
**Hochschule Darmstadt**

- Faculty of Computer Science -

# Design a mobile logistics solution utilising modern Software and Services

**Bachelor thesis**
submitted in fulfilment of the requirements for the degree of
Bachelor of Science (B.Sc.)

13 August 2012

submitted by
Viet-Cuong Cao

| | |
|---|---|
| Examiner: | Dr. Christian Baun |
| Co-examiner: | Prof. Dr. Alois |

# Erklärung

Ich versichere hiermit, dass ich die vorliegende Arbeit selbständig verfasst und keine anderen als die im Literaturverzeichnis angegebenen Quellen benutzt habe.

Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder noch nicht veröffentlichten Quellen entnommen sind, sind als solche kenntlich gemacht.

Die Zeichnungen oder Abbildungen in dieser Arbeit sind von mir selbst erstellt worden oder mit einem entsprechenden Quellennachweis versehen.

Diese Arbeit ist in gleicher oder ähnlicher Form noch bei keiner anderen Prüfungsbehörde eingereicht worden.

Frankfurt, den 13 August 2012

Viet-Cuong Cao

# Abstract

Many technology implementations are nowadays used as helpful tools for the business world. Beside computer software, mobile applications are considered as important innovation resources.

The subject of this thesis is the design of a solution that would help to improve the performance of delivering food orders in Restaurants.

To achieve this goal, research on various online ordering systems will be carried out and introduced. A mobile application will then be built. This application is the focus of the thesis.

The idea with the mobile application is to create utilities for the delivery driver to use in his mobile phone:

At first, the delivery-driving route will be calculated, based on the location of the customer through using Google Map API (Application Programming Interface) Web Services. These routes are sorted and displayed on the mobile device of the driver. He then just follows the route on his device, optionally using turn-by-turn navigator application.
Second, the application on the mobile device also provides tools for cashiering.
Finally, it communicates with a server to give customers the order's status.

# Acknowledgement

Foremost, I would like to thank Dr. Christian Baun for his kindly help during the writing of this thesis.

I would also like to thank Professor Dr. Alois Schütte as my second examiner.

I wish to thank my brother for giving me the motivation to complete the writing.

My special gratitude is heartily expressed to my mother and my girlfriend, who have stayed by side with me and never hesitate to give me any support.

# Contents

# List of figures

# Introduction

## Motivation

Apple reported, on June 2012, that since the inception of App Store, 30 billion apps have been downloaded [11]. At the same time, Xyologic also estimated that the total mobile applications on four popular platforms are about 1.5 million and that there are currently 400,402 publishers [12]. More and more mobile applications are downloaded and uploaded daily. Users have now become acquainted with this type of software: the mobile app.

Due to the mobility and the ability for calculating, mobile apps on mobile devices seem to be very helpful in everyday life.
Furthermore, when integrating with modern online services, mobile apps have the potential to be a useful and powerful resource for every business.

This thesis will discuss and introduce some technologies into building a mobile application for a restaurant, the application will also integrate with Google Map API Web Services provided by Google.
The thesis has one goal: to simplify daily tasks by utilizing modern software and services.

## Outline of the thesis

1. **State of the art:** this chapter describes research on the actual logistics situation by restaurants: three online ordering services will

be introduced; one of them has invented an innovative device, the Lieferheld Express Box. The chapter then refers to the need for a tool for delivery drivers, and introduces a sample mobile application. Finally, the chapter summarises the three most currently used mobile systems.

2. **Types of mobile applications:** this chapter describes three types of mobile applications: native app, web app and hybrid app.
3. **Development of a hybrid app for the delivery driver:** the whole process of creating the mobile application is presented in this chapter. At first, the requirements of the application are analysed and a solution is suggested. Then, the concept of using web technologies to build the mobile application is discussed. After that, this chapter describes the implementation process including packaging the application into a hybrid app with Phonegap. At the end, the evaluation of the app is made.
4. **Summary:** this writing ends here by giving a summary of the thesis.

# 1 State of the art

## 1.1 Logistics solution

Logistics plays one of the most important roles in any business that deals with product delivery. This also applies to the food service industry.

In a restaurant, customers mostly want to give orders quickly and expect to receive the delivery in the shortest possible time. On the other hand, the restaurant tries to serve as many orders as possible to increase sales.
An optimal process from receiving orders to delivering food always helps to improve the quality of out-of-house service. It also benefits the restaurant owner.

Aware of this issue, a few companies have built services to help restaurants and customers communicate with each other more smoothly, and in ways that are more convenient.

### 1.1.1 Online food ordering service

Currently, there are three big names in Germany providing online food ordering services for restaurants. The very first and well-known “Pizza.de” founded in January 2007, is dominating this business segment. Two similar companies that have subsequently opened are “Lieferheld”, founded in November 2010 and “Lieferando”, launched in January 2011. [13]

These companies have almost the same concept: they serve users’ orders through a website. This website can be opened on a customised mobile application. It lists all the restaurants with their menus. Users just browse through

and choose favourite foods, then complete the order process and wait for the delivery.
For example, on “Pizza.de”, a user goes through the steps below:

1. *Opens internet browser and goes to http://pizza.de or downloads and installs the application “pizza.de” from app market1*
2. *Enters the zip code and starts the search. After the webserver finishes searching, it lists the restaurants, that deliver to the zip code entered*
3. *Clicks on the restaurant’s name to go to the restaurant’s page*
4. *Chooses food from the restaurant’s menu*
5. *Enters delivery address and payment method*
6. *Finally clicks on button “Order now” to finish the order*
7. *Waits for confirmation via E-Mail or SMS*[2]

Users can also register with the website, so that they must only enter their address and payment information once. The next time, when placing an order, they just log in with their usernames and all the registered information will automatically pop up in the relevant boxes.
The confirmation at step 7 is only to confirm that the order has been recorded and that it is being transferred to the restaurant. This confirmation is from the ordering service, not from the restaurant.

At the other end, the restaurant will receive a fax (and optionally an email) about the order. In any event, if the restaurant wants to contact the customer, the restaurant employee must call him on the phone.

[1] App market is an online location, where users can download new applications for their smartphone. The name of this market is not the same between different phone providers: by Apple, it is called Apple App Store; by Google, it is called Google Play Store; by Microsoft, it is called Windows Phone Marketplace; and by RIM it is called BlackBerry App World.

[2] SMS: Short Message Service

The pictures below are screenshots from the app[3] “pizza.de” on mobile phone:

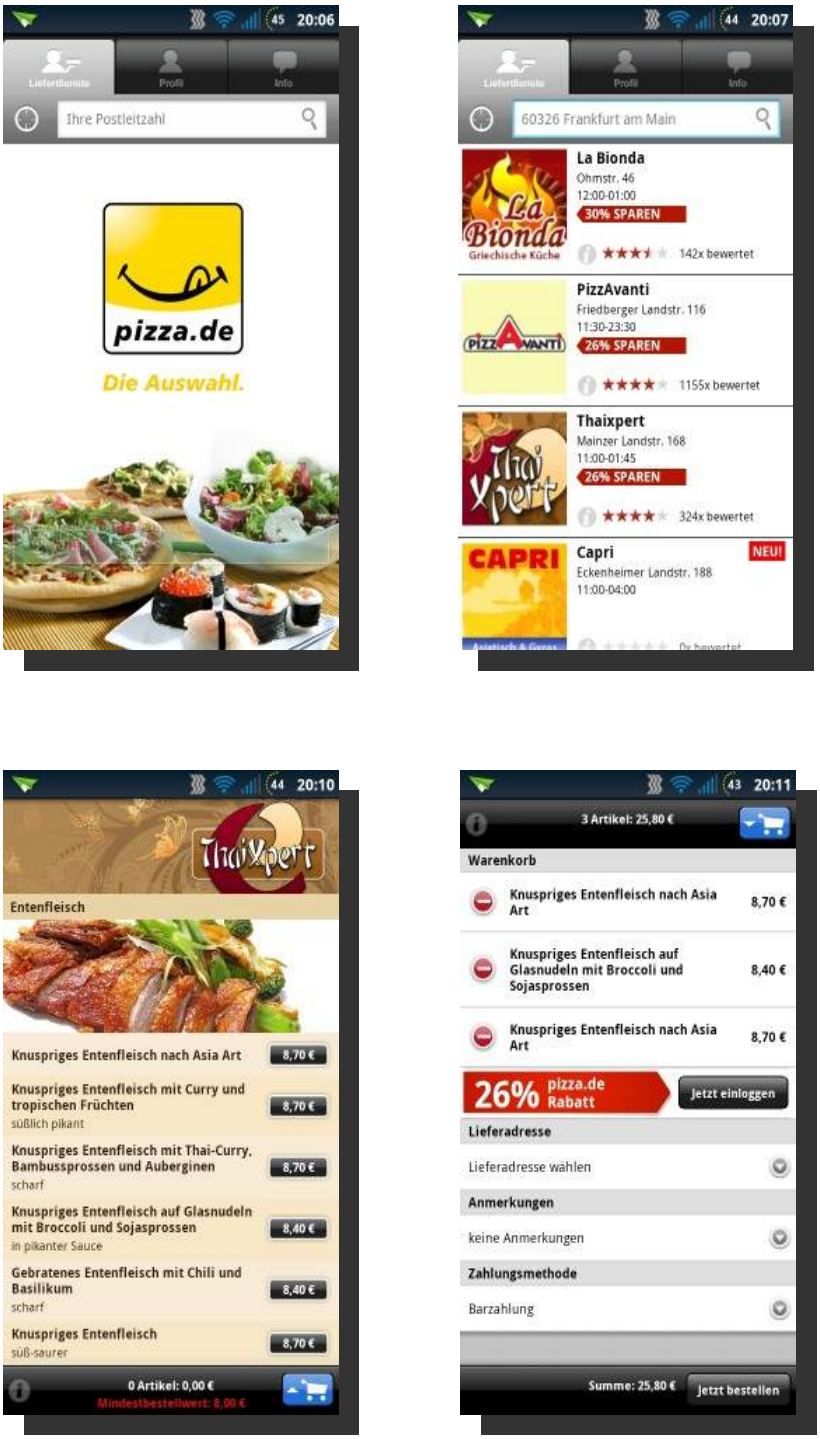



*Figure 1: Screenshots from application “pizza.de” on a mobile phone*

[3] App: Application

As described above, restaurants receive orders not directly from customers but through a third party, in this case, the online food ordering service. By using this service, the restaurant has some benefits as follows:

- *The web page of the restaurant is professionally set up by the online ordering service company in a consistent way, so that customers can easily choose their favorite foods.*
- *Reduction of the amount of work the employee has on receiving orders directly from customers per telephone.*
- *The restaurant is simply listed on the Website of online food ordering service, which is well advertised to the market.*
- *Good restaurants with high reputations will be presented by good ratings.*

It is obvious that online food ordering service is helpful. However, there are still some inconveniences in the process.
When finishing the order, the customer does not know how long it will take to have food delivered. He just receives only the confirmation from the ordering service that confirmming that the order has been made. Otherwise, he must contact the restaurant directly if he wants to get more information.
Furthermore, transferring orders by fax or email does not always run smoothly: paper jam by the fax machine, quiet fax signal, late arriving email, etc. can possibly occur.

### 1.1.2 UMTS Express Box

Lieferheld CEO[4] Fabian Siegel really wanted to improve the quality of the order process. He and his staff then took a significant step by inventing the UMTS[5]

[4] CEO: Chief Executive Officer
[5] UMTS: Universal Mobile Telecommunications System

Express Box. This innovative product does what it is expected to do: fills the gap between customers and restaurants. [14]

The Express Box is a M2M solution; it means communication from Machine to Machine. This box has a small thermo printer plus some simple buttons, and functions without the need of internet connection or any computer aid. The power cable needs only to be plugged into the electric socket, and the box will automatically connect to the mobile communication network itself and be ready for receiving orders from customers. [15]

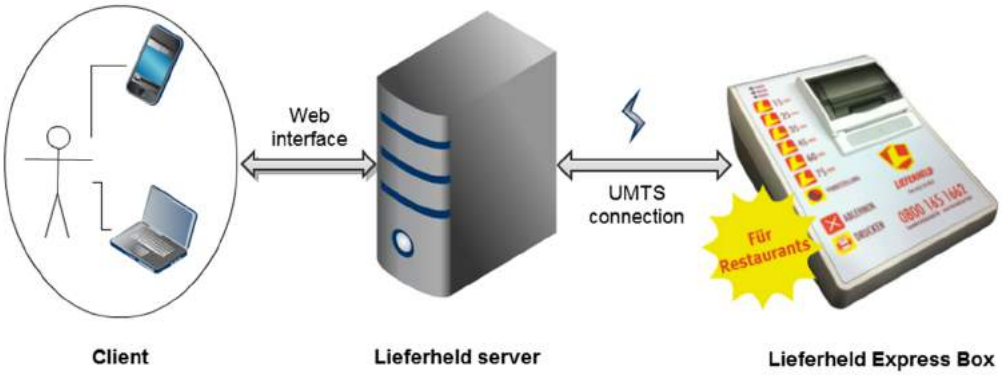



*Figure 2: The way UMTS Express Box works*

The customer browses to the web of Lieferheld, carries out the ordering process as usual; the webserver will send the order to the Express Box located at the restaurant. At this point, the box prints out a receipt with the customer's order on it. The restaurant can directly confirm the order and give the estimate delivery time by pressing one button on the Express Box. Via UMTS connection, the confirmation comes right away back to web server and forwarded to customer per SMS.

This box costs 99€, the restaurant owner has to pay only once and does not need to pay for UMTS connection. 1500 out of 5800 restaurant members from Lieferheld have already used the box [16].

By using this box, restaurants eliminate the unstable issue of fax machine or email. The customers get a clear confirmation and feel more comfortable.

### 1.1.3 Tool for delivery driver

A delivery service primarily consists of at least one delivery driver. The delivery works like an off-site waiter, out of the restaurant. For this reason, improving the quality of work of the driver also leads to better performance on serving restaurant's customers. This, besides a good ordering service, also has to be taken into consideration.

In most of the restaurants that offer a take-away delivery food service, the delivery driver takes the finished cooked food from the kitchen and packs it into his car, then drives to the addresses of the customers. He is the one who controls which route to take based on his driving experience. He also calculates the changes quickly in mind when receiving payments. The payments will be then summed up at the end of the day with the restaurant owner.
All these processes are common at the restaurants. They could be optimized in a way that reduces the amount of work each individual undertakes and speeds up the repeating procedures. Creating software, which can run on the mobile phone of the driver and do the repetitive tasks, is a good solution.

An example of this software is the application "Delivery Droid", written by Marc Holder Kluver for mobile phones run on Android system. [17]
This application has some functions that could ease the delivery process like archiving delivered orders, calculating payment tip, driving map, etc.
However, it requires the driver to manually input the orders' details, and it lacks the ability to seek out the best route through many destinations. These features could be enhanced or completed, thereby helping the driver a lot.
The pictures below show some screenshots of the application "Delivery Droid"

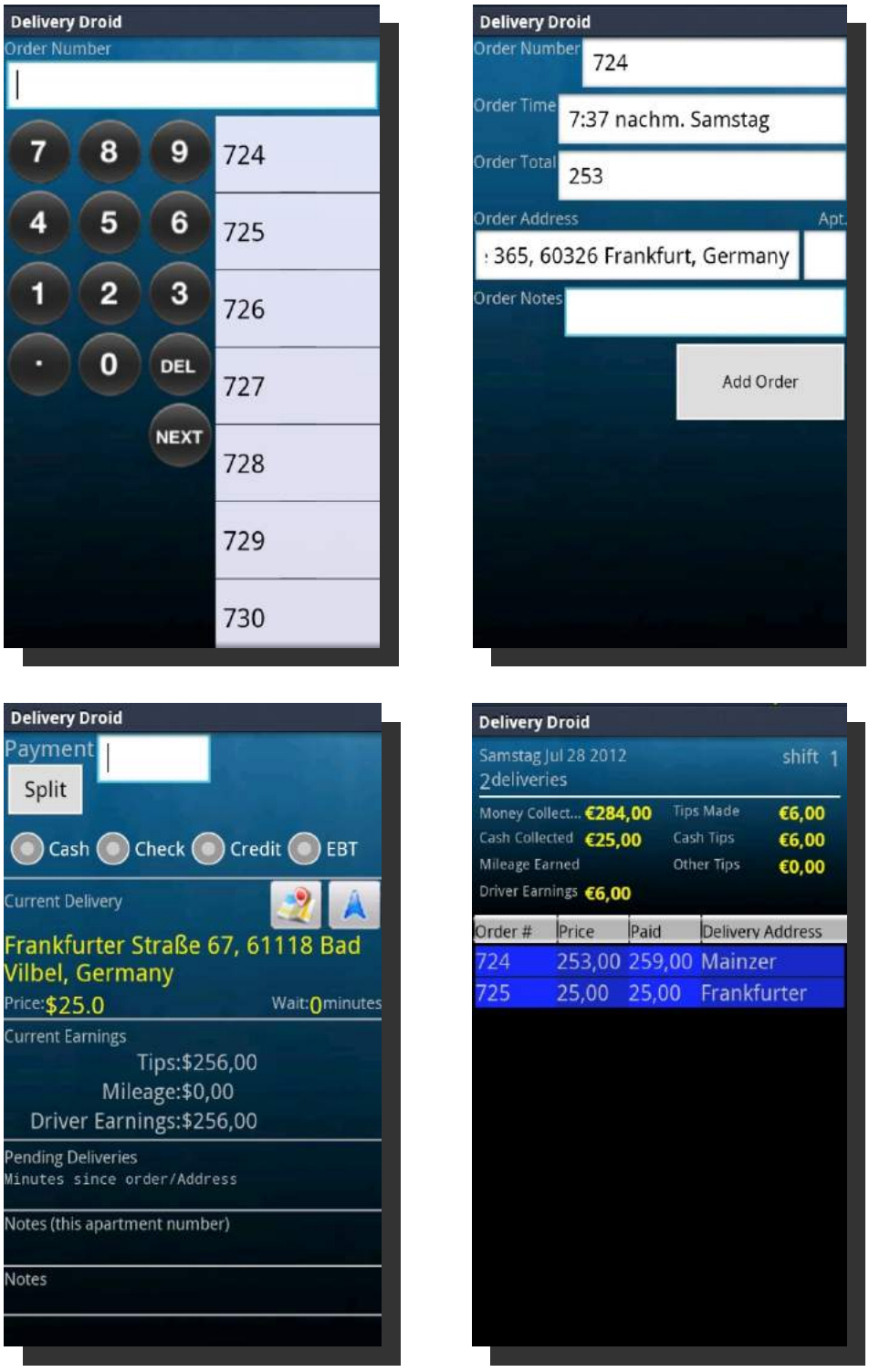



*Figure 3: Screenshots of the tool for the delivery driver*

## 1.2 Mobile Systems

According to [18], in the first quarter of 2012 there were 419.1 million mobile phones sold to end users worldwide. These mobile phones work based on different platforms.

Standard cell phones usually run on an operating system fixed to the hardware. The vendors produce these phones together with the supplied applications and sell them as a package to customers. The system like a black box are closed to extern developers; extra new applications should normally built on another application development platform supported by the phone.

On the other hand, modern phones are often delivered with powerful hardware; they normally run on a flexible platform, which has APIs opened to all the developers. Developers can easily build more native applications by using the SDK[6] released by the operating system vendor. These modern phones are called smartphones.

The graphic below gives an overview of the market shares of operating systems on Smartphones in 2012 and a forecast for the year 2016.

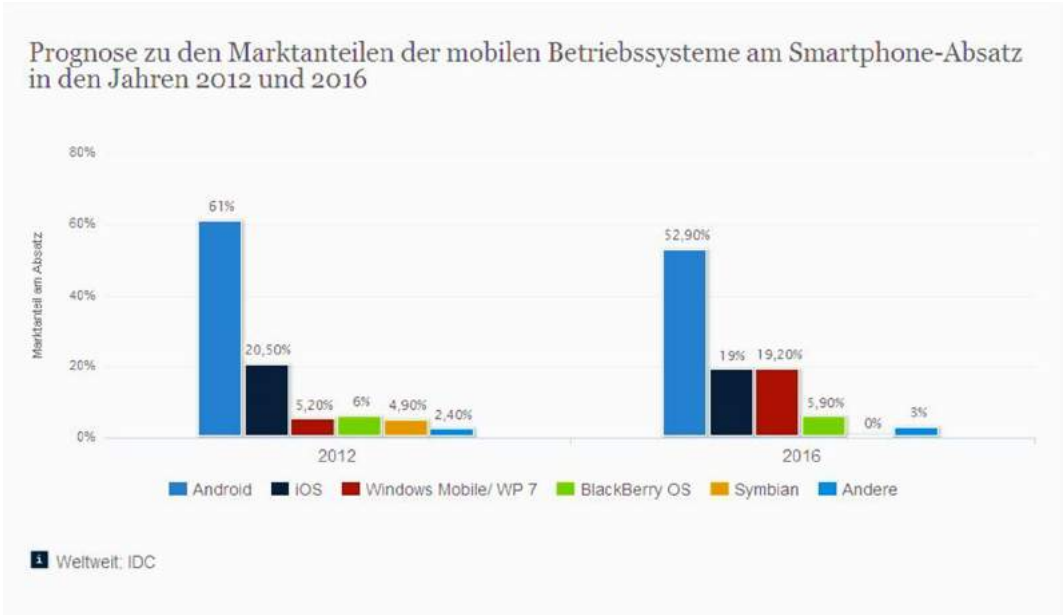



*Figure 4: Forecast of worldwide market shares for mobile operating system in years 2012 and 2016. [1]*

[6] SDK: Software Development Kit

Some of the most popular platforms for smartphones are Android, iOS, Windows Phone and Blackberry OS[7]. The others are either incompatible for building smartphones or not so innovative and subsequently have fewer shares on the market. This section will introduce three platforms, which are expected to grow on the market in future.

### 1.2.1 Apple iOS

January 2007, iPhone appeared on the market from the best-known hardware company in the world, Apple. At that time, iPhone was the very first generation of smartphones, which changed the way of using mobile phone. Steve Jobs, the CEO of Apple reportedly said, “Every once in a while a revolutionary product comes along that changes everything” [19]
The success of this phone was not only due to the hardware but also to the operating system inside it.

iPhone ran on the operating system Mac OS X. At first, the phone worked only with Apple supplied applications. On March 2008, the SDK was been released [20]; developers could already build their own native applications for the system. On 6th June 2010, Apple renamed the operating system from iPhone OS to iOS and announced the release of version 4 with 1500 new features [21].
Native applications for iOS are written in Objective C with xCode as the development environment. Xcode includes an IDE[8] and a simulator; it runs only on Mac OS [22]

### 1.2.2 Google Android

A group of companies led by Google then created Android [23]. This first truly open source operating system for mobile devices has been developed since 5th November 2007. Some of the group founding members that have joined the development are T-Mobile, HTC, Qualcomm, and Motorola.[24].

[7] OS: Operating System
[8] IDE: Integrated Development Environment

Android is based on the Linux Kernel, and utilises a customized virtual machine named Dalvik, which was designed for low memory requirement. One of the openness by Android for end users as developers is the ability to write applications using the very same resources that core applications were using. Developers have the equal rights to access all capabilities of the phone. [25]
Native applications for android are written mostly with programming language Java and Android SDK in an Eclipse IDE, which has the ADT[9] plugin installed [26].

### 1.2.3 Microsoft Windows Phone

Windows Phone was developed by Microsoft to replace its predecessor Windows Mobile. At Mobile World Congress 2010, CEO Steve Ballmer announced this series of Operating System [7].
A notable difference of Windows Phone OS is its design, the Metro design style. It brings a new approach to user experience through a novel way of using colour, typography and motion [27].
Every programming language supports that .NET[10] framework can be used to write applications for Windows Phone. IDE Visual Studio, which includes a Phone Emulator, is also used for developing apps.
It is also to be noted that Microsoft requires phone manufactures to follow a strict hardware specification. This ensures the system works consistently, independent from hardware manufactures. [29]

### 1.2.4 Summary

Each of the systems above has advantages and disadvantages. The table below will summarise some properties of these Systems:

[9] ADT: Android Development Tools

[10] .NET is a software framework develped by Microsoft. It has a big class library, which can be used accrossed different supported programming languages (e.g. ASP.Net, C#, Visual Basic.Net) [28].

| | **iOS** | **Android** | **Windows Phone** |
|---|---|---|---|
| **Companies behind** | Apple | Open Handset Alliance | Microsoft |
| **Common programming languages** | Objective C | Java | .NET |
| **Common IDE** | Xcode | Eclipse (with ADT plugin) | Visual Studio |
| **Actual SDK** | iOS SDK 5.1 | Android SDK 20.0.1 | Windows Phone SDK 7.1 |
| **Important aspect** | Development environment for native app works only on MAC OS | 3th party developers have full access to phone's capabilities | Strictly hardware specifications |

*Figure 5: Overview of three most prominent operating systems on the mobile market [2],[3],[4]*

# 2 Types of mobile applications

Applications for mobile phones do not always reside on the phone. In some situations, the app stays on a server and users access it through a browser. There are three types of applications: native app, web app, and hybrid app. Each of them has advantages and disadvantages when used in a specific context.

## 2.1 Native app

Native apps are built specifically for a particular platform or mobile device. This type of apps is installed directly on the device and can access all the hardware features provided through the platform's APIs.
Creating native app often needs more overhead than by the others. Sybase Inc. stated here [30] that some companies found that building an native app costs three times as much as building a hybrid app.

**Android**

To develop a native app for Android, the developer uses programming language Java to write code and compiles it with Android SDK tools. The product then will be packaged into a file with the extension .apk. This file is the app, which needs to be installed on the device.
Each app runs in a separate process and has its own instance of the Dalvik virtual machine. The java classes are transformed into .dex format and executed by the VM[11]. The VM sits on top of a Linux kernel, which carries our the core services, e.g. hardware drivers, process and memory management, etc. [4], [5]

[11] VM: Virtual Machine

Below is an overview of android's architecture:

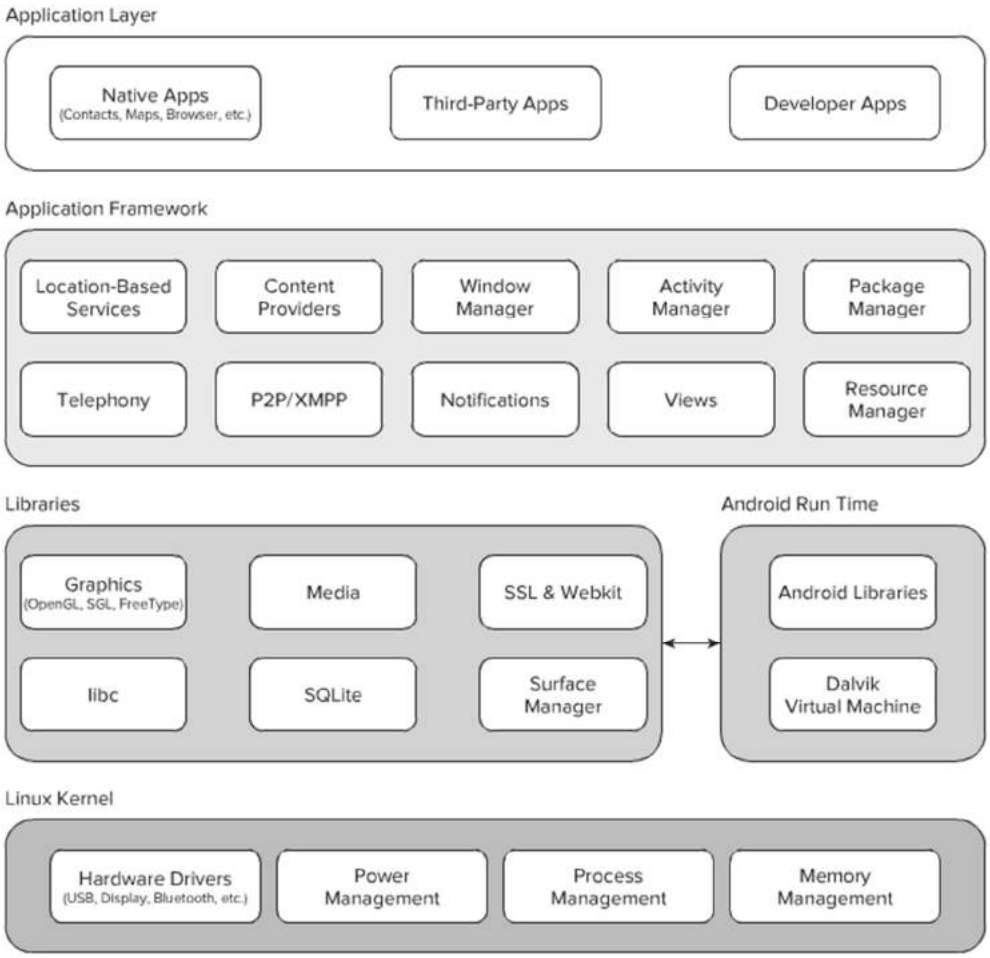



*Figure 6: Android's architecture [5]*

**iOS**

Native apps for iOS are written in Objective C with iOS SDK. iOS abstractly consists of four different software layer as follow:

- *Cocoa Touch is the top layer and has the most abstraction to the physical hardware. It contains the key frameworks for building iOS apps.*
- *Media layer provides iOS with audio, video, animation and graphics capabilities.*

- *Core Services layer contains the fundamental system services that all applications use.*
- *Core Os layer is the nearest layer to the physical hardware. It provides low-level services like networking, memory management, etc.[6]*

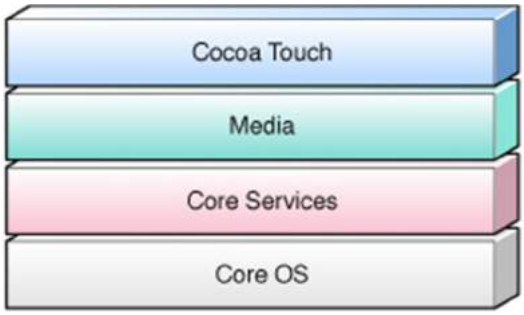



*Figure 7: Layers of iOS [6]*

Developers do not always program for the top layer only; they can also write code to other layers below. However, the lower the layer is, the more line of code should be written, and it leads to more efforts.

**Windows Phone OS**

Microsoft provides Visual Studio as an IDE and Windows Phone SDK for developers to use in creating native Windows Phone apps. All .NET languages can be used to write code but C# is often used.

On top of the .NET framework, the platform consists of two major frameworks: Silverlight and XNA.

Silverlight is used for creating normal business app, whereas XNA is often used for game development. [7]

Below is the runtime component from Windows Phone platform:

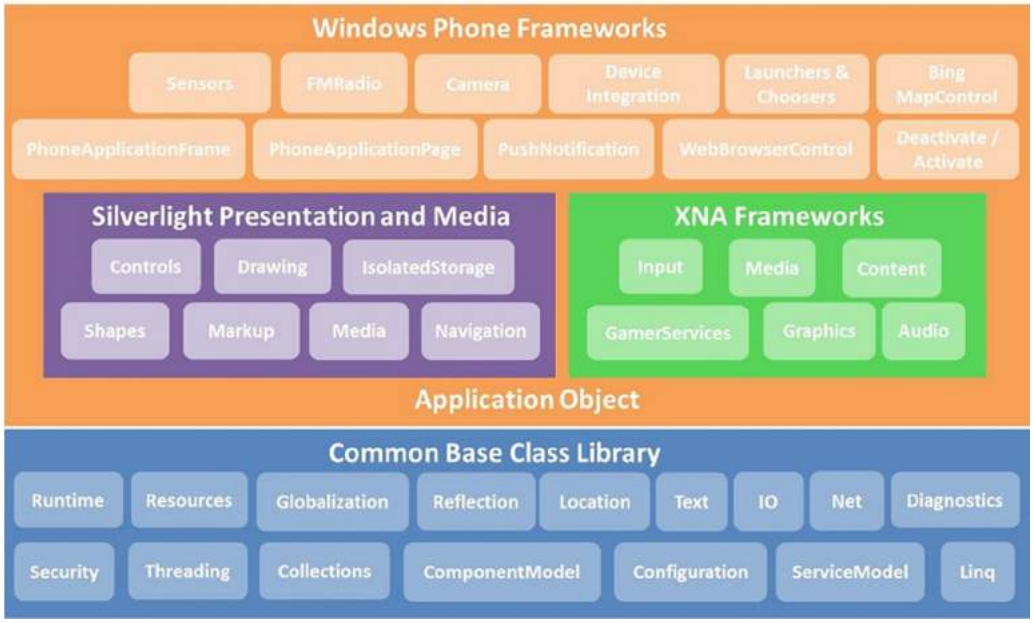



*Figure 8: The runtime component of Windows Phone platform [7]*

In addition, in order to create native apps for a specific device, developers need to learn about many other different operating systems, on which the app is to be installed. Some of them are Symbian, Blackberry OS, Bada, WebOS, etc.
Despite the speed in performance and the native feeling when using this type of app, creating native apps faces a challenge: cross platform compatibility.

## 2.2 Web app

Web apps are not actually new. They are just another term for websites, which were developed based on web technologies like HTML[12], JavaScript, and CSS[13]. The term “Web app” is nowadays often used when talking about a web-site developed specially for a mobile device.
In contrast to native apps, web apps stay on a web server outside of the mobile platform. Users execute a web app by opening the browser and connecting to the server to download the whole or a part of the content of the application.

[12] HTML: Hyper Text Markup Language
[13] CSS: Cascading Style Sheet

Because of this characteristic, that the content of web apps is not directly put on the device but localized on a web server, it is easier for the users to keep the application up to date. A change on the server will be distributed equally to all clients, with the advantage that users do not need to reinstall the application manually each time it is updated.

This feature is also applied to the development process: the web app is only written once, and it can be used for all device platforms, using an internet browser. This great feature is the advantage of web app over native app: platform independence.

Developers often use HTML5 together with jQuery Mobile when developing mobile web app. These techniques have special focus on mobile devices and provide some useful features: offline cache, local storage, video and audio streaming, canvas drawing, geo location API. Almost all modern mobile operating systems support these HTML5 features [31].

## 2.3 Hybrid app

Although the overall performance from native app is better [32], [33], [34], web apps are easier for developers to build, to maintain and can be run on many platforms.

However, the advantage of web apps distributed on the web is also its disadvantage when observed from another point of view: unlike native apps, web apps cannot fully work offline; it requires an internet connection.

Another inconvenience is that, instead of installing the app on mobile devices, users have to open the browser and type in the address of the web app in order to use the app, a big drawback for the user.

Hybrid app is a kind of application, which is developed with web techniques, works like a webbapp but can be installed on the device. The idea behind it is, to take the web app code and wrap it with a native container.

By following this approach, the application can also communicate with the device APIs. Overall, the application can now practically act as a native app.

IBM, in a white paper [8], illustrated the three approaches to build mobile applications:


*Figure 9: Three approaches to build mobile applications [8]*

To build hybrid apps, developers can write their own bridge to connect the native container to the device APIs or use frameworks provided by a third party.

Adobe [35] is one of the companies providing an open source framework to build hybrid apps, PhoneGap. Currently, developers can use framework PhoneGap to build hybrid apps to seven different mobile platforms.

Another framework is Titanium from Appcelerator [36]. However, the development approach by Titanium is a little different: the company offers the framework as a package including an IDE and Titanium SDK; developers write full code in javascript, without using HTML or CSS. The framework will build this code into true native code. Currently, Titanium only supports iOS and Android in its released version.

# 3 Development of a hybrid app for the delivery drivers

## 3.1 Requirements

### 3.1.1 Actual situation

KimChan is a bistro in Dreieich, Germany. Besides indoor services, the bistro also provides food delivery to customers out of house. Currently, KimChan has applied a software program for receiving orders into its ordering system. This software program, however, can only help the employees indoor take orders quickly based on the customer reference number. The actual situation can be described as follow:

1. *Customer calls to bistro to make an order.*
2. *Employee picks up the phone and asks first for a customer reference number. Go to step 5 if the customer has no reference number.*
3. *Employee enters the customer reference number into the ordering program. Then verifies the delivery address from the program with customer.*
4. *Employee notes the customer's order.*

5. *If customer does not have a reference number, employee asks for his/her name, address as delivery address and inputs to the ordering program. The program will generate a reference number for the customer.*
6. *Employee sends order to the kitchen, and the delivery address to the driver at once.*
7. *Kitchen cooks the food from order. When finished, kitchen calls delivery driver to receive food.*
8. *Delivery driver packages food and delivers to the address received from step 6.*

From this process, it can be seen that the delivery driver has no back up from the ordering program. The program only helps customers and employees indoors to speed up the ordering process.
Before delivering food to the customer, the driver must check the delivery address with the driving map hung somewhere in the kitchen. Then, based on his driving experience, he will choose the best route to drive.
Furthermore, at the end of the day, the driver and the restaurant owner will sit together and manually sum up all the receipts to calculate the revenue of one day.

### 3.1.2 Suggested solution

The actual ordering and delivering process by KimChan Bistro can be improved by modern software and services.

It is suggested, that bistro KimChan Bistro registers with one of the online delivery services described in the section 2.1 for receiving orders from customers. By doing this, the restaurant will become better known in the area. Thus, with

the services provided by the online ordering services, it will be easier for the bistro and customers when making the ordering process.

Additionally, a system should be created to help the delivery driver to perform his job better than before. This is also the focus of this thesis.
Below is the design for the suggested system:

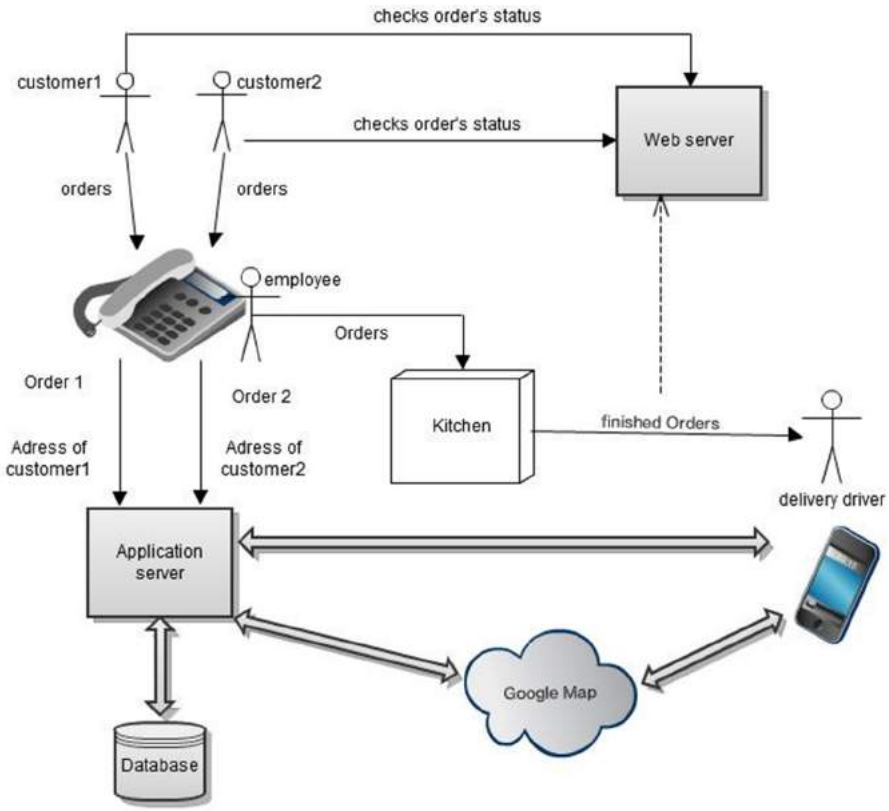



*Figure 10: Design of suggested system*

The suggested system consists of these components:

1. *An application server has an interface to handle the incoming orders*
2. *A database to save customers' information*
3. *A web page for customers to check their orders' status*
4. *A mobile application for the delivery driver*

5. *Google Map Web Services*

#### 3.1.2.1 Interface to handle incoming orders

This interface is a tool for employee to receive orders' information from customers and save them to the database. The interface runs on an application server and should at least have these functions:

- *Manage customers' information*
- *Manage the dishes being provided by the bistro*
- *Manage orders*

The ordering service cannot cover all of the customers, the ordering methods or all the transactions of the restaurant. Therefore, independent of whether the bistro was to register with an online ordering service or not, there would still be a need for this kind of interface.
Actually, KimChan Bistro has already had a software program, which could be extended to integrate with the online ordering service's software. However, this is not a main area of this thesis.

#### 3.1.2.2 Database to save customers' information

This database stores data such as customers' information, orders' information, transactions' details, price list, etc. The database is not only used with the interface mentioned above, but it also communicates with other components in the system: mobile application retrieves delivery order list from this database; the webpage, which displays order's status could also connect to this database as well.

#### 3.1.2.3 Webpage for customers

A customer at home can open this webpage. It runs on a webserver and has one function:

- *Display the status of customer's order*

#### 3.1.2.4 Mobile application for delivery drivers

Mobile application runs on the mobile phone of the delivery driver. It has the functions listed below:

- *Display the list of orders*
- *Calculate the best driving route for delivering food*
- *Check when receiving payment*
- *Display delivery summary*

#### 3.1.2.5 Google Map API Web Services

Google provides an API web service [37] for developers to build applications utilising its powerful services.
Google Map Services assists in planning a better driving route through many destinations.

## 3.2 Concepts

For an appropriate development cost and flexibility in using the mobile application, these components will be used as described below:

- *Apache HTTP[14] server*
- *MySql database*
- *PHP[15] as programming language on HTTP server*
- *HTML5, CSS, jQuery Mobile for developing the web app*
- *Phonegap to package the web app into a standalone program*

[14] HTTP: Hypertext Transfer Protocol
[15] PHP: Hypertext Preprocessor

- *Google Map API Web Services*
- *In addition, the delivery driver has a mobile phone with internet connection*

The following sections describe the components in details:

### 3.2.1 Apache HTTP server

Apache HTTP server is an open source web server; this project is maintained by the Apache Software Foundation [38]. It is the world most used server by active site across all domain names. In comparison with IIS[16], another server from Microsoft [39], Apache HTTP server served 43% domains more in 2011 [40].

In web technology, Apache web server sits between the clients and the web site. It redirects all the requests from a web browser to the resource on the server, and then delivers back the answer as web pages through HTTP protocol.

In the system suggested, Apache HTTP server works as an interface to communicate with the database for inputting and receiving data.

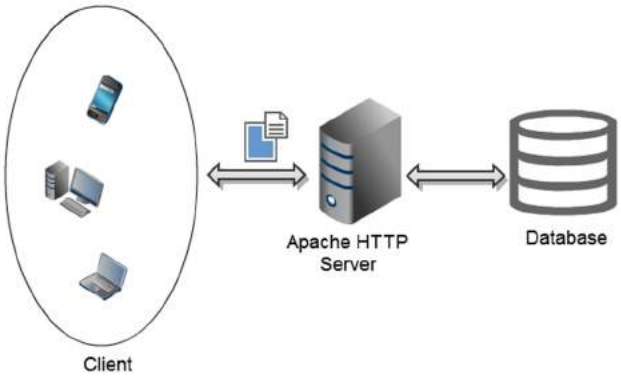



*Figure 11: Apache HTTP Server*

[16] IIS: Internet Information Server

The client in the figure above is a software user agent, transferring a request of the user to the server, based on an HTTP protocol. In the suggested system, the clients are web browsers running on customer's computer, restaurant computer and mobile phone of the delivery driver.

### 3.2.2 MySQL & PHP

Beside Apache HTTP server, MySQL is one of the most successful open source products. It is a relational database management system, which can run on more than 20 operating systems. It has features of high performance, reliability, and ease of use. Many large organisations have been using SQL including Facebook, Google, Adobe. [41]
MySQL stores data in database objects called tables and uses SQL[17] as a query language. Below are some queries that are used in MySQL:

```
1   -- Create a database
2   CREATE DATABASE database_name
3
4   -- Insert data into a table
5   INSERT INTO table_name (column1, column2,...)
6   VALUES (value1, value2,...)
7
8   -- Select data from a table
9   SELECT column_name(s)
10  FROM table_name
11
12  --Update data in a database
13  UPDATE table_name
14  SET column1=value, column2=value2,...
15  WHERE some_column=some_value
16
17  --Delete data in a database
18  DELETE FROM table_name
19  WHERE some_column = some_value
```

*Figure 12: Some queries that are used in MySQL [9]*

To connect to a database and send queries, another language is used in web technologies: PHP[18]. PHP is a server side scripting language; its scripts are ex-

[17] SQL: Structured Query Language
[18] PHP: Hypertext Preprocessor

ecuted on the server. PHP is open source, runs on different platforms and supports many databases, including MySQL.
PHP can be written in a separate file or mixed into the HTML document. A very simple PHP script to output “Hello world” looks like this:

```
1 <?php
2   echo "Hello World";
3 ?>
```

*Figure 13: A very simple PHP script*

To connect to a database, PHP uses this script:

```
1 <?php
2   mysql_connect(servername,username,password);
3 ?>
```

*Figure 14: PHP script to connect to a database*

The combination of Apache HTTP server, MySQL and PHP is often used on a Linux system for Web development. It is usually called a LAMP stack.

### 3.2.3 HTML5, CSS and JQuery Mobile

HTML5 is the fifth version of HTML, a language developed by W3C[19] for use in creating semantic documents on the World Wide Web.
In the early days, HTML documents were in very simple form: they are mostly texts. A great feature of an HTML document was the ability to link different documents easily from one location. This feature is called Hyper Text Link.

As the World Wide Web increases in size and subsequently being used more widely, the specifications and features of HTML must also be changed accordingly. HTML, nowadays, is not only used with simple text pages, it is also built to contain rich media resources. Web pages, which are created with HTML5, function even more intelligently. For example, it can support videos as an element of

[19] W3C: World Wide Web Consortium

the document; developers do not need to embed Flash objects into the page any more.

HTML5 appeared in 2007 when WHATWG[20], a group of technology companies include Apple, Mozilla, Opera was not interested in W3C's direction of continuing the development of XHTML[21], a language with a more strictly defined specification. It was decided that WHATWG stay with HTML standard and to expand it with new features. W3C was then also has interested in the project. [43], [44]

Like HTML, a document created with HTML5 includes some basic elements. A sample structure of HTML5 document looks like this:

```
1  <!DOCTYPE HTML>
2
3  <html>
4
5  <head>
6    <meta charset="UTF-8">
7    <title>Sample document</title>
8    <link href="css/styles.css" rel="stylesheet" />
9    <script src="js/script.js"></script>
10 </head>
11
12 <body>
13  <p>Sample paragraph.</p>
14 </body>
15
16 </html>
```

*Figure 15: A sample HTML5 document*

Creating an HTML5 document is not too difficult, as can be seen from the document shown above:
The first line tells the browser that this is a HTML5 document. The code inside the "< >" are tags; Tags describe semantically what it contains. The content displayed in between a start tag and a close tag, together with the tags, is called "element". A tag can contain an attribute to provide additional information for the element.

[20] WHATWG: Web Hypertext Application Technology Working Group [42]
[21] XHTML: Extensible HyperText Markup Language

The whole code above creates an HTML5 document with the title “Sample document” and the content “Sample paragraph”.

As mentioned in the first paragraph, HTML is used to create semantic documents. Its main task is in describing the content, not formatting it. However many developers did not use it seriously; they used some tags to style their texts, and were unprofessional in their usage of the language.
For formatting texts, W3C provides another language, CSS. A CSS style sheet is a list of definition, which defines how each HTML element will be displayed. This list can be contained inside the HTML document or externally and linked from the HTML.
A sample HTML5 document with CSS looks like below:

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <meta charset="UTF-8">
5     <title>Sample document</title>
6     <link href="css/styles.css" rel="stylesheet" />
7     <script src="js/script.js"></script>
8   <style type="text/css">
9       body{
10          background-color:#fff;
11          text-align:center;
12      }
13      h1{
14          color:blue;
15      }
16      p{
17          font-family:"Times New Roman";
18          font-size:18px;
19      }
20  </style>
21 </head>
22
23 <body>
24   <h1>Sample heading</h1>
25   <p>Sample paragraph</p>
26 </body>
27 </html>
```

*Figure 16: A sample HTML5 document with CSS*

When opened by the browser, the page will look like this:

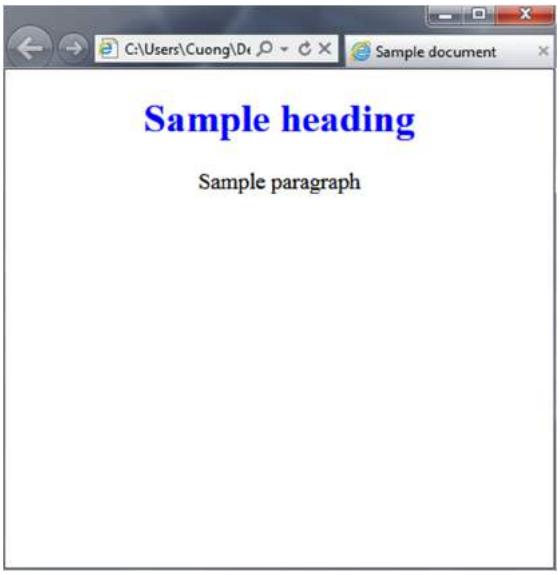



*Figure 17: Sample HTML5 document with CSS opened in the browser*

When bundling with CSS and jQuery Mobile, HTML5 becomes a very good combination for serving rich media content to different platforms, particularly mobile devices.

In addition to the normal functions, HTML5 provides new features to support development website for mobile device. Some of these features are very useful for developing the mobile application:

- *Meta name Viewport: this will resize the viewport of the web site to fit device's width.*
- *Offline Caching: allow users to view web site offline after having visited it before.*
- *Local storage: let the web server to store data on the device so that it can be used later directly from the device. [45]*

If HTML5 is for the content, CSS for the styling, then jQuery Mobile comes to support the behaviour and mix them all to give webapps a look nearly the same as a native app. The jQuery Foundation has introduced jQuery Mobile on their website like this:

> *"A unified, HTML5-based user interface system for all popular mobile device platforms, built on the rock-solid jQuery and jQuery UI foundation. Its lightweight code is built with progressive enhancement, and has a flexible, easily themeable design." [46]*

jQuery Mobile has been widely used in creating native-like mobile web apps. Although, the framework is based on javascript, developers will find it easier to implement than javascript:

The web app must include the jQuery Mobile script in the html code. In the body of html, some predefined elements of the framework will be used to mark the content of web app. jQuery Mobile will find these elements based on the appropriate tag names and attributes. The web app at the end will be rendered to have a nice interface on a mobile device.

Below is a sample jQuery Mobile code [10] and the output of it on a mobile phone:

```
1   <!DOCTYPE html>
2   <html>
3   <head>
4   <title>This is a jQuery sample</title>
5   <meta name="viewport" content="width=device-width, initial-scale=1">
6   <link rel="stylesheet" href="css/jquery.mobile-1.1.1.min.css" />
7   <script src="js/jquery-1.7.1.min.js"></script>
8   <script src="js/jquery.mobile-1.1.1.min.js"></script>
9   </head>
10
11  <body>
12  <div data-role="page">
13  <div data-role="header">
14    <h1>jQuery Mobile</h1>
15  </div><!-- /header -->
16  <div data-role="content">
17    <ul data-role="listview" data-inset="true" data-filter="true">
18      <li><a href="#">Acura</a></li>
19      <li><a href="#">Audi</a></li>
20      <li><a href="#">BMW</a></li>
21      <li><a href="#">Cadillac</a></li>
22      <li><a href="#">Ferrari</a></li>
23    </ul>
24  </div><!-- /content -->
25  </div><!-- /page -->
26  </body>
27  </html>
```

*Figure 18: A sample web app written with jQuery Mobile [10]*

When opening the web app, as displayed above, from a mobile phone, it will look like the screenshot below:

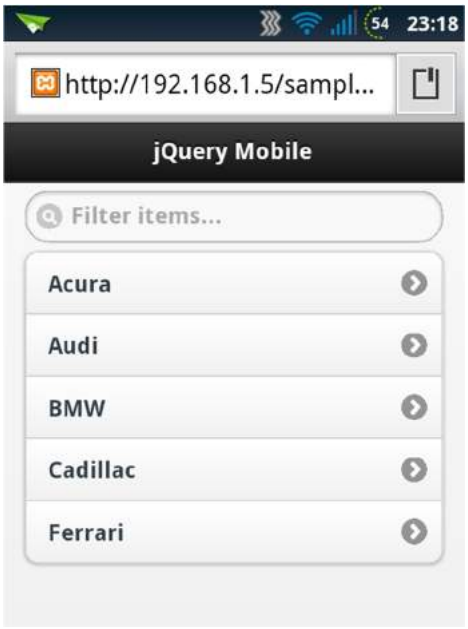



*Figure 19: Screenshot of the sample web app on mobile phone*

### 3.2.4 Phonegap

Phonegap is an open source framework, which is developed by Nitobi but was acquired by Adobe in October 2011 [47]. It helps developers in creating hybrid apps by using usual web technologies. Phonegap provides its SDK named Cordova. The SDK offers API to access many devices' features from Accelerometer to Storage. The main function to be focused upon in this thesis is its ability to convert a web app into a hybrid app so that it can be installed on a mobile phone by the driver.

To use Phonegap, developers generally go through some steps according to which platform the app will run on. Below is a brief description of the steps needed to build hybrid app for Android:

1. *Download and install Android SDK*
2. *Download Cordova*

3. *Set up an android project in IDE Eclipse*
4. *Copy files downloaded from Cordova into appropriate folder*
5. *In Eclipse, edit to include Cordova jar file in the build path*
6. *Make some change in the main java file and AndroidManifest.xml*
7. *Edit index file to include Cordova javascript file*
8. *Run the project [48]*

After running through these steps, it is possible to deploy the application into the mobile device.

Another way to build Phonegap hybrid app more comfortably is by using its online build service [49]: Developers register an account at *http://build.phonegap.com*, then upload the whole web app project to that particular website, choose the destination platform and let the service do all the building process itself.

### 3.2.5 Google Map API Web Services

Google has a very popular online map, which can be freely used by anyone. Together with this map, it provides many tools for developers to use in conjunction with the map.

The Web Services is a collection of http interfaces to Google for getting useful geocoding data. [50]

This thesis will use Direction Service in its implementation. Currently, the service offers developers 2500 requests per day with up to 10 waypoints per request for freely use. [51]

Otherwise, it is to be noted that *"the Directions API may only be used in conjunction with displaying results on a Google map; using Directions data without displaying a map for which directions data was requested is prohibited [51]".*

## 3.3 Implementation

### 3.3.1 Architecture

The app will be developed as a web app based on HTML5, CSS and jQuery Mobile. As for the backend services, PHP together with MySQL will run on Apache HTTP server. At the end, Phonegap will be used to convert the web app into a hybrid app.
A simple web page for customer will also be created using usual web technologies.

The project is divided into two parts: one is at the server side; and the other is at the client side, the mobile device.
KimChan Bistro currently has its software runs with MySQL database on a computer with Windows XP. For a best compatibility, this thesis also chooses windows as the operating system for development. The new database tables should be designed to fit the old structure of the database; so that it will be easy to integrate the app to the system of KimChan Bistro later.

The Apache HTTP server has a distribution for windows. However, to simplify the setup process, this thesis recommends using the software XAMPP [52] as a server solution stack package. This software stack already includes MySQL and supports programming language PHP.

The project is designed that there is an interface at the server side, which communicates with the database and serve the data back to requests from clients. The scripts to build this interface are written in PHP language. These scripts will be placed in a folder named “/services”, it will locate in the subfolder “/htdocs” of the Apache’s directory.
The mobile application has its own structure on a separate directory. This structure is a normal website’s structure. It has html files at the root directory; these files include javascript and css into its header. The javascript files and css files

will locate in the subfolder “/js” and “/css”. Furthermore, there will be a folder “/images” to store the logos and images.

The following subsections contain diagrams describing the project.

#### 3.3.1.1 Use case diagrams

##### 3.3.1.1.1 Mobile application

Below are the use-cases, which exist when the delivery driver uses the mobile application

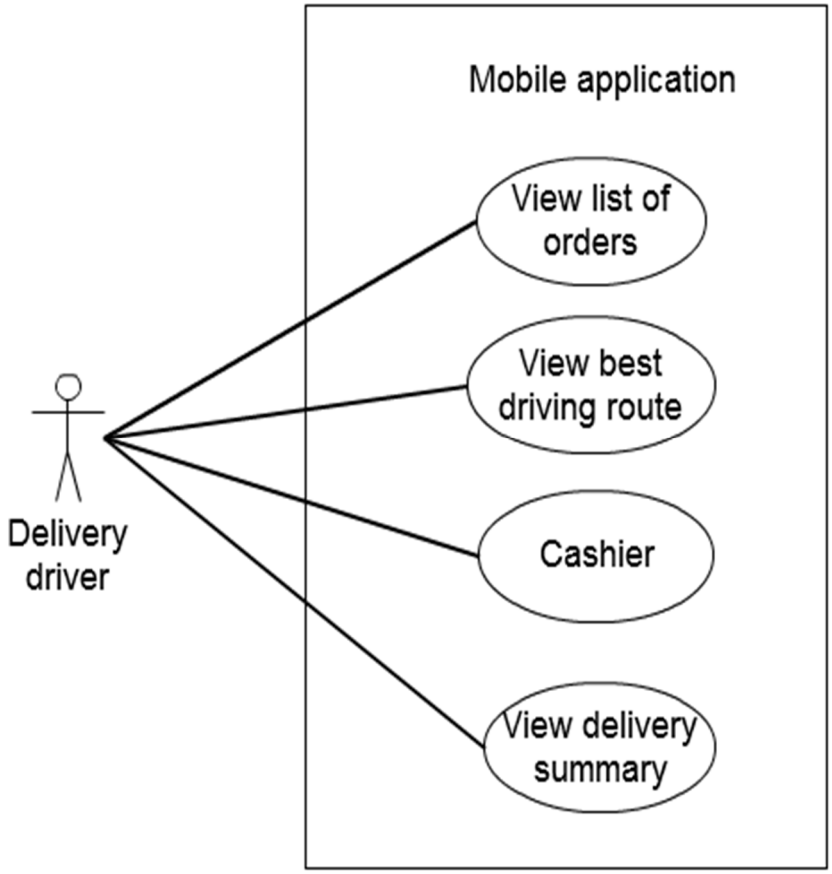



*Figure 20: Use case diagram “Mobile application”*

Description:

| | Use case | Description | Actor |
|---|---|---|---|
| 1 | View list of orders | Delivery driver can see the list of orders with customers' names and addresses. | Delivery driver |
| 2 | View best driving route | After choosing which orders are finished, the delivery driver can see the best driving route to deliver all orders on a Google map including driving guide. | Delivery driver |
| 3 | Cashier | Delivery driver can view the order's price and calculate payment. | Delivery driver |
| 4 | View delivery summary | Delivery driver can view a list of delivered orders. | Delivery driver |

*Figure 21: Description of the use case diagram "Mobile application"*

### 3.3.1.1.2 Web page for customers

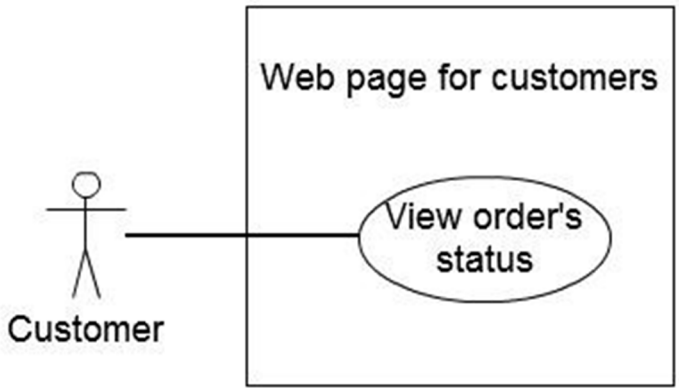



*Figure 22: Use case diagram "Web page for customers"*

Description:

| | Use case | Description | Actor |
|---|---|---|---|
| 1 | View order's status | The customer can enter Customer reference number and zip code to see the order's status. | Customer |

*Figure 23: Description of the use case diagram "Web page for customers"*

### 3.3.1.2 Flowcharts

#### 3.3.1.2.1 Mobile application

The flowchart below describes the actions the delivery driver takes when delivering food to customers.

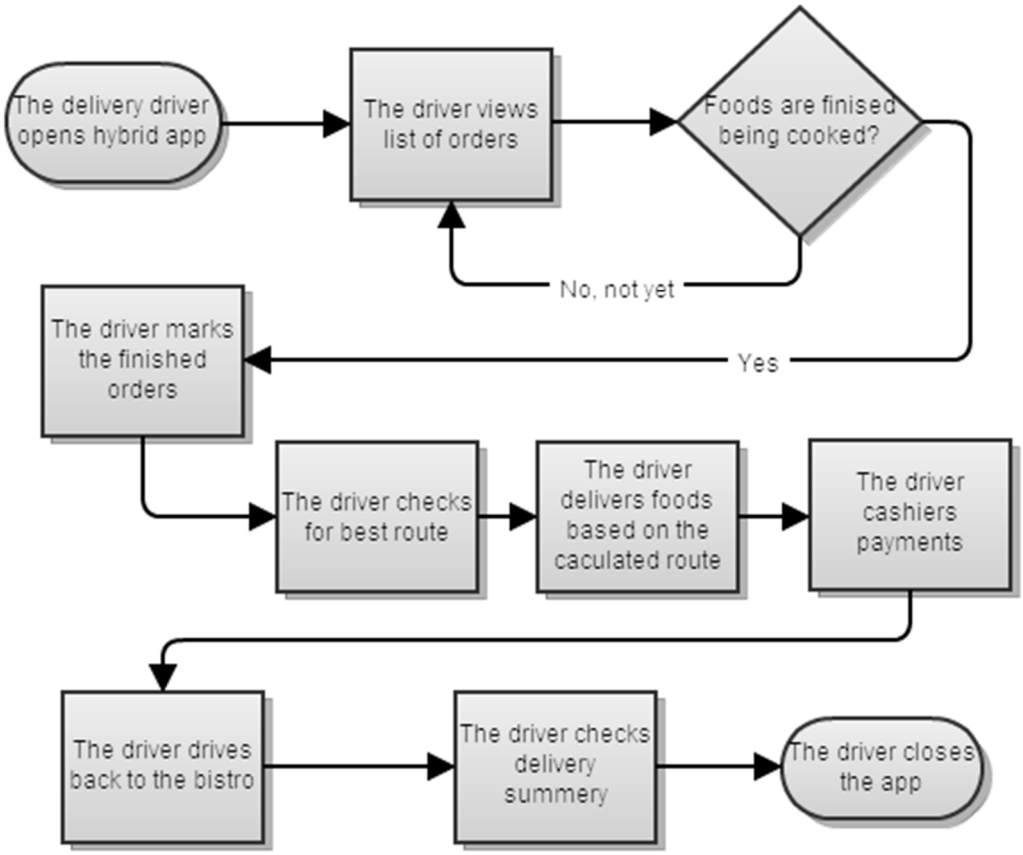



*Figure 24: Flowchart describes actions the delivery driver does with the application*

#### 3.3.1.2.2 Web page for customers

The flowchart below describes the actions a customer takes when checking the order's status.

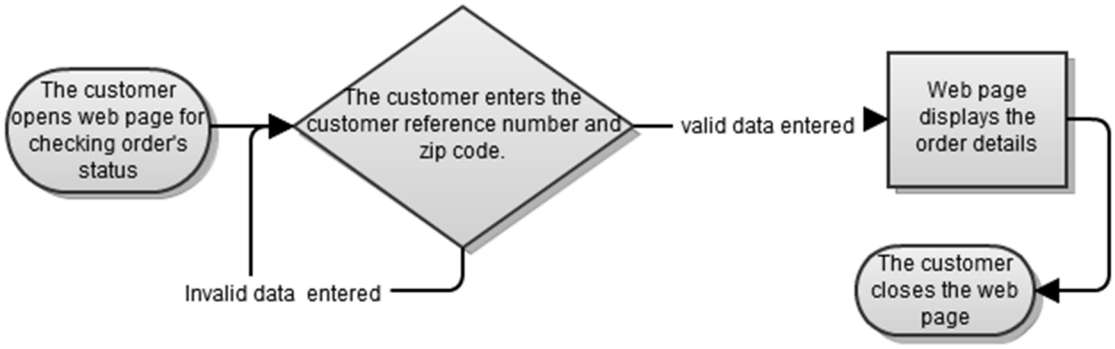



*Figure 25: Flowchart describes actions a customer takes on the web page for customers*

### 3.3.2 Database

Below is a design of the database schema:

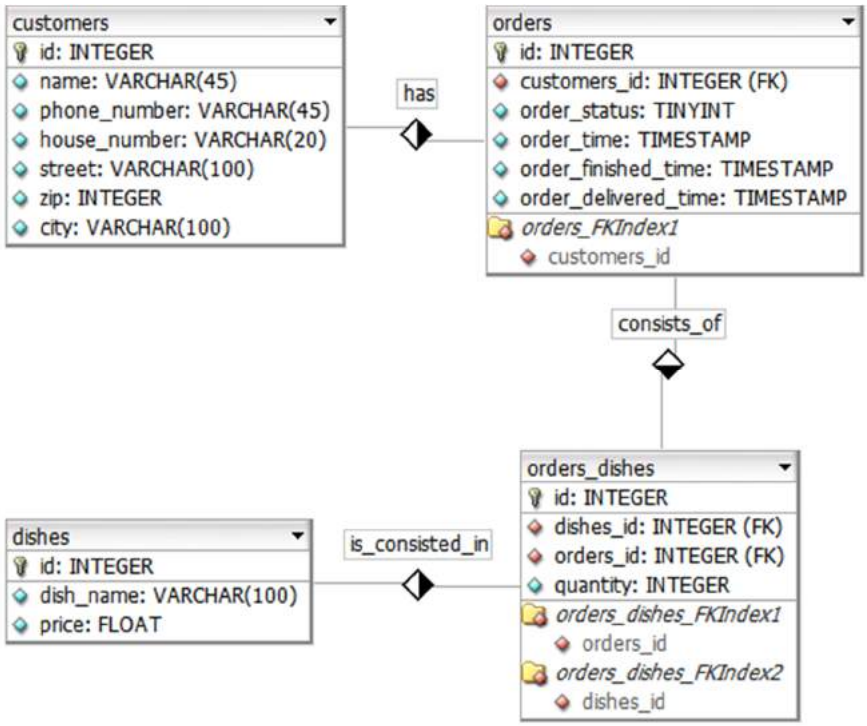



*Figure 26: Database schema*

The database schema is designed to fit the current database used at KimChan Bistro. The current database, however, contains only data of customers. The new schema includes new tables for orders and for the dishes served at the Bistro.

The table “orders” stores information about the status, price, payment and time relevant to orders.

The column *“order_status”* is type *tiny int*, it uses a small number to represent the status and will be updated according to in which state the order is. It has four stages:

- *0 : The order is just received and transferred to the kitchen*
- *1: The order is finished, and on the way to customer*
- *2: The order is delivered, payment has made*
- *9: The order is archived, it was in more than a day in database*

The table *“dishes”* stores information about the food served at the Bistro. This table has a *many-to-many* relationship with the table *“orders”,* therefore another table *“orders_dishes”* is needed to connect them together. This middle table acts as if it has many-to-one relationship to the two original tables.

This database serves data for requests from mobile app and for the requests from the web page for customers.

A connection to this database is made with PHP through the script below:

```
1 <?php
2   $con = mysql_connect("localhost:3306", "kimchan", "bTfm4n8rKWzRZA2G");
3   if (!$con)  {
4       die(mysql_error());
5   }
6   mysql_select_db("kimchan") or die(mysql_error());
7   mysql_query("SET NAMES utf8");
8 ?>
```

*Figure 27: PHP script to connect to the MySQL database*

Line 2 establishes the connection with database address at *localhost* on *port 3306* and connects to the database *“kimchan”* with the password following after.

After the connection was successful, the request to database is made by using SQL query in PHP script file to get the data needed.

```
require "lib/connectDB.php";

$sql = "SELECT o.id as order_id, o.order_price,
o.order_status, o.order_time,
c.id as customer_id, c.name as customer_name,
c.phone_number, c.house_number,
c.street, c.zip, c.city
FROM orders as o INNER JOIN customers as c ON o.customers_id=c.id
WHERE o.order_status <> '9'
ORDER BY o.order_time ASC";

    $result = mysql_query($sql,$con) or die(mysql_error());
$orders = array();

while($c = mysql_fetch_assoc($result)) {
    $orders[] = $c;
}

echo '{"orders":'.json_encode($orders).'}';
```

*Figure 28: A SQL query used in PHP script to get data from database*

The data received from the database could be displayed on a web page or will be transferred directly to the request clients. For the latter, it must be converted into a standard format, which clients can use. In this thesis, the JSON[22] format is used.

### 3.3.3 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++,*

[22] JSON: JavaScript Object Notation

*C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language." [53]*

Above is a definition of JSON. In PHP, data can be converted into JSON format by using the function *json_encode().*
JSON data can be parsed by different language because it is just a simple text data structure. In this implementation, the data will be received from web services running on the server. The requests are made from javascript files running on mobile device through AJAX[23] calls.

AJAX in web development is simply explained as a technology to send asynchronous request to server, so that the web page or a part of it can be updated without refreshing. [54]
Below is a part of the javascript on the mobile application, this javascript file sends an AJAX calling to the service *"get_orders.php"* on the server to get information of the order for use in calculating the best delivery driving route.

```
1 $.getJSON(serviceURL + 'get_orders.php', function(data) {
2   var customer_address = [];
3   var orders = [];
4   var waypts = [];
5   orders = data.orders;
6
7   $.each(orders, function(index, order) {
8       if (order.order_status == "1"){
9           customer_address = order.house_number+
10          ' '+order.street+', '+order.city+
11          ', '+order.zip+', Deutschland';
12          waypts.push({
13              location:customer_address,
14              stopover:true
15          });
16      }
17  });
18  ...
19 });
```

*Figure 29: A piece of javascript code to handle json data*

[23] AJAX: Asynchronous JavaScript and XML

The code above uses function “getJSON()” from jQuery Mobile to handle the JSON data.

### 3.3.4 Using jQuery Mobile to modify HTML pages

Beside the useful functions of jQuery, the javascript framework jQuery Mobile offers an easy way to build mobile app with html5.

The mobile app for KimChan is planned to be converted into hybrid app. Thus, a preparation is made by separating the static content with dynamic data from web server.
The HTML5 file contains mostly tags and attributes to mark the elements. jQuery Mobile script will inject the content into these elements.

```
<div id="list_of_orders_page" data-role="page"
     data-add-back-btn="true">

  <div data-role="header">
  <a href="index.html" data-icon="home"
     data-iconpos="notext" data-direction="reverse"></a>
  <h1>Bestellungsliste</h1>
  </div>

  <div data-role="content">
   <ul id="orders_list" data-role="listview"
        data-filter="true" data-theme="d">
   </ul>
   </div>

</div>
```

*Figure 30: The HTML file to display the list of orders with jQuery Mobile attributes*

Above is the static file with reserved element marked for the dynamic content. The java script bellows will do the injecting process:

```
...
li_append = '<li order_id="' + order.order_id +'">' +
      '<h3>'+order.customer_name+'</h3>' +
      '<p>' +
          '<strong>Kdnr. '+order.customer_id+'<br/>' +
                  order.street+' '+order.house_number+'<br/>' +
                  order.zip+' '+order.city + '<br/>' +
          '</strong>' +
          '<a href="tel:'+order.phone_number+'">'+
          order.phone_number+'<a/>'+
       '</p>' +
      '<p class="ui-li-aside">Best.-Nr. <strong>'+
     order.order_id+'</strong></p>' +
  '</li>';

$('#orders_list').append(li_append);
...
```

*Figure 31: The jQuery Mobile script used to inject orders' contents into the static html*

It is noted that multiple pages can be combine in one html file. jQuery Mobile will parse a *div* with the attribute *data-role="page"* as a separate page.

When an action is made on the mobile device, it triggers an event. This event is *binded* into the element, which receives the action.

Below is a snippet of code to bind the event *"swipe"* to the list element. The full code has its function to update orders' status when the delivery driver swipes on finished cooking orders on the mobile device.

```
$('#orders_list li').live('swipe', function(){
 ...
});
```

*Figure 32: A snippet of code to bind an event "swipe" to a list*

### 3.3.5 Using Google direction service

Finding the best-optimized path from one location to another one is a difficult issue. The delivery situation by KimChan Bistro is even harder to be solved because it involves many destinations. The delivery driver often takes 3-4 orders at a time to deliver.

Thanks to Google, this traveling salesman problem [55] is not for developers to solve themselves. Google has provided the Direction Service APIs in their Map API Web Services for calculating route, which includes many destinations. The destinations are called *waypoints* by Google.

To use this service, developers either call directly to the URL[24] offered by Google or include the api into the project and formulate a request with options and send with javascript to Google. This thesis chooses the latter.

```
<script src="http://maps.googleapis.com/maps/api/js?sensor=false"></script>
```

The script above will be included into html file. The attribute "*sensor*" lets Google indicate if a GPS or any sensor device is used to the determine user's location.

Below is the javascript to create request to the Direction service.

```
...
1  directionsDisplay.setMap(map);
2  directionsDisplay.setPanel(document.getElementById('summary_panel'));
3
4  var request = {
5   origin: init_address,
6   destination: init_address,
7   waypoints: waypts,
8   optimizeWaypoints: true,
9   travelMode: google.maps.TravelMode.DRIVING
10 };
11 directionsService.route(request, function(response, status) {
12  if (status == google.maps.DirectionsStatus.OK) {
13  directionsDisplay.setDirections(response);
14 }
...
```

*Figure 33: Piece of code to display map with calculated driving route*

It is a little effort to integrate the Direction Service from Google to the KimChan app. However, the result is enormous helpful.

### 3.3.6 Deployment with PhoneGap

The integration into a hybrid is made by using the online service of Phonegap at *http://build.phonegap.com*.

[24] URL: *Uniform Resource Locator*

Before submitting the code to Phonegap build, a small configuration should be made. Because the application runs on mobile device and send request to the server, the address of the server must be defined beforehand.

```
var serviceURL = "http://192.168.1.5/kimchan/services/";
```

It is in the variable *serviceURL* and locates in the file "*index.js".*

# 3.4 Test and conclusion

## 3.4.1 Environment

- *The services files are uploaded to the web hosting of KimChan Bistro. It is located at http://kim-chan.de/lieferung/services*
- *The website for customers is also uploaded to this web hosting. It is located at http://kim-chan.de/lieferung/web*
- *The application is compressed into a .zip file and uploaded to http://build.phonegap.com to let it be built into a hybrid app.*
- *The application can also be accessed on the internet as a web app at http://kim-chan.de/lieferung/app*
- *The database also is also located on the same web hosting. There are some sample data already inserted into this database.*
- *It is noted that the file index.js in the folder "\app\js\pages" of the application; the file "connectDB.php" in the folder "\services\lib" of the services; and the file "connectDB.php" in the folder "\web" were configured to for the web hosting above.*

## 3.4.2 Test run process and screenshots

After uploading the application to the online build service, Phonegap builds it and offers different completed files for different OS. Users can also download these setup file direct to the the devices by using QR-code[25].
However, the iOS developers must register a developer key in order to get the app built for iOS.

[25] QR-code: Quick Response code - http://www.qrcode.com/en/qrfeature.html

Below is the screenshot of the online build service from Phonegap:

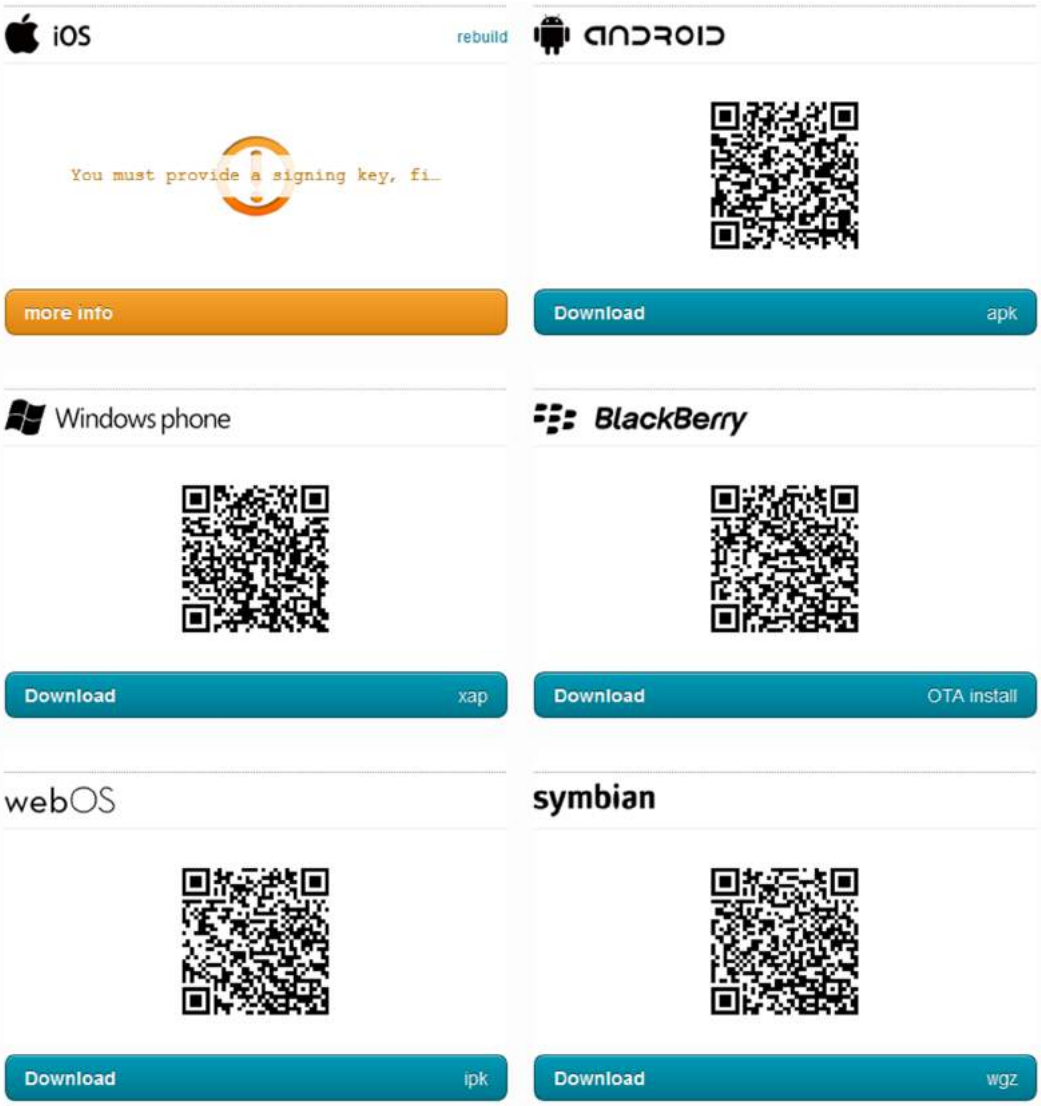
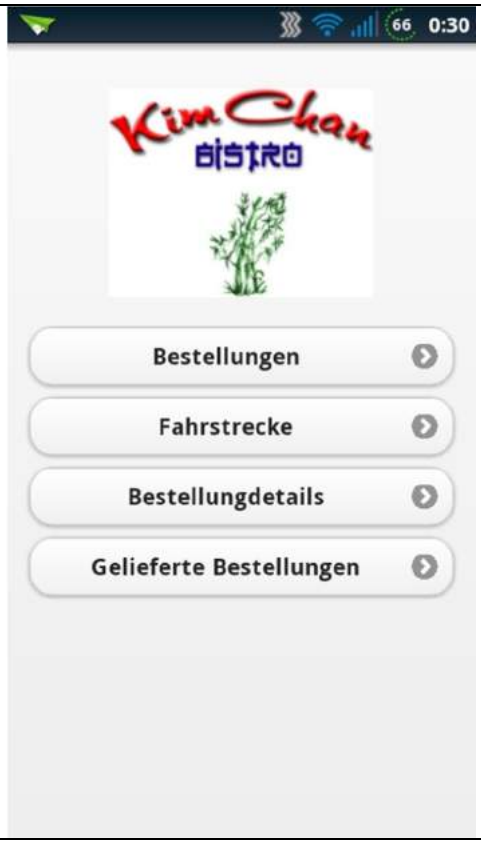



*Figure 34: The screenshot of online build service Phonegap*

The file .apk is downloaded to make a test on the mobile phone Motorola Defy+. Following are the screenshots of the test.




Screen 1:

This is the main screen of the application


Screen 2:

This screen lists all the orders, which are already finished (green) or are being cooked.

The delivery driver will swipe left or right on each order to toggle its status.


Screen 3:

After marking which order has been finished cooking at screen 2; when ready for delivery, the delivery driver can view a map with optimized route through all destinations.

The marker A and D are at the same address (Kimchan Bistro). Because they overlap each other, there is only marker D can be seen on the map.


Screen 4:

When arriving to the customer's address, delivery driver can view the details of the order.

The driver types the amount of money customer has paid, the app will automatically calculate the amount of change to give back.


Screen 5:

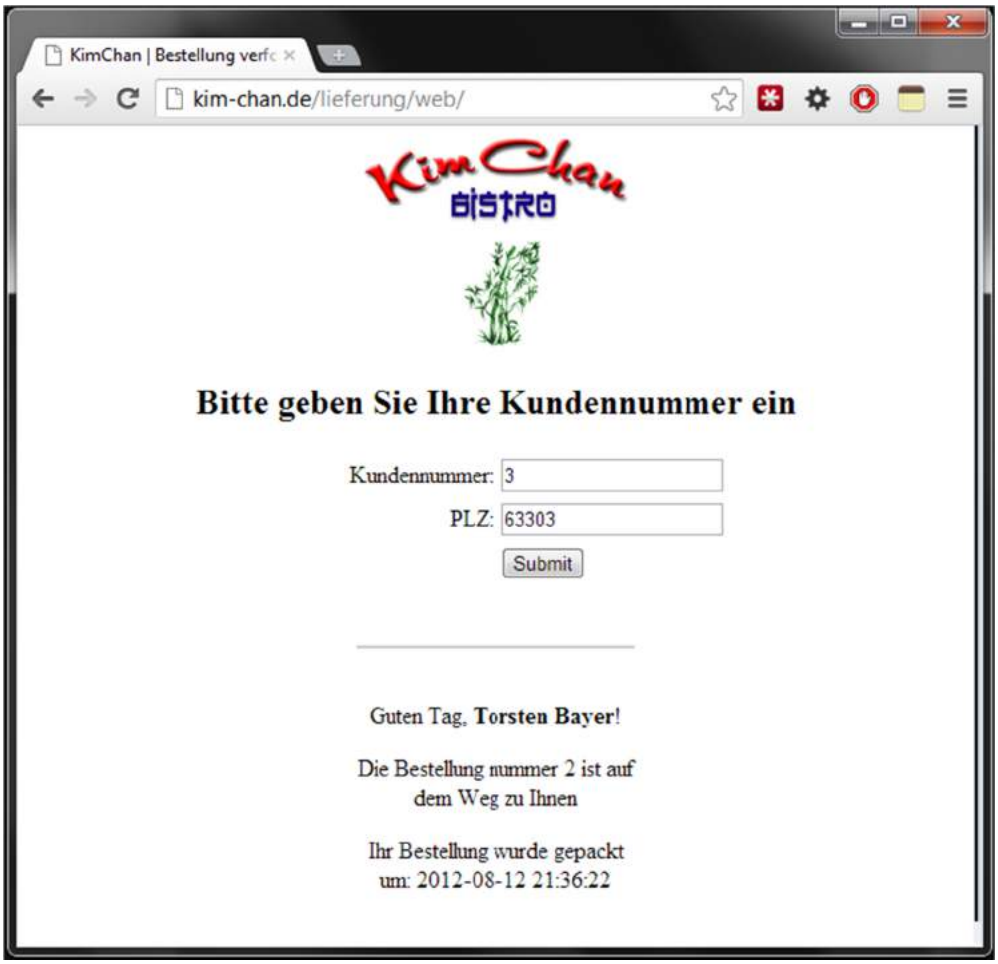
The driver taps the button “kassiert” at screen 4 to mark that the payment has been made.

A confirmation will pop up to ensure that the order is really paid.


Screen 6:

The delivery driver can have a look on all of the delivered orders.

To prevent mistakes, there is a button to mark that this order has actually not delivered. The order will again go to the list of orders at screen 2.




This is the web page for customers. Customers have to enter the reference numbers and the address' zip code to see their orders' status.

*Figure 35: Screenshots of the hybrid app KimChan and the web page for customers*

### 3.4.3 Conclusion

The hybrid app for KimChan Bistro has been successfully developed. It ran well on an android mobile phone. In addition, it would be built to six other platforms supported by Phonegap.

The development process was comfortable thanks to the framework jQuery Mobile. However, because this type of application based on web technologies, which works with stateless protocol, it was difficult to track when there are any error to be debugged.

The performance of the app in comparison with a normal native app is obviously slower by its speed. Nevertheless, the ability to be built for multiplatform is wonderful.

Although the app has met the main requirements of the suggested system, it could always be improved.
A nicer interface by styling the layout with CSS will make the app more enjoyable in its look.
The more features of HTML5 should also be applied into the app to make it works better.
A further development should be made is the integration of the app into a new system when KimChan Bistro decides to register with an online ordering service.

In conclusion, the idea of integrating modern software and online services into building a logistics solution was makeable and it has been successfully implemented.

# 4 Summary

The thesis, at first, describes the usual situation by restaurants. Then it names three popular online ordering services in Germany. By using one of these services, the restaurants can improve their ordering process and thus gains more customers from internet.

There was a graphic from IDC giving forecast about market shares of different operating system in 2016. According to this forecast, Android will dominate the field. Some mobile systems have also been introduced.

The thesis continues by explaining three different types of mobile applications. Each of them has advantages and disadvantages. The powerful of native apps is in its speed and performance. The beauty of web apps is in its ability to run on different platforms. Lastly, hybrid apps have the flexibility of working like the web apps but are packaged to have a native look as of native apps.

The main chapter of the thesis focuses on developing a hybrid app for the delivery drivers. The thesis has chosen to build the app with web technologies and to package it with Phonegap.

The thesis has found a way to calculate the optimized delivery driving route through many destinations. This is a difficult problem even for mathematicians. However, by utilizing Google Map API Web Services, and especially the Google Direction Service, the problem is solved. Google has made the services API easy enough to use. The thesis has a chance to integrate an external powerful computing resource into its project.

During the implementation of the hybrid app, some useful technologies have been used:

Apache HTTP server, MySQL, PHP have been gathered as a software stack to work on the server side to serve request from clients.

HTML5 is a markup language and it is used for building the underlying structure of the application.

jQuery Mobile is used to request data and to put these data into the HTML5 document. It is also a helpful javascript framework to deal with creating mobile interfaces and reacting on events occurred from the users.

JSON has helped as an exchange data format between the app and the server.

After having being developed, the app is packaged into a hybrid app with the online build service from Phonegap. This hybrid app runs well with the test environment and is expected to run the same on other platforms, which were supported by Phonegap.

The implementation finishes with a successful result. The thesis also notices some points, which could be still improved.

In a whole, the thesis has introduced a way to use modern software and online services for improving the performance in a field of logistics.

This is an innovative solution and should be considered to apply into any other business.

# References


[1] IDC, *Marktanteile Smartphone-Betriebssysteme 2012 und 2016 | Prognose.* Available: http://de.statista.com/statistik/daten/studie/182363/umfrage/prognostizierte-marktanteile-bei-smartphone-betriebssystemen/ (2012, Jul. 27).

[2] Apple Inc, *iOS Dev Center - Apple Developer.* Available: https://developer.apple.com/devcenter/ios/index.action (2012, Jul. 31).

[3] Microsoft, *Windows Phone Development.* Available: http://msdn.microsoft.com/en-us/library/ff402535(v=vs.92) (2012, Jul. 31).

[4] Open Handset Alliance, *Android Developers.* Available: http://developer.android.com/index.html (2012, Jul. 31).

[5] R. Meier, *Professional android 2 application development*. Indianapolis, IN: Wiley, 2010.

[6] Apple Inc, *iOS Technology Overview: About iOS Development.* Available: http://developer.apple.com/library/ios/#documentation/Miscellaneous/Conceptu-al/iPhoneOSTechOverview/IPhoneOSOverview/IPhoneOSOverview.html#//apple_ref/doc/uid/TP40007898-CH4-SW1 (2012, Aug. 01).

[7] Microsoft, *Application Platform Overview for Windows Phone.* Available: http://msdn.microsoft.com/en-us/library/ff402531(v=vs.92) (2012, Jul. 28).

[8] IBM, *HTML5, Hybrid or Native Mobile App Development.* Available: http://www.worklight.com/assets/files/HTML5,%20Hybrid%20or%20Native%20Mobile%20App%20Development.pdf (2012, Aug. 05).

[9] W3C, *SQL Tutorial.* Available: http://www.w3schools.com/sql/default.asp (2012, Aug. 08).

[10] The jQuery Foundation, *jQuery Mobile Docs - Quick start.* Available: http://jquerymobile.com/demos/1.1.1/docs/about/getting-started.html (2012, Aug. 07).

[11] Jordan Crook, *Apple's App Store Hits 30 Billion Downloaded Apps, Paid Out $5 Billion To Developers | TechCrunch.* Available: http://techcrunch.com/2012/06/11/apples-app-store-hits-30-billion-downloaded-apps-paid-out-5-billion-to-developers/ (2012, Aug. 07).

[12] Xyologic, *Xyologic - App Stats.* Available: http://www.xyologic.com/app-downloads-reports (2012, Aug. 07).

[13] Joel Kaczmarek, *Lieferdienste in Deutschland: Gründerszenes Wrap-up der drei deutschen Konkurrenten Pizza.de, Lieferheld und Lieferando.* Available: http://www.gruenderszene.de/allgemein/lieferdienste# (2012, Jul. 26).

[14] Louis Pfitzner | Lieferheld GmbH, *Pressemitteilung LIEFERHELD Express Box* (2012, Jul. 26).

[15] Karsten Pfrommer, *Blitzschnell geordert – prompt geliefert.* Available: http://www.mm-logistik.vogel.de/distributionslogistik/articles/364906/ (2012, Jul. 28).

[16] ____, *M2M-Boxen kommunizieren per einfaches Gateway.* Available: http://www.mm-logistik.vogel.de/distributionslogistik/articles/364906/index2.html (2012, Jul. 26).

[17] Marc Holder Kluver, *Delivery Droid Free*: Google Play, 2011.

[18] Gartner Inc, *Gartner Says Worldwide Sales of Mobile Phones Declined 2 Percent in First Quarter of 2012; Previous Year-over-Year Decline Occurred in Second Quarter of 2009.* Available: http://www.gartner.com/it/page.jsp?id=2017015 (2012, Jul. 27).

[19] Mathew Honan, *Apple unveils iPhone | Macworld.* Available: http://www.macworld.com/article/1054769/iphone.html (2012, Jul. 29).

[20] A. Inc, *Apple - Press Info - iPhone SDK Downloads Top 100,000.* Available: http://www.apple.com/pr/library/2008/03/12iPhone-SDK-Downloads-Top-100-000.html (2012, Jul. 30).

[21] Nilay Patel, *iPhone OS 4 renamed iOS 4, launching June 21 with 1500 new features -- Engadget.* Available: http://www.engadget.com/2010/06/07/iphone-os-4-renamed-ios-gets-1500-new-features/ (2012, Jul. 30).

[22] Apple Inc, *Start Developing iOS Apps Today: Set Up.* Available: http://developer.apple.com/library/ios/#referencelibrary/GettingStarted/RoadMapiOS/chapters/GetToolsandInstall.html (2012, Jul. 30).

[23] android.com, *Philosophy and Goals | Android Open Source.* Available: http://source.android.com/about/philosophy.html (2012, Jul. 30).

[24] Group Pledges to Unleash Innovation for Mobile Users Worldwide, *Industry Leaders Announce Open Platform for Mobile Devices.* Available: http://www.openhandsetalliance.com/press_110507.html (2012, Jul. 30).

[25] Open Handset Alliance, *Android Overview | Open Handset Alliance.* Available: http://www.openhandsetalliance.com/android_overview.html (2012, Jul. 30).

[26] Android Developers, *Building Your First App.* Available: http://developer.android.com/training/basics/firstapp/index.html (2012, Jul. 31).

[27] Samuel Moreau, *Designing Metro style: principles and personality | BUILD2011 | Channel 9.* Available: http://channel9.msdn.com/Events/BUILD/BUILD2011/APP-395T (2012, Jul. 31).

[28] Microsoft, *.NET Framework Developer Center - WPF, WCF, WF, ADO.NET, CardSpace, LINQ | MSDN Online.* Available: http://msdn.microsoft.com/de-de/netframework/ (2012, Jul. 31).

[29] ____, *Hardware Specifications for Windows Phone.* Available: http://msdn.microsoft.com/en-us/library/ff637514(v=vs.92).aspx (2012, Jul. 31).

[30] Sybase Inc, *The Cost of Building and Maintaining Native Mobile Apps Article: - Sybase Inc.* Available: http://www.sybase.com/detail?id=1095384 (2012, Aug. 01).

[31] Mobile HTML5.org, *Mobile HTML5 - compatibility tables for iPhone, Android, BlackBerry, Symbian, iPad and other mobile devices.* Available: http://mobilehtml5.org/ (2012, Aug. 06).

[32] Real Tech IT blog, *Web App vs. Native App?* Available: http://blog.realtech.com/web-app-vs-native-app/ (2012, Aug. 06).

[33] Global Intelligence, *Native or Web Application?* Available: http://www.globalintelligence.com/insights-analysis/white-papers/native-or-web-application-how-best-to-deliver-cont/GIA%20Industry%20White%20Paper%202_2010_Native%20or%20Web%20App_How%20Best%20to%20Deliver%20Content%20and%20Services%20to%20Your%20Audiences%20over%20the%20Mobile%20Phone_Apr-10.pdf (2012, Aug. 06).

[34] JT Mudge, *Native App vs. Mobile Web App: A Quick Comparison.* Available: http://sixrevisions.com/mobile/native-app-vs-mobile-web-app-comparison/ (2012, Aug. 06).

[35] Adobe, *PhoneGap | Home.* Available: http://phonegap.com/ (2012, Aug. 06).

[36] Appcelerator, *Titanium Mobile Development Platform.* Available: http://www.appcelerator.com/ (2012, Aug. 06).

[37] Google Inc, *Google Maps API — Google Developers.* Available: https://developers.google.com/maps/ (2012, Aug. 04).

[38] Apache Software Foundation, *Welcome! - The Apache HTTP Server Project.* Available: http://httpd.apache.org/ (2012, Aug. 05).

[39] M. I. N. Team, *Overview : The Official Microsoft IIS Site.* Available: http://www.iis.net/overview (2012, Aug. 05).

[40] Netcraft, *July 2011 Web Server Survey | Netcraft.* Available: http://news.netcraft.com/archives/2011/07/08/july-2011-web-server-survey.html (2012, Aug. 05).

[41] MySQL, *MySQL :: Why MySQL?* Available: http://www.mysql.com/why-mysql/ (2012, Aug. 08).

[42] Web Hypertext Application Technology Working Group, *Web Hypertext Application Technology Working Group.* Available: http://www.whatwg.org/ (2012, Aug. 05).

[43] W3C, *Introduction — HTML5.* Available: http://dev.w3.org/html5/spec/introduction.html#introduction (2012, Aug. 05).

[44] WHATWG, *WHATWG Wiki.* Available: http://wiki.whatwg.org/wiki/FAQ#What_is_the_WHATWG.3F (2012, Aug. 05).

[45] Mark Pilgrim, *Dive Into HTML5.* Available: http://diveintohtml5.info/ (2012, Aug. 07).

[46] The jQuery Foundation, *jQuery Mobile | jQuery Mobile.* Available: http://jquerymobile.com/ (2012, Aug. 06).

[47] Adobe, *Adobe Announces Agreement to Acquire Nitobi, Creator of PhoneGap.* Available: http://www.adobe.com/aboutadobe/pressroom/pressreleases/201110/AdobeAcquiresNitobi.html (2012, Aug. 07).

[48] ____, *Apache Cordova API Documentation.* Available: http://docs.phonegap.com/en/edge/guide_getting-started_android_index.md.html#Getting%20Started%20with%20Android (2012, Aug. 07).

[49] ____, *PhoneGap Build.* Available: https://build.phonegap.com/ (2012, Aug. 07).

[50] Google Inc, *Google Maps API Web Services - Google Maps API Web Services — Google Developers.* Available: https://developers.google.com/maps/documentation/webservices/ (2012, Aug. 12).

[51] ____, *Google Maps API licensing - Google Maps API — Google Developers.* Available: https://developers.google.com/maps/licensing (2012, Aug. 08).

[52] Apache Friends, *apache friends - xampp.* Available: http://www.apachefriends.org/en/xampp.html (2012, Aug. 11).

[53] json.org, *JSON.* Available: http://www.json.org/ (2012, Aug. 11).

[54] Mozilla Developer Network, *Getting Started | Mozilla Developer Network.* Available: https://developer.mozilla.org/en-US/docs/AJAX/Getting_Started (2012, Aug. 12).

[55] N. S. F. S. o. I. a. S. E. a. G. T. Office of Naval Research, *The Traveling Salesman Problem.* Available: http://www.tsp.gatech.edu/problem/index.html (2012, Aug. 12).