


Figure 4.3.1.1 the Structure of the System

### 4.3.2 User Case Diagram (Figure 4.3.2.1)

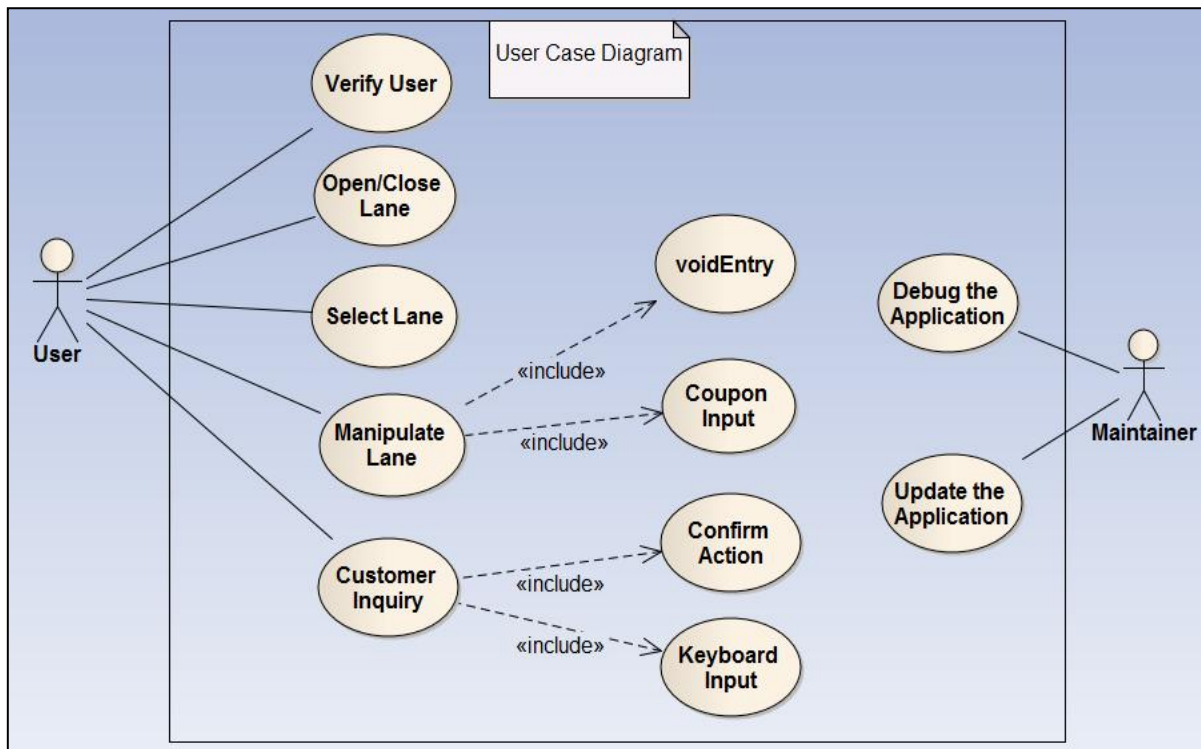


Figure 4.3.2.1 User Case Diagram

#### Description of Actors

Name	User
Description	The user is a staff of a shop or store. He/she wants to use the application to manipulate the lanes, like open or close and monitor the lanes' status. He/she needs to make the responses to the inquiry from lanes, like confirm or input some data.

Name	Maintainer
Description	The maintainer is a staff of Wincor Nixdorf. He/she needs test the application and debug when there are some problems. He/she also needs to update the application for customers by developing and releasing new version of application

## Description of User Case

Name	Verify User
Purpose	Verify the user whether he/she is a staff of the shop or store
Trigger	The user need to use the application to work
Precondition	The application is deployed on the store server correctly
Description	The user input the user name, then the system check it with the storage. If the information is correct, the main interface of the software will be displayed. If not correct, the request dialog will pop up again
Postcondition	The main interface is displayed
Quality Requirement	The verify process should be executed within 0.5s

Name	Open/Close Lane
Purpose	Open/close a selected lane
Trigger	Open/close lane that the staff wants by the mobile attendant
Precondition	The user is verified and the lane is started
Description	The staff clicks the button “open lane”, the lane will be opened and the list of the lane will be updated. The staff clicks the button “close lane”, the lane will be closed
Postcondition	The lane is opened/closed
Quality Requirement	The lane should respond within 3s.

Name	Select Lane
Purpose	View the detail information of the lane the staff wants
Trigger	The staff want to view the status of a lane
Precondition	The lane is opened
Description	By select different lanes, the list will be updated corresponding to the selected lane.
Postcondition	The list is updated, and the selected lane is highlighted

Quality Requirement	The update process should be finished within 1s
---------------------	---

Name	Manipulate lane
Purpose	Use the detailed function of each lane
Trigger	The staff needs to execute some actions to the lane
Precondition	The lane is selected
Description	The staff can void entry, input coupon and make some other manipulation to the lane
Postcondition	The request the received by the lane and come into action
Quality Requirement	The lane should respond within 3s

Name	Customer inquiry
Purpose	Handle the request from customers
Trigger	Customers inquire the help or the system inquiries actions from attendant
Precondition	Customers click the help button or the lane is blocked in the inquiry status
Description	The confirm panel or the keyboard panel will pop up when customers inquire the help. The staff needs to click the confirm button or input the data to handle the request
Postcondition	The lane and mobile attendant return to the unblocked status
Quality Requirement	The mobile attendant and lane should respond within 3s

Name	Void Entry
Purpose	Cancel the selected entry in the list
Trigger	The user wants to cancel the entry
Precondition	The entry is selected
Description	The user click the "cancel entry " button, then the application will send a cancel request to the lane
Postcondition	The entry is canceled in lane
Quality Requirement	The mobile attendant and lane should respond within 3s

Name	Coupon Input
Purpose	Input the number of coupon to reduce the payment
Trigger	The customer wants to use coupon
Precondition	The inquiry interface is displayed
Description	The user inputs the number of the coupon and sends it to the lane
Postcondition	The lanes receive the number and reduce the payment
Quality Requirement	The lane should respond within 3s

Name	Confirm Action
Purpose	Confirm the status of the lane
Trigger	The user requests help or the identity of the customer need to be confirm, like age
Precondition	The confirm panel pops up
Description	The confirm panel pops up, then the user clicks the “confirm” button if the identity is valid
Postcondition	The lane returns to the unblocked status
Quality Requirement	The lane should respond within 3s

Name	Keyboard input
Purpose	Input the digit or alphabet data
Trigger	The customer requires the user to input the data
Precondition	The keyboard panel pops up
Description	The user input the digit, date, letters or password by click the button on the keyboard panel. Then click “Enter” button to finish the action
Postcondition	The lane and mobile attendant return to the unblocked status
Quality Requirement	The mobile attendant and lane should respond within 3s



Name	Debug the Application
Purpose	Fix the bug of the Application
Trigger	The users or maintainers find the bug of the application
Precondition	The bug is found
Description	The maintainer finds out the position and the cause of the bug, and works out a solution for it. Then test the modified application to check whether the bug is fixed.
Postcondition	The bug on the MA is fixed
Quality Requirement	The debug time of each bug should be less than one week

Name	Update the Application
Purpose	Update the new version of application for the user
Trigger	The user has new request or some new features are added into the application
Precondition	The new version of the application is ready and well tested
Description	The maintainer deploys the new version of the application in the store server and tests whether it works well.
Postcondition	The new version of application is well deployed
Quality Requirement	The update time should be less than one day

### 4.3.3 Sequence Diagram

The sequence diagram shows how the communication in the system is executed. All the information exchange is synchronous.

At the beginning of using the system, the users need to start the Mobile Attendant, the proxy and the lanes. Once the proxy is started, the WebSocket is open for the MAs. Once the lane is started, the WebService is published for invocation. When a user opens the Mobile Attendant by loading the page from the proxy, he/she is required to verify his/her identity. The user needs to input the User ID and send it to the proxy. The proxy will read the user file which stores all users' ID to find out whether the user ID is valid. Then the proxy returns a piece of correct or error information back. If the information is correct, the user can use the application. All operation on the application can be realized by button clicking and the input.

The communication starts when a user inputs some data or clicks a button. The MA sends the operation result by a HTTP Request to the proxy and waits for the result. When the proxy

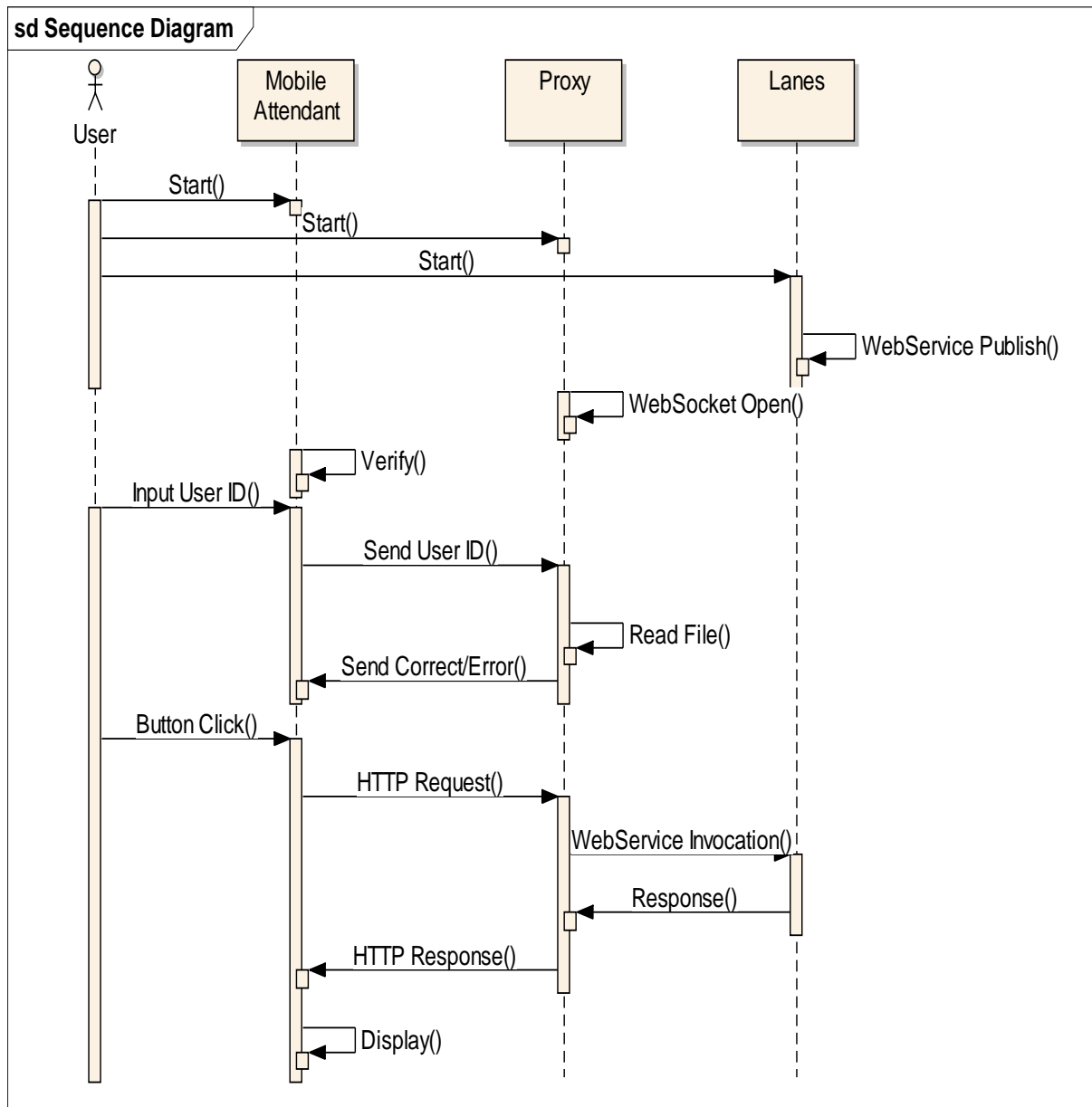


Figure 4.3.3.1 Sequence Diagram

receives the request, it will forward the information to the lanes directly. The lane extracts the information and makes a response to the proxy. The proxy will send the data to the MA as a response. In this way, the basic communication is finished. (See Figure 4.3.3.1)

#### 4.3.4 Activity Diagram

The activity diagram shows the basic work flow of the application. The activity of the application starts with the verify action. If the verify process is passed, the user can open or close the lanes and select a lane to monitor. If the status of the selected lane is changed, the list will be update or an inquiry panel will pop up, including the input and confirm action. If the status of the selected lane is not changed, the user can also make some active operations like void Entry or close the lane. If the lane is closed, the activity will end. If not, the state will return back to the “monitor lane”. (See figure 4.3.4.1)

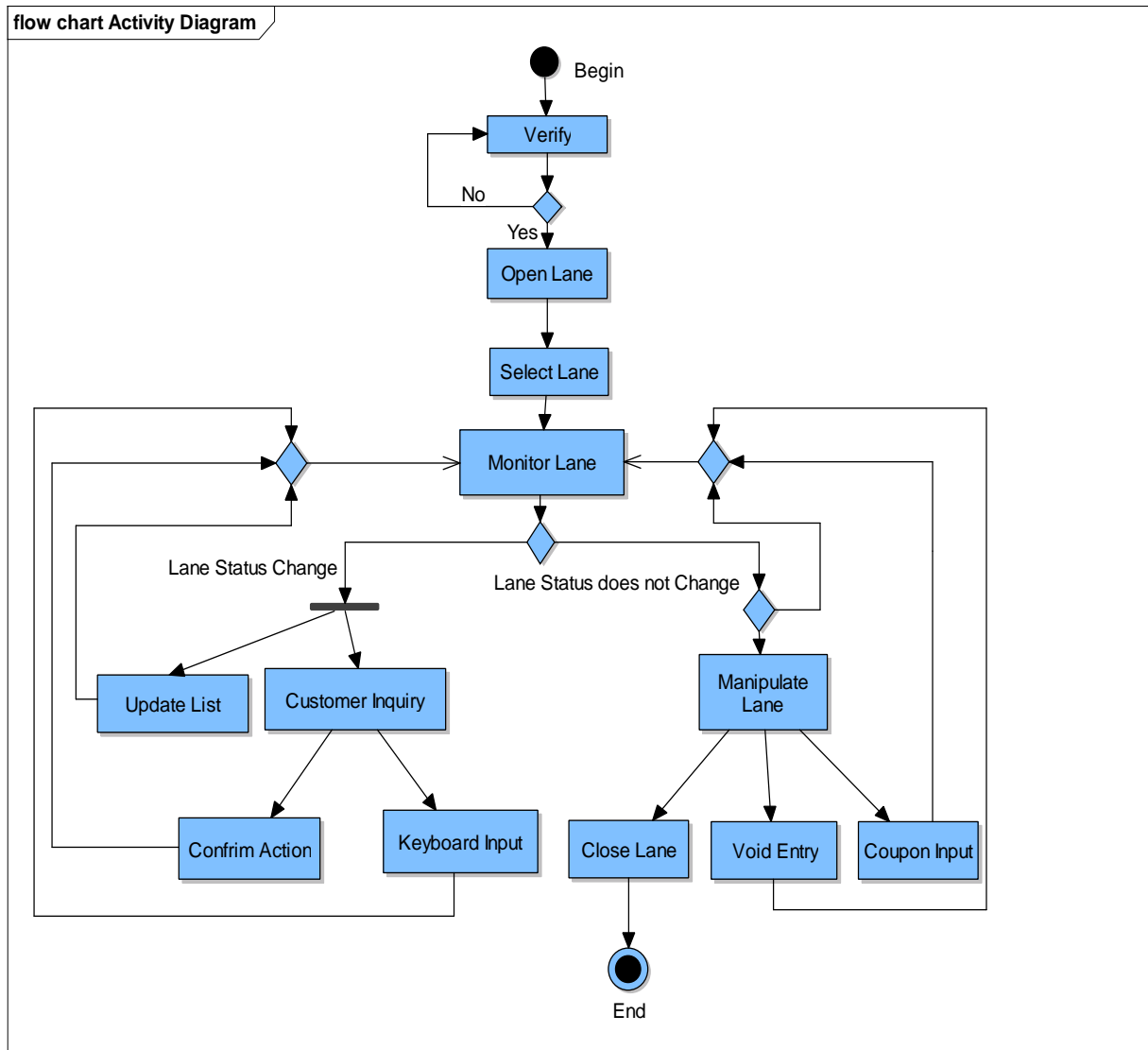


Figure 4.3.4.1 Activity Diagram

### 4.3.5 Class Diagram

The class diagram shows the relationship of the classes in the system. (See figure 4.3.5.1)

There are a lot of files used for UI implementation. The "main.html" defines the UI for this application. "main.css" defines the appearance of the UI. All the Javascript files, "verify.js", "init.js", "slide.js", "voidEntry.js" and so on are linked with the main file to realized the functions for the application.

The "startWebsocket.js" and "webService.js" are used for communication with proxy.

The files "StartServlet.java", "Server.java", "Config.java" and "Proxy0" form the proxy of the system. They communicate with "webService.js" to execute the request and response actions.

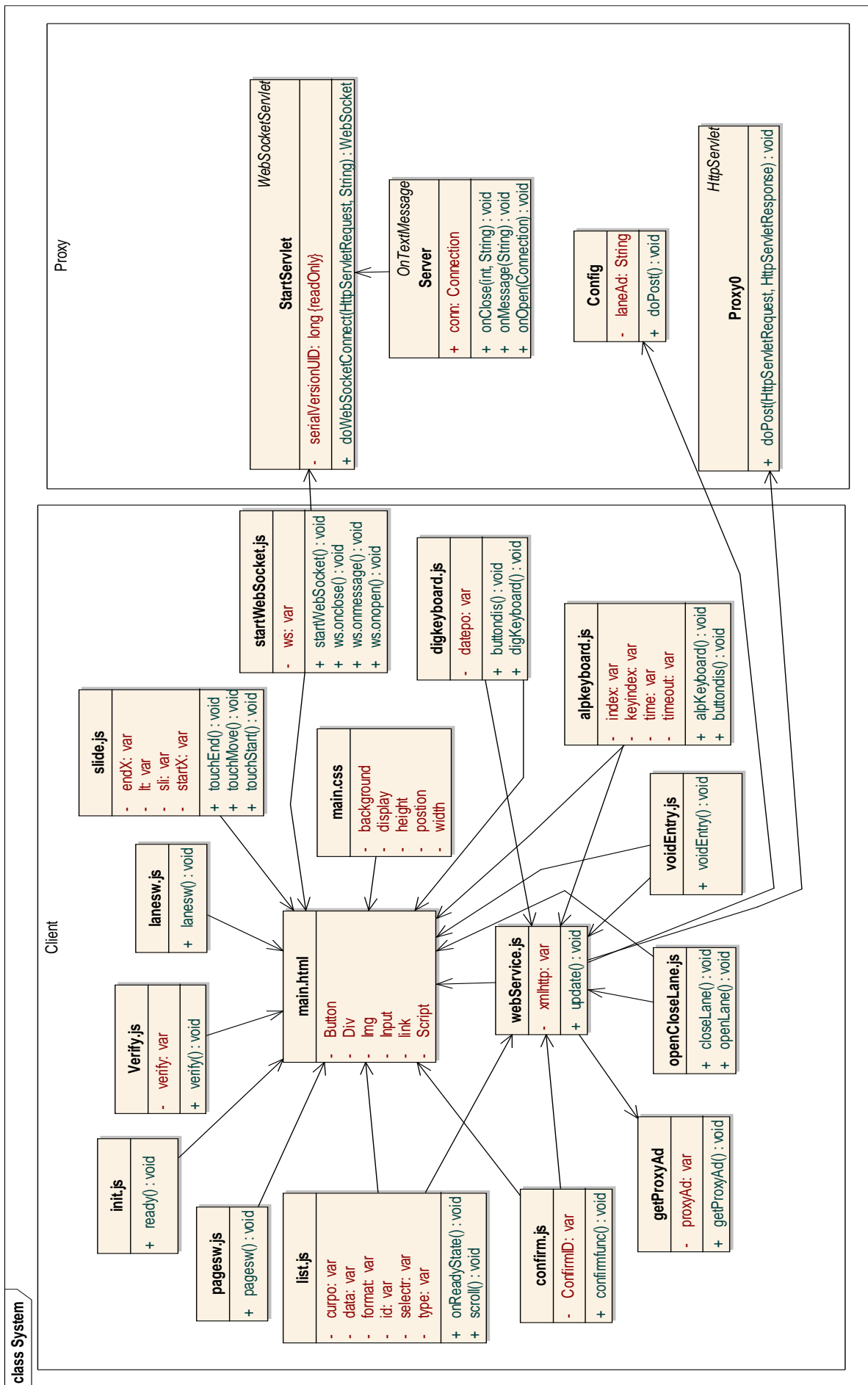


Figure 4.3.5.1 Class Diagram

# Chapter 5

## Implementation

This chapter will introduce the basic idea of the implementation of different functions and present some key code for them.

### 5.1 User Interface Design

User Interface Design is the design of the appearance of an application or a website. It includes the components needed to be shown and the basic functions of the UI itself. It focuses on the user experience and the interaction between the users and the application.

The purpose of the UI design for this application is displaying the same components those are already integrated in the attendant station and add some new features like slide to overcome the limitation of screen size.

#### 5.1.1 Component Introduction

Table 5.1.1.1 shows the name and function of each file that is used for the UI.

main.html	Contain all elements those need to be displayed in the UI
getProxyAd.js	Define the function “getProxyAd()” to obtain the proxy address
verify.js	Define the function “verify()” to verify the user name
init.js	Define the function “ready()” to load all the functions needed
slide.js/slidepc.js	Define the function of touch/click event listener to slide the Function Panel and Lane Panel
voidEntry.js	Define the function “voidEntry()” to void an entry in the list and send voidEntry request to the lane
pagesw.js	Define the function “pagesw()” to display the different subpanels of the Function Panel

lanesw.js	Define the function “lanesw()” to display different lanes and update corresponding list
OpenCloseLane.js	Define the function “openLane()” and “closeLane()” to open and close lanes
confirm.js	Define the function “confirm()” to send to confirm request to the lane
digkeyboard.js	Define the function “digKeyboard()” to display a digit keyboard for the digit and date input request
alpkeyboard.js	Define the function “alpKeyboard()” to display a alphabet keyboard for letters and password input request
list.js	Define the function “update (index)” to send the request and update the list on the UI

Table 5.1.1.1 UI Component Introduction

## 5.1.2 Key Function Description

### 1) Verify the user

The function “verify” is used to verify the user’s identity. When the MA starts, a window pops up and requires the user to input the user ID. The function will send the inputted ID to the proxy to check. If the response is correct, the main interface will be displayed. If not, the input window will pop up again.

#### Key Code

```
function verify()
{
    var userN=prompt('Please enter the User Name'); //prompt the alert for input
    check(userN,proxyAd+"/ServerSide-Jetty2/servlet/Config");
}
function check(info,url2){ //send XMLHttpRequest with info and url
    .....
}
function onReadyState2()
{
    if(res=="correct")
    {
        xmlt=undefined;
        startWebSocket();
    }
    else{verify();}}
.....
```

### 2) Slide between Function Panel and Lanes Panel

As the limitation of the screen size, the Function Panel and Lane Panel are placed in the same position. The slide function is used for switching between these two panels. “slide.js” aims at the touch-based devices and “slidepc.js” aims at the mouse click-based devices. The basic mechanisms of them are the same.

When the touch/click action happens, the system records the start X coordinate, then listens to the move action and records the end X coordinate. If the distance of movement is greater than 300 pixels, the switch is triggered and the panel will be changed to another one.

### Key Code

```
function touchStart(event) {
    if (! event.touches.length) return;
    var touch = event.touches[0];
    startX = touch.pageX;    //get start X position
}
lt.addEventListener("touchstart", touchStart, false); //listen to start action
function touchMove(event){
    .....
    var endX = startX-touch.pageX,    //get moved distance
    if(endX>300){
        sli.style.webkitTransform = 'translate(' + -1020 + 'px, ' + 0 + 'px)'; //Safari
        sli.style.MozTransform = 'translate(' + -1020 + 'px, ' + 0 + 'px)'; //Firefox
        sli.style.msTransform = 'translate(' + -1020 + 'px, ' + 0 + 'px)'; //IE
        sli.style.OTransform = 'translate(' + -1020 + 'px, ' + 0 + 'px)'; //Opera
    }
    if(endX<-300){
        .....
    }
}
lt.addEventListener("touchmove", touchMove, false); //listen to move action
```

### 3) voidEntry

voidEntry function is used to void the entry in the list. It sends a request with the ID of the selected entry to the lane to void this entry in lane's side. Then it sends the request to acquire the new status of the lane.

### Key Code

```
function voidEntry(){
    $("#candleEntry").click(function(){ //UI change
    .....
        data5='<soapenv:Envelopexmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:gen="http://generated.lane.webservice.base.iscan.retail.wn.com/">'+
        '<soapenv:Header/><soapenv:Body><gen:voidEntry><!--Optional:--><arg0>'+id[select r]+'</arg0><!--Optional:-->'+</gen:voidEntry></soapenv:Body>
        </soapenv:Envelope>';
        update(laneID,data1);    //send void entry request to the lane
    });}
```

### 4) Switch between subpanels of the Function Panel

In the existing attendant station, there are many operations for the manipulation of lanes. They are grouped and placed into different panels. These groups form a tab menu and can be switch by clicking the tab name.

In MA, the switch is realized by setting different z-index 10 to change the position of panel in the stack.

## Key Code

```
function pagesw(){ //switch page by setting z-index
    $("#detailop input[name='page']").each(function(index){
        $(this).click(function(){
            $("#detailop div").each(function(){
                $(this).css("z-index",0);
            });
            $("#detailop div").eq(index).css("z-index",1);
        });
    });
};
```

## 5) digKeyboard/ alpKeyboard

When customers inquire the input action like the weight of the goods or the date of the transaction on the lane, the user should have the capability to input the digit or alphabet on the MA.

The keyboard is realized by the operation on a string. When the button likes “1” or “a” is clicked, the keyboard will add it in the end of the string and display it. For the mobile phone keyboard in which a button has several letters, a timer is set up to assist the input. Within a specific interval of time, the letter can be changed to the next one on the button, like from “a” to “b”. After the interval of time, the last chosen letter will be added to the string.

## Key Code

```
function alpKeyboard(){
    $(this).click(function(){
        var s="";
        s=$("#alpcaltext").val();
        s2=$(this).val();
        if(s2=="Cancel") //define the function of "Cancel" button
        {$("#alpcaltext").val(""); //clear the content
        .....
        }
        else{ //simulate the keyboard on mobilephone
            if(s2.length>2){ //each button has more than one letter
                if(timeout){ //change inputted letter by clicking one button
                    clearTimeout(timeout); //serveral times within a specific time interval
                }
                var myDate= new Date();
                if((myDate.getTime()-time[$(this).index()])<2000) //within the 2s
                {
                    .....
                    index=$(this).index(); //modify the input
                    s+=s2.substring(a,a+1);
                    $("#alpcaltext").val(s);
                    if((myDate.getTime()-time[index])<2000){
                        s=s.substring(0,s.length-2)+s2.substring(a,a+1);
                        $("#alpcaltext").val(s); }
                    time[index]=myDate.getTime();
                }
                timeout=setTimeout(function(){$("#alpcaltext").val(s.substring(0,
                    s.length-1)+"*");},3000); //set a timeout
                if(s2.length==1){
                    s+=s2;
                    $("#alpcaltext").val(s);
                }
            }
        }
    });
};
```



## 6) Update the list

When the list on the lanes is changed, it will be sent to the MA. After receive it, the MA extracts the key information in the list and display it on the UI. If the table is overflow, the button "up" and "down" will appear and help to scan by scroll. If the customer has some other request like "weight input" "password input" or just wants to call the assistant, the Confirm Panel or Keyboard Panel will appear. The list also records the ID of each entry and highlights it if it is selected by click.

The list works based on the DOM selector and XPath. It will find the destination node by the code like `$(a).find("openRequest>id")`. After extract the information in it, it will build a table and display it on the UI.

### Key Code

```
function onReadyState()    //triggered when the state of the XMLHttpRequest change
{
    var a=xmlhttp.responseXML;    //get response
    $(a).find("entries").each(function(){    //execute for each entities tag
        $("#total").val($(a).find("total").text());
        var lines=$(this).find("lines").text(); //get the text within the
<lines></lines>
        var amount=$(this).find("amount").text(); //get the text within the
<amount></amount>
        $("#list").append("<tr><td
style='width:630px;height:50px;font-size:30px'>"+lines+"</td><td
style='width:130px;height:50px;font-size:30px'>"+amount+"</td></tr>"); //display
the text

        .....
        if($("#list").height()>500){    //display "down" button

            $("#down").css("display","block");

        }
    });
    .....
    if(rtext=="Input of a digit sequence (attendant)")
    {
        $("#cal").css("display","inline-block");
        $("#caltext").val("");
        $("#leftthree").css("display","none");
    }
    .....
    function scroll(){
        .....
    }
```

## 5.2 UDP Communication

The UDP communication is realized by WebSocket.

WebSocket is used for communications between server and client. It is a new feature of HTML5 and is becoming more and more popular in web technology.

For this application, WebSocket is used for forwarding the UDP packet from lanes to the MAs. The proxy will open a WebSocket for the MAs when it starts. Also, the proxy will embed a

UDP receiver in the WebSocket and listen to the UDP channel all the time. Once the UDP package arrives, the send function of WebSocket will be triggered to forward this package. The MA will connect to the WebSocket when it starts and listen to the WebSocket all the time.

## 5.2.1 Component Introduction

Table 5.2.1.1 shows the name and function of each file that is used for the UDP Communication.

StartServlet.java	Extend the WebSocketServlet and overwrite the method doWebSocketConnect
Server.java	Implement the OnTextMessage to implement different methods like onClose, onOpen and onMessage. And Set up a UDP socket to receive the multicast and forward it to the MA by WebSocket
startWebSocket.js	Create a WebSocket and connect it to the proxy

Table 5.2.1.1 UDP Communication Component

## 5.2.2 Key Function Description

### 1) WebSocket on Proxy

The set-up of a WebSocket on the proxy is simple. The Servlet extends the WebSocketServlet which is defined in jetty and implements the interface onTextMessage. When rewriting the method “onOpen”, a UDP socket is created and connects to the multicast group of the lanes. If there is a multicast in the group, the proxy will receive it and extract the information in it. Then it sends information to the MA by “send” method in WebSocket technology.

#### Key Code

```
public class Server implements OnTextMessage {
    public void onClose(int arg0, String arg1) {
        .....}
    public void onOpen(Connection conn) {
        .....
        try{
            InetAddress group = InetAddress.getByName("230.0.0.42");
            MulticastSocket s = new MulticastSocket(4447); //set up a socket to receive
UDP multicast
            s.joinGroup(group);
            byte[] buf = new byte[1000];
            DatagramPacket recv = new DatagramPacket(buf, buf.length);
            while(true){
                s.receive(recv);
                String re=new String(recv.getData());
                String re1=re.substring(0,8);
                conn.sendMessage(re1);}} //write the udp package back to client by WebSocket
```

## 2) WebSocket on MAs

When a MA starts, it will create a WebSocket and connect it to the WebSocket on the proxy.

The step of communication is similar to that on proxy. However, the WebSocket on the MAs need to distinguish the source of the information. If the information comes from the monitored lanes, it will update the list. If not, it will send the request to the source and find out whether there is an inquiry and display a pop up window if there is an inquiry. This mechanism realized parallelism of monitor.

### Key Code

```
function startWebSocket() {  
    if (!window.WebSocket) alert("WebSocket not supported by this browser!");  
    ws = new WebSocket("ws://" + proxyAd + "ServerSide-Jetty2/StartServlet.do");  
    ws.onmessage = function(evt) {    //if the UDP multicast is from selected lane,  
    update  
        if(evt.data==laneAd[0])  
            { laneID=0;}  
        if(evt.data== laneAd[1])  
            { laneID=1; }  
        if($("#input[name='lane']:checked").val()==laneID){  
            update(laneID,data1);  
        }  
        else{    //if not, try to find the inquiry and display  
            check(data1,proxyAd+"/ServerSide-Jetty2/servlet/Proxy1");  
            hostdata=evt.data;  
        }  
    }
```

## 5.3 WebService Consumption

WebService is a kind of service that the server provides for the client to realize communication between server and client.

In this application, the lanes provide the WebService for the MAs to obtain the information about it or make some operation on it. The MAs send the HTTP Request to the proxy and the proxy creates a new XMLHttpRequest to the lanes to consume the WebService. After the lanes respond, the proxy will forward the response information to the MAs.

### 5.3.1 Component Introduction

Table 5.3.1.1 shows the name and function of each file that is used for the WebService

webService.js	Set up an AJAX XMLHttpRequest and send XML format information to consume the WebService on the Lane
Proxy.java	Receive the post information from client and set up a new HTTP request to the lane and forward the information from lanes to client
Config.java	Realize the functions of verification and setting the lane address

Table 5.3.1.1 WebService Component Introduction

## 5.3.2 Key Function

### 1) WebService Consumption on MAs

When the WebService consumption is needed, the MA creates an XMLHttpRequest object and sends the request via it. Then it listens to the status change of the object. When the ready state is four, it means the data is sent. Then the MA can process the response.

#### Key Code

```
function update(index,data) //webservice method for communication
{
    xmlhttp = new XMLHttpRequest();
    url=proxyAd+"/ServerSide-Jetty2/servlet/Proxy"+index;
    if(xmlhttp)
    {
        xmlhttp.onreadystatechange = onReadyState;
        try
        {
            xmlhttp.open( "Post", url, true);
            xmlhttp.setRequestHeader("Content-Type","text/xml");
            xmlhttp.setRequestHeader("Cache-Control", "no-cache");
            xmlhttp.send(data);
        }
        catch(exception){}
    }
}
```

### 2) WebService Consumption on Proxy

When an XMLHttpRequest arrives, the proxy will extract the data in it and create a new HTTP Request by HTTP Client to post the data to the lanes. When the response from lanes arrives, the proxy will extract the data from it and make a response to the MAs.

#### Key Code

```
public class Proxy0 extends HttpServlet{
    public void doPost(HttpServletRequest hrequest,HttpServletResponse hresponse)
    throws IOException {
        HttpClient httpclient = new DefaultHttpClient();
        hresponse.setContentType("text/xml");
        HttpPost httppost = new
        HttpPost("http://"+Config.laneAd[0]+":13102/iSCAN-Lane/Lane"); //define post
        address
        .....
        buf.read ajaxContent,0,length); //get request content posted by client
        String receivedContent = new String(ajaxContent,0,length,"UTF-8");
        //transfer the content
        soapRequestData.setContentType("text/xml");
        httppost.setEntity(soapRequestData); //set the content to a new request
        HttpResponse response = httpclient.execute(httppost); //do post and get
        response
        HttpEntity entity = response.getEntity();
        if (entity != null) {
            try { pw.write(output); //write response back to the client
            }}
    }
}
```

## 5.4 Camera and Barcode

As the APIs for the camera is quite new, only the browser Opera supports it. On mobile devices, only Opera Mobile supports it, the camera and barcode function are not integrated into the whole application. They are kept as a standalone application.

### 5.4.1 Component Introduction

Table [5.4.1.1](#) shows the name and function of each file that is used for the Camera and Barcode

camera.html	Set up a video tag to present the camera stream and a canvas to present the captured picture.
camera.js	Obtain the local camera and present the stream on the web page. Define a method to capture the picture when the button is click and then analyze it to obtain the barcode

Table 5.4.1.1 Camera Component Introduction

### 5.4.2 Key Function

#### 1) Camera

When the web page is load, the application obtains the local camera by an API “getUserMedia”, and then it returns the success for fail. If success, the controller will return the stream as a parameter and the stream will be present on the web page.

#### Key Code

```
var video = document.getElementsByTagName('video')[0];
if(navigator.getUserMedia) {
  navigator.getUserMedia('video', successCallback, errorCallback); //get camera
function successCallback( stream ) {
  video.src = stream;    //assign the stream to the element, present it
}
function errorCallback( error ) {
  alert("error");
}
function capture() {
  var canvas = document.getElementById('myCanvas');
  var ctx = canvas.getContext('2d');
  ctx.drawImage(video, 0, 0, canvas.width, canvas.height);    //take the picture by
  drawImage
} }
```

#### 2) Barcode

The barcode analysis is thoroughly the image analysis. It is needed to read the image information line by line, to distinguish the color on the image. There is a free barcode recognition code which is licensed by MIT published on the Internet [\[GIT1\]](#).

## KeyCode

```
var doc = document,
    rows=2,
.....
while(numLines--){
    console.log(spoints[numLines]);
    var pxLine = ctx.getImageData(0, slineStep * spoints[numLines], width,
rows).data,
    .....
    for(var i = 1, l = lines.length; i < l; i++){
        if(code.length < 6){ var group = lines.slice(i * 4, (i * 4) + 4); }
        else{ var group = lines.slice((i * 4) + 5, (i * 4) + 9); }
        var digits = [
            Math.round(group[0] / bar),
            Math.round(group[1] / bar),
            Math.round(group[2] / bar),
            Math.round(group[3] / bar)
        ];
        code += u[digits.join('')] || u[digits.reverse().join('')] || 'X';
    }
    .....
}
```

# Chapter 6

## Result and Evaluation

This chapter will present the prototype of the application. And test it on different platforms, devices and browsers. Then it will analyze the result and evaluation the prototype. Finally, it will compare the prototype with fixed attendant station and introduce some existing problems.

### 6.1 Result

#### 6.1.1 Function Implementation

After the implementation of each function, a prototype is built and it has the fundamental functions in UI and communication. It has fashionable features in the user interface with necessary functions (see Figure 6.1.1.1). The positions of division 1, 2 and 3 are the same with the fixed attendant station. Division 4 is placed in the same position with the division 3 and hidden as default.



Figure 6.1.1.1 Main UI

- Before using the system, the user is required to input the user name. If it is correct, the main user interface will be displayed. If not, the input dialog will pop up again to require new input (see Figure 6.1.1.2).



Figure 6.1.1.2 Log in

- The application can connect to the Proxy by WebSocket (see Figure 6.1.1.3).



Figure 6.1.1.3 WebSocket Check

- The user can switch the panel by clicking the tag on the right (see Figure 6.1.1.4).

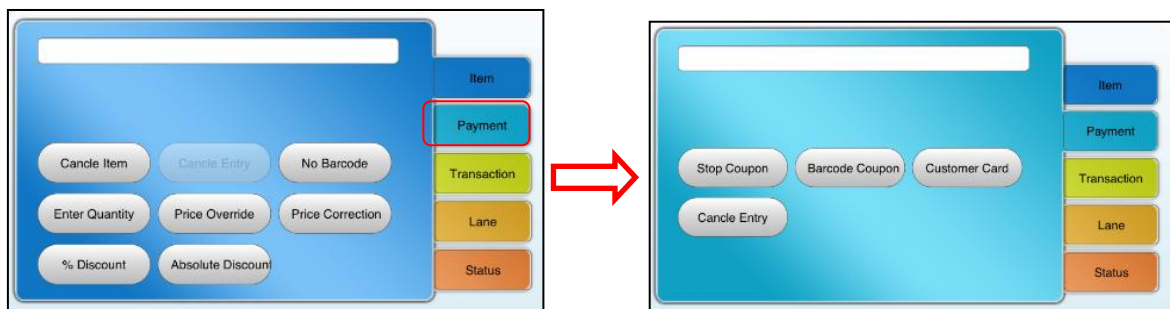


Figure 6.1.1.4 Switch the Subpanels

- The user can switch the Functional Panel and Lane Panel by slide (see Figure 6.1.1.5).
- When the status of other lanes changes, a small window will pop up in the right-down corner and display the name of the host in it (see Figure 6.1.1.6).

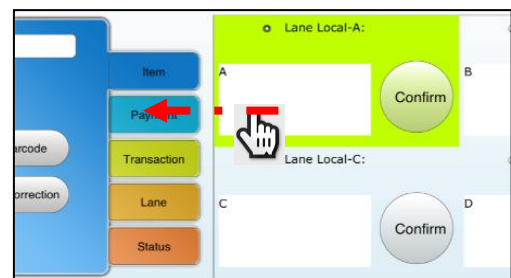


Figure 6.1.1.5 Panel Switch

- The user can void entry and input coupon by clicking buttons on the Function Panel.
- When the list on the lane changes, the list on MA will update at the same time.
- When customers inquire help, the confirm panel will be displayed.
- When customers inquire an input action, the different keyboard will be displayed according to the different input format (see Figure 6.1.1.7).

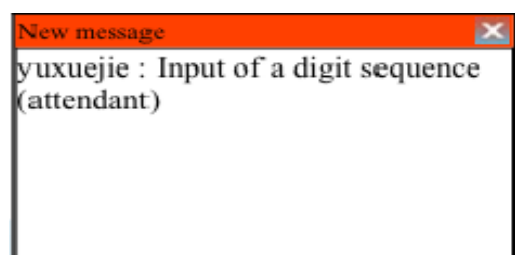


Figure 6.1.1.6 Alert Window

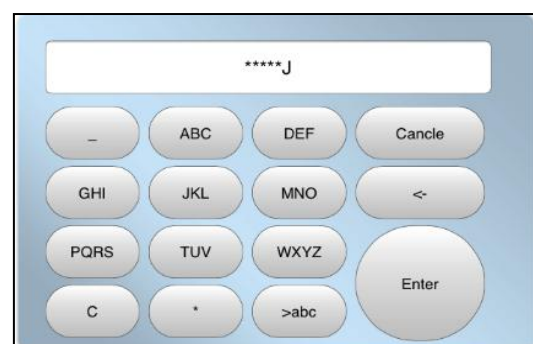


Figure 6.1.1.7 Password Input Keyboard



## 6.1.2 Application Test

The test is done on the PC (Windows 7 and XP), Tablet (iOS 4.0 and iOS 5.0) and the Smartphone (Android 4.0). The test is executed on the popular browsers for each different platform. It tests the basic UI functions, the communication functions and the camera. The result is shown in the Table 6.1.2.1.

For Windows, the Firefox (version 13.0.1), Chrome (version 19.0.1084.56 m) and Safari (version 5.1.2) supports the UI functions and the communication functions well, including the slide and WebSocket. But they do not support the camera. The Opera (version 12.00) support the UI functions and camera well, but it does not support the WebSocket. The Internet Explorer (version 8.0.6001.18702IC) (IE) does not support the WebSocket and camera. What is more, it cannot render the UI. So the UI is in a mess in IE.

For iOS, Safari (iOS 4.0, 5.0) and Diigo (version 3.4) support the UI functions and the communication functions well, but it does not support the camera. Opera Mini (7.0.1) cannot load the application from server at all.

For Android, Android browser (version 4.0) supports the UI render and slide function. But because of the screen size, the parameter of slide distance is different from that in the tablets or PCs. It does not support WebSocket and camera. Opera Mobile (version 12.00) supports the UI functions but also need distance parameter modification. It supports the camera but does not support the WebSocket. Firefox Mobile Beta (version 14.0) supports the UI functions

Platform	Browser	UI functions	WebSocket	Camera
Windows	Firefox	X	X	
	Chrome	X	X	
	Safari	X	X	
	Opera	X		X
	IE			
iOS	Safari	X	X	
	Diigo	X	X	
	Opera Mini			
Android	Firefox	X	X	
	Chrome	X	X	
	Android Browser	X		
	Opera	X		X

Table 6.1.2.1 the Test Result

X: support

Empty: nonsupport

/ :unknown, cannot test

without modification and the communication functions well, but it does not support the camera. Chrome Mobile Beta (version 0.18) supports the UI render and slide function but also need distance parameter modification. It supports the WebSocket but not camera.

### 6.1.3 Result Analysis

According to the result on different platforms and browsers, it can be concluded that the application relies more on browsers and devices than platforms.

From the view of browsers, the new versions of Firefox, Chrome and Safari on different platforms support the HTML5, CSS3, Javascript and its library jQuery well. But because the APIs of camera manipulations is quite novel, the APIs are not integrated into the browser. For Opera on different platforms, it supports the camera APIs and UI functions. But it does not support the WebSocket.

From the view of devices, the difference occurs only in the UI functions. The slide function is triggered by the slide distance. Because of the smaller screen size of the Smartphone, the slide distance of the Smartphone is much smaller than that in Tablets and PCs. As a result, the slide function on Smartphone needs modification of the parameter in UI.

## 6.2 Application Evaluation

The application evaluation is based on the criteria. The evaluation is executed in two aspects: the user aspect and the maintenance aspect. The criteria are defined according to some rules of the “Ten Usability Heuristics” rules which are introduced by Jakob Nielsen [JN1] and some rules of the “Software Evaluation Process” which is introduced by OMII-UK organization [OMI1].

### 1) User Aspect

#### Understandability

Understandability is one of the most fundamental requirements of software. The user should know what can the software do and what are the functions. It depends on the user interface layout, the letter, the logo and pictures which are used in it.

In this UI, the label like “Your shopping” “total” indicates that it is used in the settlement in shops or stores. The tag menus with different names like “Item” “Lanes” bellowed imply that the user can switch different panel by click different button. The list area can display the list of shopping. In the Lane Panel, the user can have an idea of selecting lanes without seconds.

## **Installation**

The software consists of two parts, the client side and proxy architecture.

For the client side, it is a web-based solution written in HTML, CSS, Javascript and jQuery. It is support by all popular browsers, like Firefox, Chrome, and Safari. The user does not need any configuration. He/she just need to type in the URL of the web page and then enjoy the functions in it.

For the proxy, it is Java Servlet and has WebSocket server side. It needs to be run on the Jetty. Jetty is a light-weight web container which is popular and be used by Google. It is very simple to deploy the application.

## **Compatibility**

As the goal of this research is to build a web-browser based application for the self-checkout machines, the compatibility of the application is one of the most significant features. The result of application test shows that the application is highly compatible with different browsers and devices.

## **Visibility of system status**

The visibility of status means that the system should provide an appropriate feedback when the system is running. It helps the uses have an idea what the system is doing. But unfortunately, because when a lane updates one time, it will multicast several UDP packages. As the result, the MA will update several times for one status change. The high frequency of update and the short time delay make it improper to inform the user the current process. The application does not inform the user what is happening. It just displays the result directly. Or if fails, nothing happens.

## **User control and freedom**

It means the application should provide functions like exit, redo and undo to guarantee the user's freedom. However, for such settlement application, it does not make sense to redo or undo an action. Users void the entry or input data will come into action once the button is clicked. Therefore, redo and undo are not the appropriate function.

## **Error prevention**

The functions of the application are quite simple and intuitive. "Choose" and "Click" with some input can finish all the function needed. It can ensure the rate of correct action by limit the buttons and input content. The user can even have an idea of how to use the application without any pre-training.

## **Recognition rather than recall**

All functions of the application are explained clearly in the UI and sorted in reasonable groups. The user just needs to click different buttons according to the text on them. When an input inquiry occurs, the purpose of the input will also displayed above the list to help the user understand what to input and what it is for. Also, by reasonable grouping, the user does not

need to remember the position of each button.

### **Aesthetic and minimalist design**

As a business application, it displayed everything needed and hidden anything irrelevant. To improve the experience of use, the software chooses the light bright color and clear division. By overlapping different panels and switching by clicking or sliding, the software presents a clear UI with powerful function.

### **Help users recognize, diagnose, and recover from errors**

Error is inevitable for any application. The application cannot provide any function to help error recognize or diagnose and recover from errors.

### **Help and documentation**

The software does not provide any help functions and documents to help the user.

## **2) Maintenance Aspect**

### **Testability**

The application can be tested by the simulation lane and pos machine. The simulation lane is a lane which runs on PCs and can realize all the functions of a real lane in a shop or store. It accompanies a simulation POS machine that can send the pos requests to the attendant. The tester can test all the function of the application by changing the status of the lane like changing the list by scanning new items and requires the help by clicking the button on the lane. Also, the tester can send inquiries from the pos machine to require the user to input the password, date or weight. All results can be displayed on the UI.

What is more, the test can execute not only on the mobile devices, but also the PCs. As long as the same browser is used, the result will be identical.

### **Portability**

Since the software is web-based. It can run on all the browsers those support WebSocket and camera invocation, including PCs, Tablets and Smartphones.

### **Changeability**

The main part of the software is written in Javascript, CSS and html. The separation of the layout and functions make the software flexible to modify. Modification of the layout or the functions can be executed individually. Also, the user can modify the communication address by modification of a text-based configuration file.

## **6.3 Discussion**

### **6.3.1 Comparison of the PC-based application and the Web-based application**

#### **1) Compatibility**

##### **PC-based application**

PC-based application is design for the PCs. The whole application is written in Java, so it is platform independent. However, it still dependences on the Java Virtual Machine (JVM). Only where there is a JVM, can the application run.

##### **Web-based application**

Web-based application is design for the mobile devices and PCs. The application is written in HTML, CSS, Javascript, jQuery and some part of Java. It depends on the browsers which are integrated in any operating systems and becoming an indispensable component for the systems. HTML, CSS and Javascript can be rendered by any browser. Therefore, it even does not need to install a virtual machine. The “virtual machine” for the application is already integrated by the devices.

#### **2) User Interface**

##### **PC-based application**

The PC-based application builds up the user interface by java swing. It is a powerful toolkit for the UI design. The UI of the application is elegant and clear. With the help of the operating system, it works fluently and responds quickly. However, the inseparable programming of UI components and its appearance make the maintenance harder and more complex.

##### **Web-based application**

The web-based application builds up the user interface by HTML, CSS, Javascript and jQuery. The HTML provides the elements like input area, buttons. The CSS defines the appearance of these components, like width, color. The Javascript and jQuery set the functions of the elements, like click, hide. The separation of the UI design makes it much easier for maintenance. Also, it takes the advantage of the touch-based devices, But because of the capable limitation of the web browser, in some devices, the UI functions not as fluently as the PC-based one, especially in render.

#### **3) Communication**

##### **PC-based Application**

The PC-based application communicates with the lanes via RMI. And it can receive the UDP multicast by itself. As a result, the application can execute all the functions those are needed for the whole communication process. And as it is directly communication and data is

transmitted by wire, the speed of communication can be very fast.

### **Web-based application**

Because of the limitation of the UDP and AJAX, the web-based application cannot communicate with the lanes directly. The proxy architecture is the relay station in the information chain. Inevitably, the communication speed will be slowed down. The speed also will be affected by the speed of the LAN. Thus, compared with the PC-based application, it is more complex and slower.

### **6.3.2 Problem Discussion**

During the test of the web application, some problems occur and need to be fixed to promote the quality of the application.

#### **Multi Update**

When the status of a lane changes, in order to ensure the notification, the lane will multicast three times within one second, the application will make a communication with the lane and update the list for each multicast. As a result, it will make two redundant updates which will occupy a lot resource. However, the multiple multicasts within one minute may also be produced by the quick manipulation of the customers. Sometimes, it is difficult to figure out whether the multicast is redundant or new before it communicates with the lanes. Unlike the wired transmission, the speed and quality of wireless transmission are limited. Also, as mentioned above, render speed is slow and the receipt update needs to be done three times, too. It will cause some problems like lose updates if the frequency of update is high. So this is a big problem for the web application.

#### **Camera**

In the web application, the basic schema of scanning an item is taking picture first then analyzing the picture. However, the success of analysis is depending on the quality of the taken picture. If the picture is clear, it can analyze it and get result. If not, it cannot deliver a result and then need a new picture. It is very inconvenient and unacceptable for the daily use. Also, the APIs for the camera are only supported by few browsers and incomplete. It is impossible to manipulate the detail functions of the camera like focusing and exposure.

# Chapter 7

## Summary

With the striking development of the Mobile Internet, the application development is shifting its center from local to web. More and more web applications come into reality and facilitate people's life. The thesis is attempting to evaluate different technologies which can be used in web application implementation for the attendant station of Wincor Nixdorf and build a draft prototype for it.

At beginning, the thesis presents the need of the web application. As the increasing need of mobile applications from customers and social status of the mobile devices, mobile technology attracts more and more producers who want to take advantages of business opportunities. The social background, especially the hitting of the market and the determinant of entering mobile market of Wincor Nixdorf trigger the research.

Then the thesis analyzes the structure of the existing application to specify the functions of each part. It introduces the HTML, CSS and Javascript as the basic technologies for this research. Also it introduces the basic tools, Eclipse and Aptana, which are useful for the research.

Later, the thesis concludes the requirements of this application by analyzing the existing system and demonstrates the structure of the application. It analyzes the basic scenarios of the application according to the requirements. It illustrates the user case diagram, sequence diagram, activity diagram and class diagram to specify each function and analyze the whole process of this application.

Afterwards, the thesis decides to divide the application into three parts: User Interface Design, UDP transmission and WebService. It evaluates the possible technologies for each part and chooses the jQuery, WebSocket, AJAX as the most suitable technologies.

In the implementation part, the thesis declares the basic concepts of the implementation of different functions. It presents the purpose of each function and the basic process to realize the functions.

The thesis also tests the application on different platforms, browsers and devices and then

analyzes the result. It evaluates the application by criteria to assess the web application in user aspect and maintenance aspect. What is more, it compares the web-based application with the existing PC-based application and introduces some existing problems.

## **7.1 Brief view on the Application**

The mobile attendant application is a web application which is implemented for attendant station in the system of TPiSCAN. It is evolved from the fix attendant station. The user interface and functions are realized based on the user cases of attendant station.

The application provides the login process which verifies the user's identity by comparing inputted information with the local data. After login, it presents a fashionable user interface with novel concepts like slide and pop up. Meanwhile, the application connects to the proxy via WebSocket to listen to the multicast. After the multicast information arrives, the application will require consuming the WebService with the help of the proxy architecture. Then it updates the shopping list or presents some inquiries like confirm or input. Also, the application can communicate with the lanes actively like changing the monitored lane and void the entry in the shopping list.

The web application meets the requirements of the research and realizes the vital functions of the attendant station. It builds the basic structure of a complete application in further and provides valuable hints for further development.

## **7.2 Recommendation**

The thesis analyzes the system and builds a prototype for it. However, the web application only realizes a part of the functions.

For user interface, the web application can only present one lane's list. In further research, it should implement all the rest functions of the existing attendant stations like providing a sub window to display all lanes' lists in parallel.

For communication, the web application only realizes some basic functions to give hints for other missing functions. In further research, the web application should add more functions like "void transaction" and "price overwrite" by AJAX to consume the WebService to make the application complete.

For maintenance, the web application should be linked with log files. The application should possess the capability to write log files to record the basic process of the application and the error messages to help the maintainer monitor and modify the application if needed.

As the limitation of devices, the web application is only tested on PCs, iPads and Android Smartphones. It should also be tested on different mobile devices to assure its platform independency.



# List of Figure

Figure 1.3.1 Structure of the Thesis.....	7
Figure 2.2.1 the Basic Structure of the system.....	10
Figure 3.1.1.1 User Interface of Attendant Station.....	16
Figure 4.1.1 General Architecture.....	19
Figure 4.2.1.1 the Percentage of Popularity by Use.....	22
Figure 4.2.1.2 the Percentage of Page Visitors.....	22
Figure 4.3.1.1 the Structure of the System.....	28
Figure 4.3.2.1 User Case Diagram.....	29
Figure 4.3.3.1 Sequence Diagram.....	34
Figure 4.3.4.1 Activity Diagram.....	35
Figure 4.3.5.1 Class Diagram.....	36
Figure 6.1.1.1 Main UI.....	47
Figure 6.1.1.2 Log in.....	48
Figure 6.1.1.3 WebSocket check.....	48
Figure 6.1.1.4 Switch the Subpanels.....	48
Figure 6.1.1.5 Panel Switch.....	48
Figure 6.1.1.6 Alert Window.....	48
Figure 6.1.1.7 Password Input Keyboard.....	48
Figure Appendix Technology Selection Map.....	63

# List of Table

Table 4.2.1.1 the Number of Each Category.....	21
Table 5.1.1.1 UI Component Introduction.....	37
Table 5.2.1.1 UDP Communication Component Introduction.....	42
Table 5.3.1.1 WebService Component Introduction.....	43
Table 5.4.1.1 Camera Component Introduction.....	45
Table 6.1.2.1 the Test Result.....	49

# Bibliography

- [JK1] Jennifer Kyrnin, About.com Guide, “web application” [online accessed: April 3<sup>rd</sup>, 2012] available from <http://webdesign.about.com/od/web20/g/web-application-definition.htm>
- [WIK1] Wikipedia, “Web browser engine” [online accessed: April 3<sup>rd</sup>, 2012] available from [http://en.wikipedia.org/wiki/Web\\_browser\\_engine](http://en.wikipedia.org/wiki/Web_browser_engine)
- [W3C1] W3C, “What is HTML?” [online accessed: April 3<sup>rd</sup>, 2012] available from <http://www.w3.org/MarkUp/>
- [ISU1] ISGMLUG, “SGML” [online accessed: April 3<sup>rd</sup>, 2012] available from <http://www.isgmlug.org/>
- [WIK2] Wikipedia, “WYSIWYG” [online accessed: April 3<sup>rd</sup>, 2012] available from <http://en.wikipedia.org/wiki/WYSIWYG>
- [W3C2] W3C, “What is CSS?” [online accessed: April 8<sup>th</sup>, 2012] available from <http://www.w3.org/Style/CSS/>
- [W3S1] W3schools, “What is JavaScript?” [online accessed: April 8<sup>th</sup>, 2012] available from [http://www.w3schools.com/js/js\\_intro.asp](http://www.w3schools.com/js/js_intro.asp)
- [ECL1] Eclipse, “What is Eclipse and the Eclipse Foundation?” [online accessed: April 16<sup>th</sup>, 2012] available from <http://www.eclipse.org/org/>
- [WIK3] Wikipedia, “LAN” [online accessed: June 2<sup>nd</sup>, 2012] available from [http://en.wikipedia.org/wiki/Local\\_area\\_network](http://en.wikipedia.org/wiki/Local_area_network)
- [JQU1] jQuery, Introduction on the official website [online accessed: April 20<sup>th</sup>, 2012] available from <http://jquery.com/>
- [MOO1] MooTools, Introduction on the official website [online accessed: April 20<sup>th</sup>, 2012] available from <http://mootools.net/>
- [PRO1] Prototype, Introduction on the official website [online accessed: April 20<sup>th</sup>, 2012] available from <http://www.prototypejs.org/>
- [YUI1] YUI, Introduction on the official website [online accessed: April 20<sup>th</sup>, 2012] available from <http://yuilibary.com/>
- [ACT1] Microsoft, “What is ActiveX control?” [online accessed: May 4<sup>th</sup>, 2012] available from <http://www.microsoft.com/security/resources/activex-what-is.aspx>

- [PG1] Paul Gil, About.com Guide "What is 'Flash'? Is that the same as 'Adobe Flash'?" [online accessed: May 4<sup>th</sup>, 2012] available from <http://netforbeginners.about.com/od/f/f/flash.htm>
- [SUN1] Sun, "What Is a Servlet?" [online accessed: May 7<sup>th</sup>, 2012] available from [http://java.sun.com/j2ee/tutorial/1\\_3-fcs/doc/Servlets2.html#75087](http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/Servlets2.html#75087)
- [SUN2] Sun, "JavaServer Pages Technology" [online accessed: May 7<sup>th</sup>, 2012] available from [http://java.sun.com/j2ee/tutorial/1\\_3-fcs/doc/JSPIntro.html](http://java.sun.com/j2ee/tutorial/1_3-fcs/doc/JSPIntro.html)
- [WIK4] Wikipedia, "Polling" [online accessed: May 14<sup>th</sup>, 2012] available from [http://en.wikipedia.org/wiki/Polling\\_\(computer\\_science\)](http://en.wikipedia.org/wiki/Polling_(computer_science))
- [WIK5] Wikipedia, "Long Polling", [online accessed: May 14<sup>th</sup>, 2012] available from [http://en.wikipedia.org/wiki/Push\\_technology#Long\\_polling](http://en.wikipedia.org/wiki/Push_technology#Long_polling)
- [WEB1] WebSocket.org, "What is WebSocket?" [online accessed: May 17<sup>th</sup>, 2012] available from <http://www.websocket.org/>
- [JET1] Jetty, Introduction on the official website [online accessed: May 20<sup>th</sup>, 2012] available from <http://jetty.codehaus.org/jetty/>
- [JWE1] jWebSocket, "What is jWebSocket?" [online accessed: May 21<sup>st</sup>, 2012] available from <http://jwebsocket.org/>
- [W3S2] W3schools, "Introduction to Web Services" [online accessed: May 24<sup>th</sup>, 2012] available from [http://www.w3schools.com/webservices/ws\\_intro.asp](http://www.w3schools.com/webservices/ws_intro.asp)
- [W3S3] W3schools, "The XMLHttpRequest Object" [online accessed: May 30<sup>th</sup>, 2012] available from [http://www.w3schools.com/xml/xml\\_http.asp](http://www.w3schools.com/xml/xml_http.asp)
- [JSO1] JSON, "Introducing JSON" [online accessed: June 1<sup>st</sup>, 2012] available from <http://www.json.org/index.html>
- [GIT1] GitHub. Code "get\_barcode\_from\_image.js" [online accessed: June 5<sup>th</sup>, 2012] available from <https://gist.github.com/2367689>
- [JN1] Jakob Nielsen, "Ten Usability Heuristics" [online accessed: June 7<sup>th</sup>] available from <http://web.cs.wpi.edu/~kal/courses/hci/module6/neilsen10heuristics.htm>
- [OMI1] Steve Crouch, Rob Baxter, Tim Parkinson from OMII-UK, "OMII-UK Software Evaluation Process" Revision: 0.16 [online accessed: June 8<sup>th</sup>] available from <http://www.omii.ac.uk/attach/SoftwareEvaluation/OMII-UKSoftwareEvaluationProcess-v0.16.pdf>

# Glossary

- [1] UDP: The User Datagram Protocol is a popular protocol in the Internet. It is usually mentioned with Transmission Control Protocol (TCP). It provides a connectionless transmission. The data is encapsulated into a datagram. It is a fast but unsafe protocol.
- [2] Rendering engine: "It is a software component that takes marked up content (such as HTML, XML, image files, etc.) and formatting information (such as CSS, XSL, etc.) and displays the formatted content on the screen" WIK1.
- [3] SGML: "SGML and XML are met languages - languages for describing other languages - which let users design their own customized markup languages for limitless different types of documents. SGML is very large, powerful, and complex. It has been in heavy industrial and commercial use for over a decade, and there is a significant body of expertise and software to go with it. XML is a lightweight cut-down version of SGML which keeps enough of its functionality to make it useful but removes all the optional features which make SGML too complex to program for in a Web environment" ISU1.
- [4] WYSIWYG: "WYSIWYG is an acronym for what you see is what you get. The term is used in computing to describe a system in which content (text and graphics) displayed onscreen during editing appears in a form closely corresponding to its appearance when printed or displayed as a finished product, which might be a printed document, web page, or slide presentation" WIK2.
- [5] Entity: A scanned item that added into the list is called an entity
- [6] LAN: "A local area network (LAN) is a computer network that interconnects computers in a limited area such as a home, school, computer laboratory, or office building using network media" WIK3.
- [7] Model Architecture: The user can set up a model to manage the data and operation of a server. For example, `new newModel=new Y.Model ({attributes})` can create a model named "newModel". Then the user can access the attributes by `newModel.get (attributes)`.
- [8] XHR: "The XMLHttpRequest object is used to exchange data with a server behind the scenes. It can update a web page without reloading the page, request data from a server after the page has loaded, receive data from a server after the page has loaded and send data to a server in the background" W3S3.

[9] JSON: “JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language.” JSO1

[10] z-index: z-index decides the level of the component in the stack. Higher z-index means this component has higher level and will be displayed upper than lower z-index component.

# Appendix A

## Technology Selection Map

The figure shows the whole process of technology selection

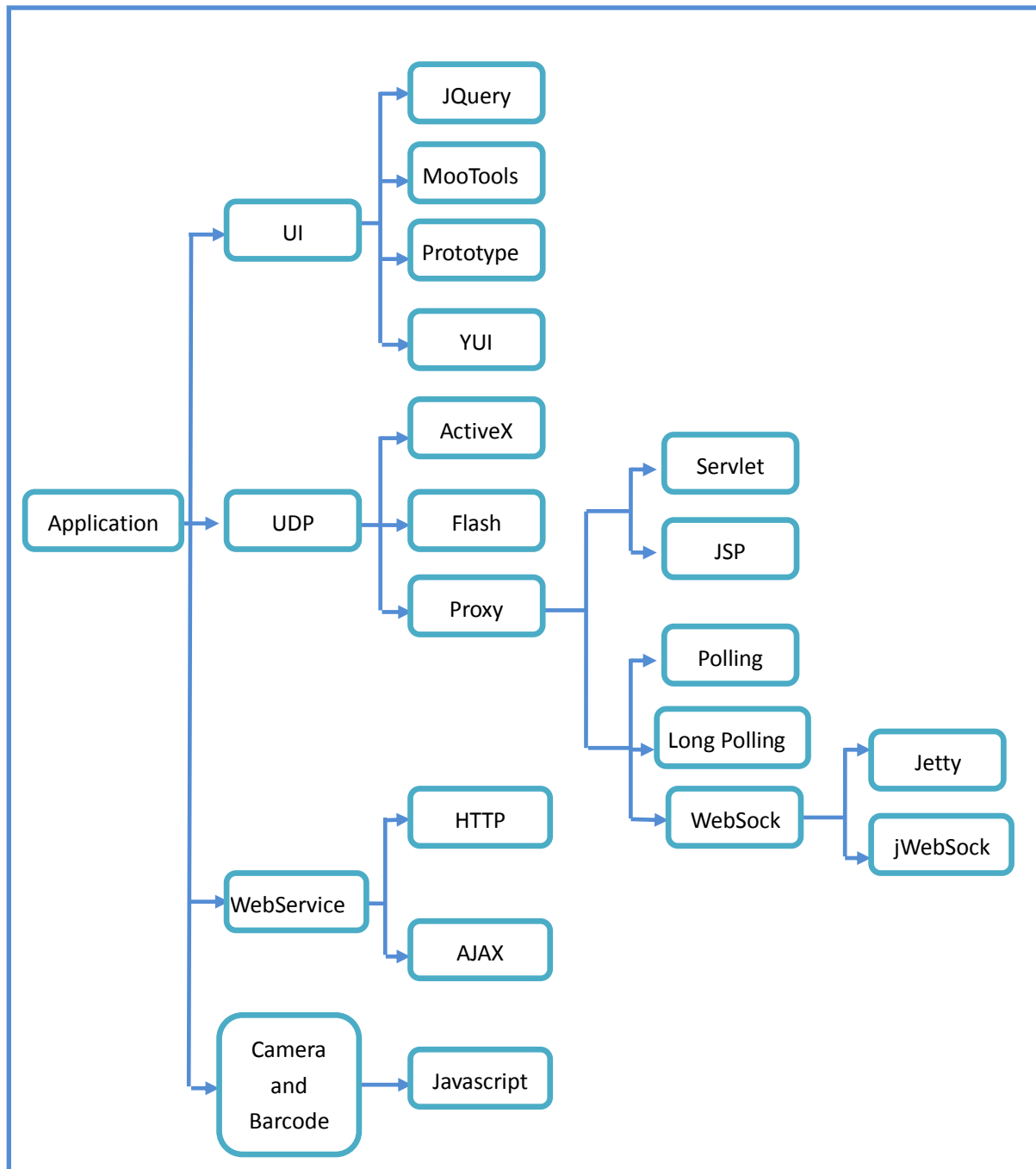


Figure Appendix Technology Selection Map