

IT 01 / 112
18/04/2016

Mobile Device Driven Taxi Management System

LIBRARY
UNIVERSITY OF MORATUWA, SRI LANKA
MORATUWA

Duminda D. F. Piumwardena
108570J

Dissertation submitted to the Faculty of Information Technology,
University of Moratuwa, Sri Lanka for the partial fulfillment of the
requirements of the Degree of Master of Science in
Information Technology

004 " 15"
004 (04-3)



109933
+
DVD (109932-45)
TH3031

109933

Declaration

We declare that this thesis is our own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been acknowledged in the text and a list of references is given.

Duminda D. F. Piumwardena

June 12, 2015

Supervised by:

UOM Verified Signature

Dr.Thilak Chaminda

June 12, 2015

Acknowledgements

I am using this opportunity to express my gratitude to everyone who supported me throughout the MSc. Project. I am thankful for their guidance and constructive criticisms and friendly advices during the project work.

I express my warm thanks to Dr. ThilakChaminda at Informatics Institute of Technology for his support and guidance.

I would also thank to the project coordinator Mr. SamindaPremarathne and office assistant Ms. Niranjala.

Thank you,

Duminda Piumwardena

Abstract

This dissertation is about the project 'Mobile Device Driven Taxi Management System'. The project consist tree main software implementations as Server Sub System (SSS), Operator Desktop – Client application (ODC) and Vehicle Driver's Mobile application (VDM). SSS is the core and service provider for both ODC and VDM applications. The main objective of the above Sub systems architecture is to use the currently available web and mobile technologies in development of the solution and high availability of the system in operation at the business point of view.

In the first phase of the Software Development Life Cycle the knowledge of taxi management domain has been acquired. They have been analyzed and best architectural design was decided based on the requirement. Code level implementation and suitable tests were conducted prior release of the software to the Business Client. Mainly the Iterative Model of Development has been used for Software development in the project to minimize the time and requirement gathering ambiguities. Then in each phase there have been modules were developed in the development of code.

The software solution has been delivered to the business client and the solution has been deployed on their computer systems.

Table of Contents

DECLARATION	I
ACKNOWLEDGEMENTS	II
ABSTRACT	III
CHAPTER 1. INTRODUCTION.....	1
1.1 Background and motivation.....	1
1.2 Problem in brief	1
1.3 Aims and objectives.....	3
CHAPTER 2. RELATED WORKS.....	4
CHAPTER 3. SYSTEM DEVELOPMENT APPROACH	7
3.1 Sub-systems of the Solution.....	7
3.1.1 Server Sub System (SSS).....	7
3.1.2 Operator Desktop – Client application (ODC)	8
3.1.3 Vehicle Driver’s Mobile application (VDM)	8
3.2 System Architecture.....	9
3.3 Application Layered Architecture.....	9
CHAPTER 4. TECHNOLOGY USED.....	11
4.1 Introduction.....	11
4.2 .Net & C# Language	11
4.3 Asp.Net and WebAPI.....	12
4.4 AngularJS.....	12
4.5 Windows Mobile SDK.....	13
4.6 Other technologies	13
CHAPTER 5. ANALYSIS AND DESIGN	15
5.1 Use-cases.....	15
CHAPTER 6. IMPLEMENTATION	19
6.1 Domain Model	19
6.2 Multi-criteria Decision Analysis.....	21
CHAPTER 7. EVALUATION.....	23
7.1 Evaluation Methods	23
7.2 Results and Outcomes.....	23
7.3 Discussion.....	24
CHAPTER 8. DISCUSSION	25
8.1 Comparison to the other solutions	25
8.2 Further developments.....	25
CHAPTER 9. REFERENCES.....	26

List of Figures/ Tables

Figure 3.2.1: High Level Architecture of the System	08
Figure 3.3.1: Layer Diagram	09
Figure 5.1.1: Use-case Diagram which interacts with 'Operator' actor	14
Figure 5.1.2: Use-case Diagram which interacts with 'Driver' actor	15
Figure 6.1.1: Domain-Class Diagram of the system	19
Figure 6.2.2: Flow chart of selecting best vehicle-driver	21
Table 2.1: Comparison of other solutions	4
Table 6.2.1: Example vehicle-driver list	20

Chapter 1. Introduction

1.1 Background and motivation

The business I selected to be developed from Ja-Ela area called Vasi Cab Services. They operate cab services to the public. Mainly they provide vans and cars and they have started a new service as budget cabs by expanding the business for commuters who are seeking low budget vehicles.

Vasi Cab Service provides vehicles for hiring short distance travels and long distance tours. When a new hire is requested by a client the operator contacts the vehicles near to the client's area through telephone calls and gets the status of the vehicle whether they are ready to take the hire. If it is success the operator inform the hire details to the driver and inform the client about the driver over the phone. When a trip is completed the driver again make phone call to the operator to inform the current location he parked and the driver should record the hire details mainly charges, distance and time taken of the completed trip.

The company owner's main concern is to maximize hires by utilizing communication between vehicle drivers and the operating office and to reduce the paper based data maintenance at the drivers end. These limitations have lead the drivers to get less hire trips and some mal practices when entering data.

1.2 Problem in brief

There are some major issues were found during the initial and subsequent requirement gathering phases of the project. They are being briefly explained in below.

- **Communication**

Communication is a key factor in any business. It is not less important in the Transportation and taxi business. There are main functional points in taxi management where communication is crucial. One is when the operator need to know the driver and vehicle's current status. Whether they are on a hire, or at a parked location, or the vehicle is broken down. Or maybe what their next destination is if they are on a hire. These information is kept by the operator when a new booking is placed and search for the candidate vehicle and driver.

There is no better way currently than get a phone call to the driver and ask about the above information and write down on a piece of paper at the operator end.

Another situation the communication important is when a new booking is placed by a customer and the operator finding out the best possible candidate driver-vehicle to be sent to the customer. When that situation also they currently doing is operator make calls to a selected list of drivers that operator deciding those drivers are near to the customer's location and ask about their availability to take the hire. For the above described scenarios most of the time drivers do not answer phones during a trip as it is a punishable offence if they get caught to a police officer. So the number of hires get missed due to the lack of communication.

- Administering

Keeping track of vehicles, who are the drivers and what are their phone numbers or contact details. Customers and their details and how quickly the operator know the recent bookings the customer made.

Most importantly maintaining the bookings and hire details. As the company get paid by them and the revenue to the business. These are lack in the current system adopted, having a major impact on the business growth.

- Operational

As a new booking is arrived, in the next minute the customer is waiting for a call from a driver who is going to pick him up, that is the reality. However when the customer place a booking, there is a big struggle finding a suitable driver for him at the operator's end. Since the information is limited at the operator's end to make decision. If otherwise the operator need to make calls to the driver as described in above communication drawbacks finding out missing details. When all things happen the customer may go for another taxi service.

1.3 Aims and objectives

Aims and Objectives of the 'Mobile Device Driven Taxi Management System' are as follows.

Aim

- To develop a solution to 'Vasi Cab' in order to overcome their current difficulties and inefficiencies, by introducing latest technology and expertise.

Objectives

The above aim will be accomplished by fulfilling the following research objectives.

- Collect the information related to the business by interviewing management, center operators and drivers.
- Review literature for similar problems in the business domain to get the idea about the solutions they have already given to them.
- Analyze and design the software by architecting for extensible for the future changes
- Develop the software solution using software development languages and technologies.
- Test the solution with test cases and do the bug fixing if any.
- Deliver and install the solution to the client.
- Do user training sessions as required. Prepare simple user guide for different functions of the system.

Chapter 2. Related Works

There are some similar solutions to the Taxi Management system could be found in the World Wide Web and they are commercial solutions from different software development companies all over the world. However there's no similar or close solution could be found in Sri Lanka.

I will discuss four solutions similar to my solution. The comparison of the solutions are tabulated as follows.

DDS Digital Dispatch <i>Canada</i>	CMS(Cab Management System) <i>Middlesex, UK</i>	iCabbi <i>Dublin, Ireland</i>	SmartMove <i>South Australia</i>
Features			
<ul style="list-style-type: none"> • Powerful dispatching • Flexible booking - Online • Comprehensive reporting • Easy payment • Cutting-Edge Hardware (Mobile Device) <p>Easily collect arrival and performance data</p>	<ul style="list-style-type: none"> • Vehicle tracking • Mobile based Cab driver application • Directions to cab driver • Online booking (mobile responsive) 	<ul style="list-style-type: none"> • Automated Dispatch System • Feature Rich Interface • Multi-Location Access • Live Taxi Map • Detailed Fare History • Consumer Security • Caller-ID • Booking Engine • Statistics & Trends • Credit Card Payments • Signature Capture feature 	<ul style="list-style-type: none"> • Automated taxi dispatch • Tracking system • Job allocation based on vehicle location • In car touch unit • Customer Tax accounting • No operator intervention after the booking • Supervisor override possibility
Technologies			
<ul style="list-style-type: none"> • Cloud based, • Web based dispatch, reporting, booking • Geo mapping • Multimode cellular data network 	<ul style="list-style-type: none"> • Need to manage servers • Geo mapping 	<ul style="list-style-type: none"> • Cloud-Based • Andriod& iPhone for drivers • Geo mapping 	<ul style="list-style-type: none"> • Managed servers, • iOS & Android Apps for booking • Geo mapping
Hardware & resources on Vehicle end			
<ul style="list-style-type: none"> • Custom device and embedded software. 	<ul style="list-style-type: none"> • Smart mobile phones 	<ul style="list-style-type: none"> • Smart mobile phones 	<ul style="list-style-type: none"> • Custom device and embedded software.

Table 2.1: Comparison of other solutions

In contrast to the above four solutions and the Taxi Management solution they are much similar in operation and the commercial software have been designed to maximize sales by modularizing features however having same set of features behind them.

In the limited time frame and the limited resource we have in the University project I had to think of most important features to be implemented in the project and keep provision to the enhancements.

Chapter 3. System Development Approach

In this chapter I am going to discuss about the operations of the system in technical level of description. This involves what are the system components, who are the users of the system, how they interact with the system, how the system responds for the users' requests and how the technology combines these aspects of the system.

Also in this chapter the system architecture and the layered architecture of the application server implementation are described.

3.1 Sub-systems of the Solution

There are mainly three sub systems in the solution. They are namely 'Server Sub System', 'Operator Desktop – Client Application' and the 'Vehicle Driver's Mobile application'. Detailed description of each sub system are as follows.

3.1.1 Server Sub System (SSS)

The server software is the core and the heart of the whole system. This software resides in a web server and when a request is received to the SSS it will respond with appropriate data by going through logics behind and it may involve data base as well. Requests basically come from two sources one from Operator Desktop (ODC) and from Vehicle Driver's Mobile (VDM) sub systems. That is ODCs and VDMs are the main consumers of the SSS services. Functions of the two will be discussed in next two sections of this chapter.

3.1.1.1 Web API

SSS has the top most Web API layer in Figure 6.1.1 which is the interface and in other words the gateway to the said two sub systems. This Web API is developed using ASP.Net MVC Web API technologies. Web API is capable of catering RESTful services. Responses are always JSON (Javascript Object Notation).

3.1.1.2 Business Logic

As in Figure 6.1.1 Business logic resides in the second layer of the SSS application. Main function is to execute business related rules and logics for each request comes to Web API. This component is written in C# .Net[4].

3.1.1.3 Data Access

Data Access is the third layer of the SSS system and all the database related functions goes here and be called by the top layers WebAPI through Business Logic.

This component has direct communication with the Database of the system. Data access layer has the OR Mapping tools to execute SQL statements and convert the results to objects and object collections. As described in 3.6.1 sub section OR mapping is done using Entity Framework.

3.1.2 Operator Desktop – Client application (ODC)

Operator Desktop is one of the Actors interface to the system. Actor or the User is Operator who are in the Central dispatch unit of the institution. There might be one or more such users in a Taxi service providing institution. Operator user's responsibilities and use-cases will be discussed on next chapter.

These Desktop Clients are Web Browser based Web Applications written using HTML, Javascripts and CSS. Behind the scene this is a Single Page Application (SPA), first time it loads only one page and when user navigates to another screen it will render few partial views in the same 'single page'. The SPA behaviors and UI logics will be handled by AngularJS Javascript library as it is described in section 3.4. One another technology I used here is SignalR that is to listen to server push responses. The functionality of SignalR here is when some Driver update any information or respond to a system generated message SignalR will capture those status changes and update the ODC screens as needed. All the Operator users' functions will be secured by user authentication mechanism. Then unauthorized access to the system will be restricted as this is not an application for general public.

3.1.3 Vehicle Driver's Mobile application (VDM)

Drivers end Mobile application is the next service consumer of the SSS. That is the user interface and the software for the vehicle drivers.

This sub system is totally done using Windows Mobile SDK and can be installed to Windows Mobile devices as an App on the device. Like in ODC all the functions that can be performed by the Driver users will be secured by user account authentication.

VDM will talk with the server using web requests and responses through available mobile data service. Since that VDM and SSS communications will not be function properly in the out of coverage areas of the mobile data provider. However fallback solutions such as offline data storage and message queuing are introduced for such situations. And the Driver user and the Operator desktop users will be notified in such situations.

3.2 System Architecture

The system architecture will be as in the following Figure 3.2.1.

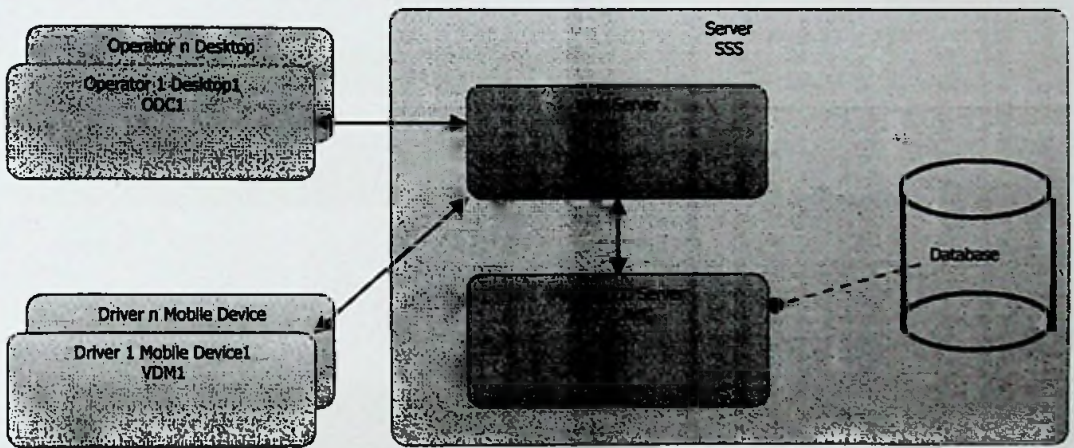


Figure 3.2.1: High Level Architecture of the System

The main three sub systems Operator Desktop (ODC), Vehicle Driver Mobile (VDM) and the Server Sub System (SSS) are depicted here and the interactions are marked in arrows for clarity. Database is placed in the Server where SSS resides but that can be a separate server hardware location in connected environment and has no extra effort or performance deviation related to the alteration.

3.3 Application Layered Architecture

The system will going to adhere to the following Layered Architecture in the development phase. Layered Diagram is as follows in Figure 3.3.1.

The advantage of having layered development architecture is easy to maintain code, can do parallel development if we go for industry level human resource allocation.

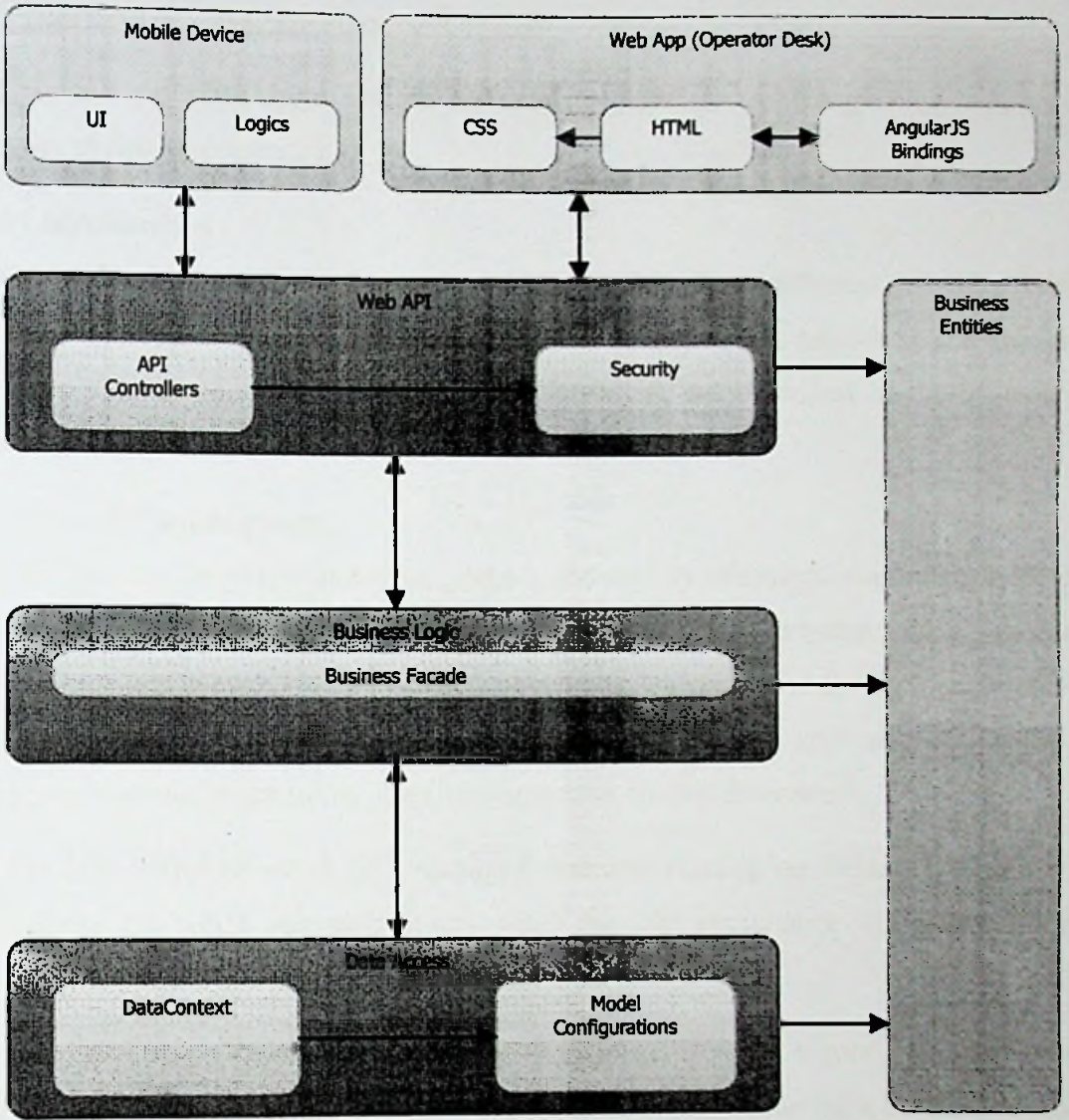


Figure 3.3.1: Layer Diagram

Chapter 4. Technology Used

Chapter 4 describes about how the software development technologies are been used in the system development.

4.1 Introduction

The technologies use in the server software and client software are latest web technologies introduced by highly trusted institutions such as Microsoft and Google incorporations. More details about the technologies are described in the following sections.

4.2 .Net & C# Language

C# is a computer programming language introduced by Microsoft incorporation [4] in early 2000. This language is an object oriented sophisticated language and it is said to be robust and secure. C# can be used developing various software such as Window clients, distributed systems, web services, Client server applications, database applications and much more. This language runs on .Net framework.

.Net (dot net) framework is a managed platform running on Windows Operating Systems. On which run applications which are .Net compliance. C# is one of the languages which are running on .Net platform.

In terms of the software solution, these two C# and .Net are a good combination to kick start a project like this. Since they are easy to learn and having a good support on memory management and resource allocation are some of the crucial and time consuming areas to develop if we started from the scratch. And .Net gives immense support on security to the application developer makes. That is another advantage of using these. Since C# is a popular object oriented language, developers can only think in terms of object oriented aspects and various tools like UML can be involved in designing the software, and can be used throughout the software development life cycle. As these are Microsoft based development tools they give a very comprehensive documentation for almost everything on language and framework. And the community support on these two technologies are very high in the internet, when a developer get stuck in a situation there are lot of community sites come to help them out.

4.3 Asp.Net and WebAPI

ASASP.Net [10] is also come in handy with .Net framework to rescue developers when they struggle to develop web related applications. There are lot of technologies come together with ASP building web applications such as HTML, CSS, Javascripts, Server side scripting, C#, Web services, Web API etc.

Latest trend in the web is having Web APIs (Advance Programing Interfaces) rather than exposing Web services for middleware business communication. Web Services are usually heavy in processing and protocol wise. Web APIs are easy to maintain, high in responsive and can get the benefits of RESTful [6] architecture. In this project Web API is used as the top most layer as in Figure 6.1.1 in the server side application, which can be communicated to both Operator Desk client application and Driver Mobile device application.

4.4 AngularJS

AngularJS [1] is a Javascript library for web applications maintained by Google Inc. This is one of the most popular library in web development world to build SPAs (Single Page Applications). SPA is another newly introduced set of techniques in developing enterprise level web applications. Using SPA application user feels smooth functioning even though it is a web application. In another words user does not feel the web page is refreshing after each submit function, so get the experience much more similar to well-designed desktop application.

Some of the well-known features of AngularJS are

- Ajax communication (Support REST)
- Client side state management
- URL routing
- Client side two-way data binding
- Templates
- MVC
- Dependency Injection

There are some advantages of AngularJS beyond the project scope like Javascript testability, parallel development (Application's UI designing using HTML, CSS can be parallel develop to client side script developments)

In this system development all the client side screens UI logics, data binding, url routing and server communication are backed by Angular as this is rich in those techniques. Aside to AngularJS, JQuery [3] is used in the background handing hand with Angular to give a rich look and experience to the user.

4.5 Windows Mobile SDK

Software development Kit for Windows Mobile [7, 14] is been used to develop the Vehicle Driver's Mobile device application. There is a set of technologies and protocols being used to build Windows Mobile native applications, such as XAML for UI designing C# for overall coding.

This SDK comes with class libraries for development and mobile device emulator to easily run and test application. Why do we care about Windows Mobile here is first thing I have a Windows phone with me so that I can test and demonstrate without worrying about another device. Another thing I selected Windows Mobile is its stability over Android and Windows Mobile are one of the emerging mobile technology in the world.

However Vehicle Driver's mobile device can be extensible to some other device such as Android or Apple since the backend of the system is completely independent as we interface with the client using Web APIs.

4.6 Other technologies

4.6.1 OR Mapping with Entity Framework

Entity Framework [5, 13] is an Object Relational Mapping Framework by Microsoft, now this code is open-source. Using an OR Mapping framework it is much easier to handle Relational databases in object oriented way.

As our main goal is to develop a system for a business rather than inventing new developer tools it is better to take advantages of above OR Mapping tools, then we can cut down some database development time but not data design time and focus on the business solution.

4.6.2 SignalR

SignalR [11] is another new Web technology. As we know in the web http or https always work as Request-Response means of communication between client and server ends. But with nowadays server push technologies like SignalR, developers can build applications to run in real-time without request-response. Here the beauty is when the server side event is triggered at a certain time in the server application life span, all the subscribed client nodes will be notified at the same time.

This is advantageous for the solution when communication messages are coming from Drivers Mobile device to the server. At the point the message hits the server, server application can notify each Operator desktop with the updated state.

4.6.3 Windows Workflow

Windows workflow foundation[15] is a framework that users can create workflows of systems in applications.

Workflow foundation can be used to create simple scenarios to complex enterprise level solutions such as order processing. Workflow foundation comes with a programming model and hosting and customizable workflow engine. Tools like Visual Studio can be used to create various type of workflows easily and quickly by the given user interactive diagram editors.

Windows workflow are used in this project to create the automated vehicle-driver selection, where multiple criteria are used to do the same. Since Workflow uses rich user interface to design the workflow it is easy to create them in the solution.

Chapter 5. Analysis and Design

Chapter 5 is intended to discuss the analysis of the business and the modeling of the analysis for the software system. This chapter will include Use-case diagram of the system, architecture diagram and various other diagrams which are used in analyzing and design the system such as Layer diagram and Class diagrams.

5.1 Use-cases

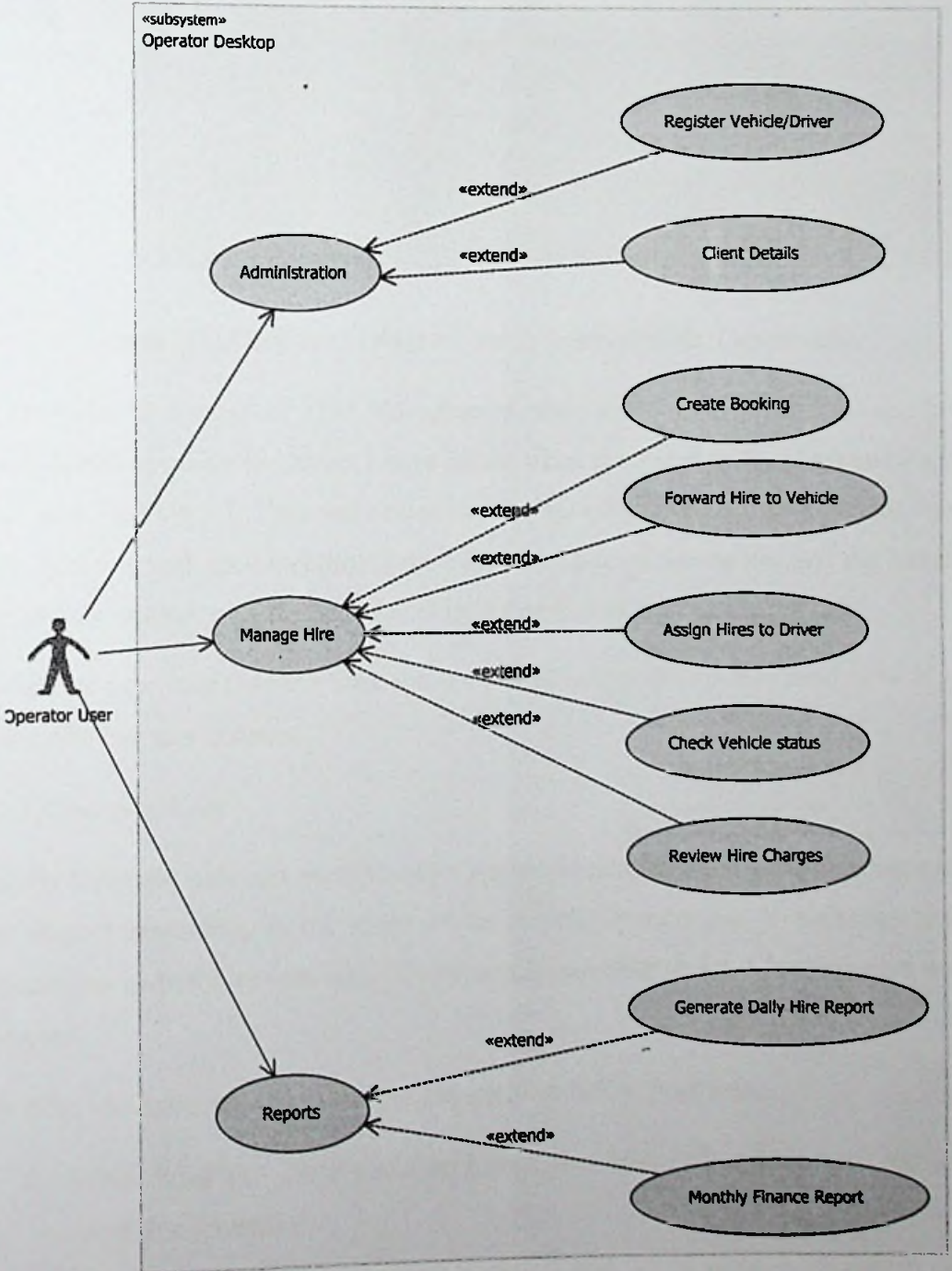


Figure 5.1.1: Use-case Diagram which interacts with 'Operator' actor

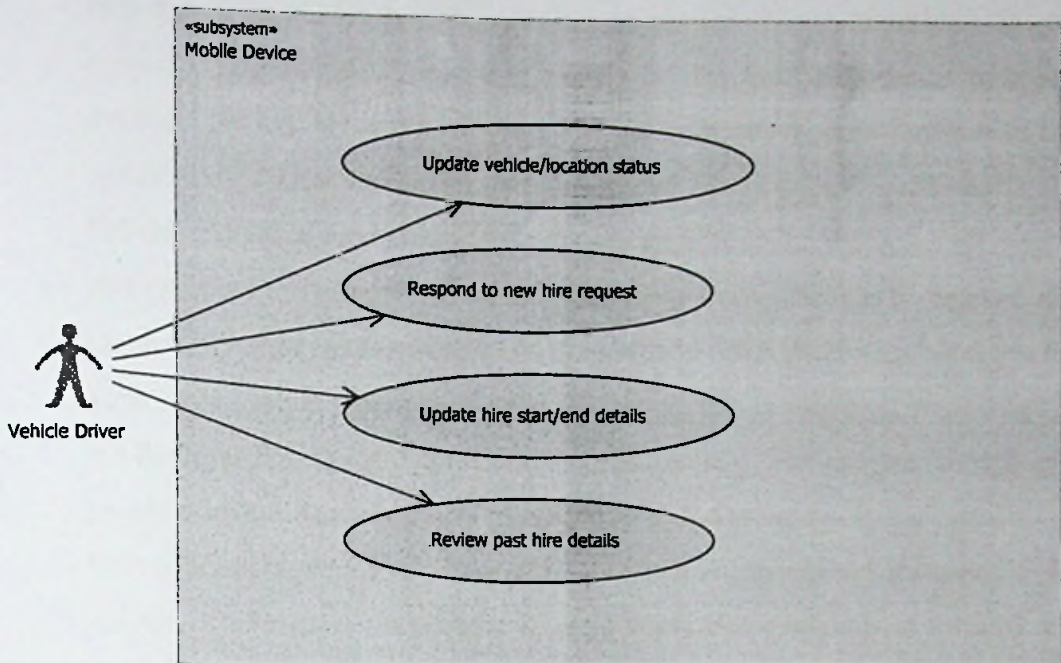


Figure 5.1.2: Use-case Diagram which interacts with 'Driver' actor

In the business domain of Taxi Management, there are six top level functions have been identified inside the scope I have taken. They are listed in the diagrams Figure 5.1.1 and Figure 5.1.2. They were separated to two sub systems as the Operator Actor only interacts with the Operation desk client interface of the system and the Vehicle driver only interact with the Mobile device he carries in the vehicle.

For all the functions Operator user and the Driver user performs they need to be log in to the relevant sub systems.

5.1.1 Operator User

Mainly Operator user can perform tasks related to administration, Hire management and Report generation. In the scope of the project Operator has Vehicle and driver registration in to the system and Clients details handling under Administration main use-case.

For Hire Management, Operator user can perform below functions

- Create Booking – New bookings he/she gets from calling clients are recorded using this screens.
- Forward Hire to Vehicle – New bookings will not be hires in the system until they get assigned to a vehicle-driver and confirmed by the driver. To get the

two things done, the booking can be forwarded to vehicle-drivers in the system. They should be operating vehicle drivers for the particular date and the vehicles can be sorted according to the currently parked location in the system in the Desktop screen. The forwarded booking can be viewed in the Drivers mobile as described in the next sub section.

- Assign Hires to Driver – Any number of driver responses can be come to the Operator Desktop as response to the forwarded bookings. But operator has the authority to select the most appropriate vehicle-driver based on some rules to get the hire. This assignment is done in this use-case. The operator should give reason why he/she assign the hire to vehicle in the system.
- Check Vehicle Status – There are situations to check the statuses of the operating vehicles in the system. Such as when a new booking is forward to a vehicle-driver and whether the vehicle-driver is operating in a particular day or to know the reason for non-operating vehicles.
- Review Hire Charges – On completion of a hire, the driver will update the hire status with the charges and other information. Those information can be obtained to the Operator desktop by the Operator user and act on when any dispute occurs.

‘Forwarding of Hires to Driver’ and ‘Assign Hires to Driver’ needed to be select drivers according to a set of rules by the business. So that these two steps in the flow were automated to select most probable driver-vehicle for the given hire by using a decision making method. This logics and the decision making method is described in the next chapter of implementation.

As in Report generation use case, there are two main reports can be generated by the Operator user for daily basis to get the summery of hire details of the day and another report on monthly basis for financial records in the institution.

5.1.2 Vehicle Driver User

Vehicle Driver user has following function in the system.

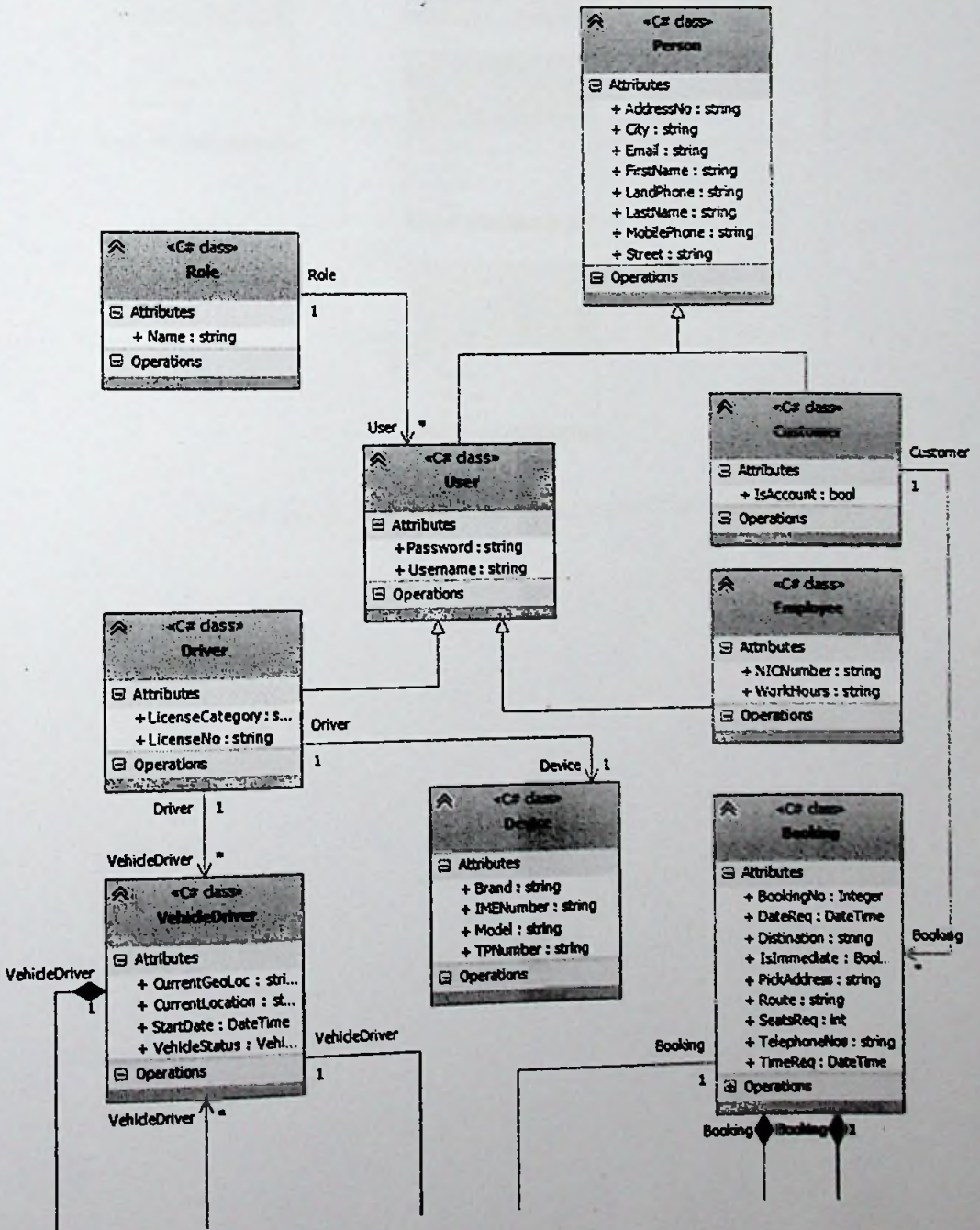
- Update Vehicle/Location Status – Using the Mobile device he can update the vehicle details and the current location he parked. These information will be fed to the SSS and reflected on the Operator Desktops (ODC) too.
- Respond to New Hire – Driver can view forwarded bookings to them in the mobile screen and he can respond to them using the same device screen. This function will update SSS and ODC
- Update Hire Start/end Details – When the driver starts a new hire he needs to enter some meter details, start time, and at the completion of the hire also. Those information will get updated on the SSS.
- Review Past Hire details – Driver can review past hires he made only for viewing purpose, they are not allowed to be amended by the driver.

Chapter 6. Implementation

In this chapter I will be giving a rough idea about how the system architecture and the domain design would be. There is a certain to make amendments to the domain classes as it goes developing the system components.

6.1 Domain Model

Domain model is depicted in the Figure 6.1.1. Up to now the model can be seen such but it may vary when more analysis and developments happen.



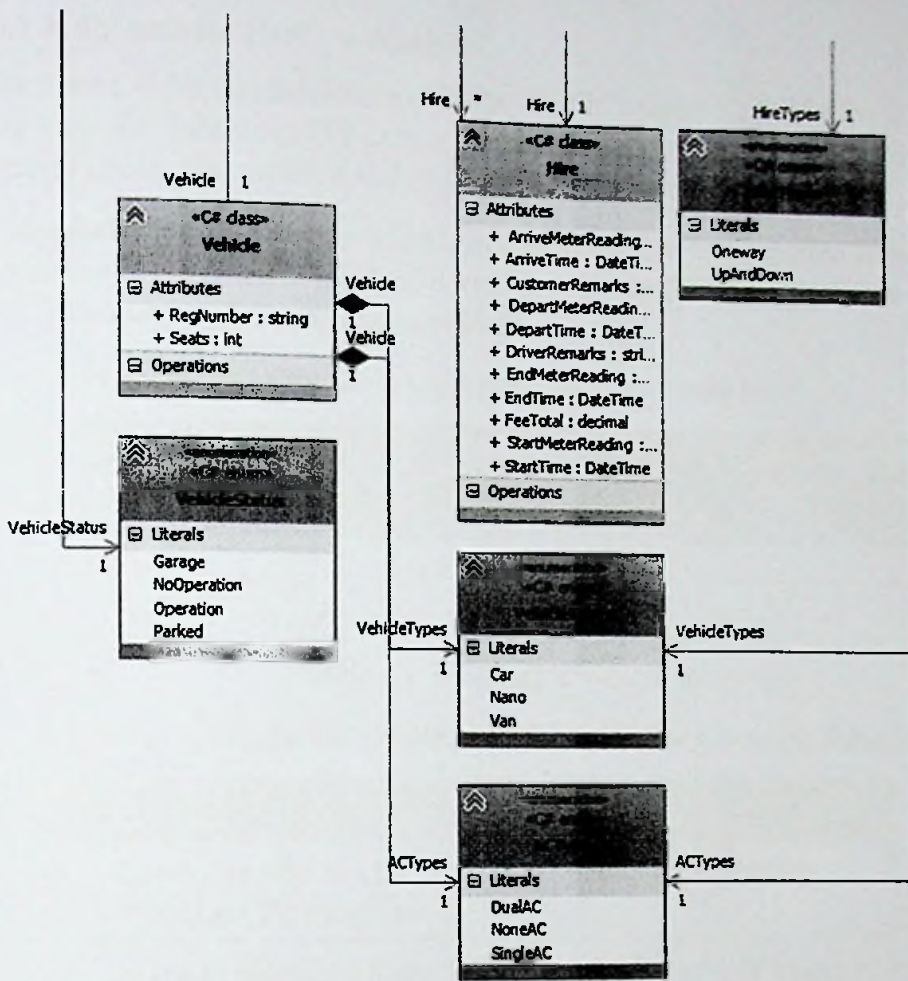


Figure 6.1.1: Domain-Class Diagram of the system

6.2 Multi-criteria Decision Analysis

As I have explained the business perspective in section 5.1.1 of selecting a driver for the new hire (booking) SSS Driver (Vehicle) selection for a given hire in the hires list comes handy with decision making ability by the system itself.

I decided to use some other technique than coming to a conclusion based on rules but to derive a Weighted Sum [2] to decide the vehicle-driver for the given booking. The implementation of the same is as follows.

Example of Driver (Vehicle) list of a particular day would be

Vehicle number	Driver Name	Hire miles of the day (km)	Inhouse /outside driver	Status	Distance from Client location (km)
V1	D1	20	False	Parked	20
V2	D2	45	True	Onhire	10
V3	D3	90	True	Onhire	5

Table 6.2.1: Example vehicle-driver list

As the business suggest the system should validate the following rules

No	Immediate hire
1	The driver who has the minimum hired miles of the day should get the next hire
2	In house should be more likely than outside vehicle to get the next hire
3	Status 'parked' vehicles more probable to get the next hire
4	The minimum distance from Parked location to client location should get the top most priorities to get the next hire.
No	Hire of nextdayonwards
1	The driver who has the minimum hired miles of the <i>week</i> should get the next hire
2	In house should be more likely than outside vehicle to get the next hire
3	Status 'parked' vehicles more probable to get the next hire

The flow of selecting the best suited driver is as following flow diagram

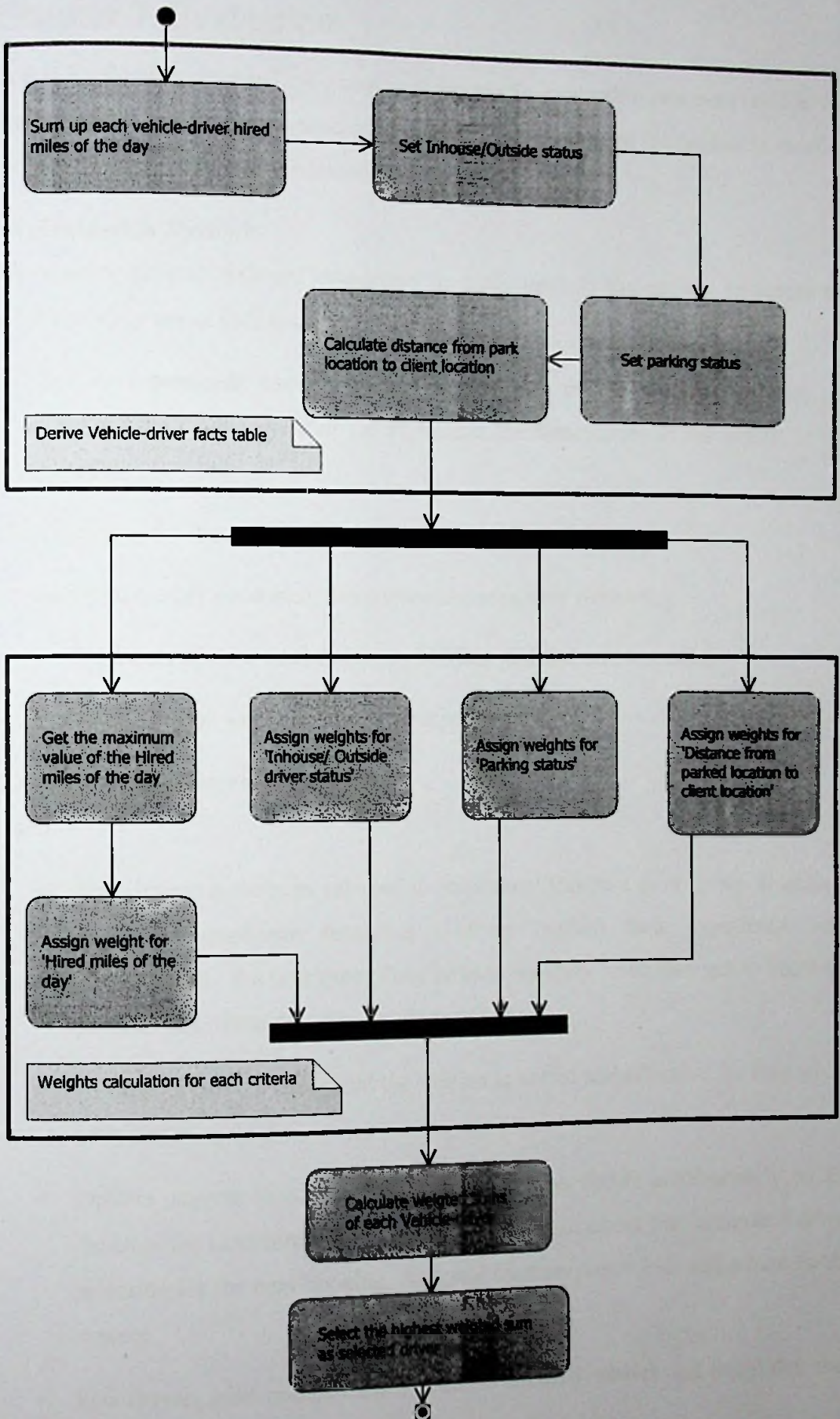


Figure 6.2.2: Flow chart of selecting best vehicle-driver

Chapter 7. Evaluation

In this Evaluation chapter I am going to elaborate the evaluation process I used in this project which is a structured process of assessing the success of the project in meeting the goals and to reflect on the lessons learned.

7.1 Evaluation Methods

There were several different evaluation methods used in the project to assess the solution. They are as follows.

Qualitative assessment could be done at the end of the solution delivery by interviewing the management of the client and the direct users of the system in the organization.

The following feedbacks have been collected from them.

For software quality assurance, there were two measures were setup

1. Unit tests of code modules using MSTest tool and unit test cases
2. Code reviews were conducted with peer reviews.

7.2 Results and Outcomes

Users –

- Few drivers have been selected to implement this as a pilot group as initially have less hardware resources. Drivers explain their experience once introduced for the first time. They have found easy when they get to know the possibilities of the driver's mobile application.
- Almost all drivers agree that the system is useful and effective for their given activities.
- Drivers suggest to capture the location by gps update automatically on the desktop for next hire. They have much concern about the automated driver selection for the next booking. And still have no proper idea and a trust on the system.
- Few drivers have complained that this is a totally useless and found that they are not willing to operate touch phones.

- Desktop Operators have been interviewed and found that they getting a benefit using the system for their activities.
- They suggest to implement a call center application to answer calls through the machine and then the application gives the call in client information immediately.

Management –

- Management has decided to continue the current manual process parallel with the new system
- Have selected a group of drivers for piloting the project and evaluate the efficiency and effectiveness of their activities after few months elapsed.
- So this is early and middle of the monitoring time.
- However the management has a positive mind with the feedback of the users and the slight business growth they experience.

7.3 Discussion

Since the project does not have Business Analysts or Quality Assurers the software solution cannot be evaluated against the business need correctly. The best possible way to evaluate it only by reducing the code level bugs and install the solution at clients end and let the users to use the system. Then by their feedbacks changes needed to the system can be extracted.

However there are few functional requirements have been missed to be implemented due to the limited time frame. Among them most important is generating management reports. This functionality is missing in the current release of the software solution. And can easily be added as there data related to reports is collected by the system in the current database.

Chapter 8. Discussion

8.1 Comparison to the other solutions

Main and the 'at a glance' difference in my solution compared to the other Taxi Management solution is using of smart phones for drivers end communication. Where others have used dedicated and specifically designed devices for the same role.

As this is a short term project the scope has been reduced to develop a feasible solution in the given time frame. Otherwise this could be a sophisticated system as I'm going to discuss in next section.

In the hardware level the industrial software have taken advantages of high performance servers, mirror and failover servers for non-interrupted operation on main server failures. Similar to them this solution also can be highly scalable to that level when needed. All the technologies and the architecture of the system are support scalability in the system.

8.2 Further developments

- Features for Clients to book online or using an mobile app can be implemented
- Same device on the Drivers end can be used to track the vehicle location and map rendering can be implemented on Operation Desks.
- To minimize hands usage in the driver's end device voice command based functions can be introduce to the VDM.

Here I have listed some of the modification we can implement to extend the functionality of the proposed system. There will be more domain specific requirements can be added to the above list

Chapter 9. References

- [1] Brad Green and ShyamSeshadri (2013), *AngularJS*
- [2] Denis Bouyssou, Thierry Marchant, Marc Pirlot, Patrice Perny, Alexis Tsouki`as, Philippe Vincke (2007) *EVALUATION AND DECISION MODELS: a critical perspective*
- [3] Jonathan Chaffer, Karl Swedberg (2009), *Learning jQuery 1.3*
- [4] Joseph Albahari and Ben Albahari (2010), *C# 4.0 IN A NUTSHELL*
- [5] Julia Lerman and Rowan Miller (2012), *Programming Entity Framework: Code First*
- [6] Mark Massé (2012), *REST API Design Rulebook*
- [7] Tomasz Szostak (2013), *Windows Phone 8 Application Development Essentials*
- [8] <http://www.digital-dispatch.com/>
- [9] <http://www.cabmanagementsystem.com>
- [10] <http://www.asp.net/web-api>
- [11] <http://www.asp.net/signalr>
- [12] <https://angularjs.org/>
- [13] <http://msdn.microsoft.com/en-us/data/ef.aspx>
- [14] <http://dev.windows.com/en-us/develop>
- [15] <https://msdn.microsoft.com/en-us/library/vstudio/ms734631%28v=vs.90%29.aspx>