

System Engineering of Taxi-Booking Mobile Application (TaxiHero)

Taiwo Mustapha O, Siti Nurhafizza Maidin, Kamilia
Kamardin, Suriani Mohd Sam, Noor Azurati
Ahmad, Azizul Azizan.

Razak Faculty of Informatics and Technology,
Universiti Teknologi Malaysia, Kuala Lumpur .

Malaysia-Japan International Institute of
Technology, Universiti Teknologi Malaysia,
Kuala Lumpur .

Wireless Communication Centre, Universiti
Teknologi Malaysia, Kuala Lumpur .

mustytaiwo@gmail.com,
sitinurhafizzamaidin@yahoo.com,
kamilia@utm.my

Article history

Received:
2 July 2019

Received in revised form:
25 August 2019

Accepted:
30 September 2019

Published online:
3 October 2019

Taiwo Mustapha O
mustytaiwo@gmail.com

ABSTRACT

The project revolves around day to day of activities of persons moving from one point to another point, with more emphasis on the taxi transportation service systems. The company in check dreamed up the idea of solving the ineptitude and unfavorable conditions such as trying to get a taxi in heavy rainfall or intense sunlight. The Taxi-Booking Mobile App aims to contribute to the revitalization of the taxi transportation industry, by harnessing the potency of the advancement made in information and communications technology. The TaxiHero Company came up with a mobile solution empowered with modern technology to expound those ineptitudes and unfavorable conditions. The TaxiHero Taxi-Booking App will be a mobile based application to be used for the Android and IOS platforms. In order to create a product of the highest quality within the time constraint set by the client company, the project shall be undertaken with heavy emphasis on achieving realistic goals by conducting the system engineering lifecycle on the TaxiHero TaxiBooking App. Thus, it will help to guide engineers in producing high quality product as it involves various engineering and design disciplines for successful development.

Keywords: *TaxiHero, Places, Ineptitude and Unfavorable Conditions, Taxi-Booking Mobile Application*

1. Introduction

In this modern era, utilization of the modern technology such as the internet is popular because it helps to ease people's daily life. For example, people can make online booking for taxi through their phone without needing to walk to the main

street. In [1], they proposed and implement the taxi booking and schedule system to help people to book a taxi through their phone. The system implemented also can help to reduce the congestion for the taxi on the road and can help the taxi driver to save their money on oil consumption. It can be seen that the advantages of the online booking system can surpass the traditional way especially in the cities. Usage of the taxi-mobile booking can save people's time because they can just make a reservation online using their phone while doing their work. In this paper, the TaxiHero application is introduced and the development of the system is done based on system engineering roles.

* *Corresponding author. kamalia@utm.my*

The TaxiHero is taxi-booking mobile application to help facilitate communication and hiring services between a taxi driver and potential passenger. The TaxiHero application is focused on harnessing the potency of the advancement made in information and communications technology. The TaxiHero application also want to give advantage to both passenger and client such as digital map for passenger and taxi driver in order for them to arrive at destination on time. A look at the project in point, one would think it is a very simple project which not requiring any sort of complex system engineering process or methodology. At face value, that may be correct, but when you look at the entire work that goes into its development, one would be humbled to see how complex it is [2]. The Taxi-booking mobile application will radically take advantage of the internet, mobile apps proliferation, cloud computing [3], Geo-positioning system services [4], and web socket technology [5]. It will be developed as a hybrid mobile application, that is, it takes use of both web technologies and mobile technologies, therefore making it cross platform.

When developing the system based on system engineering process, the possibility to detect any deflection can happen and also avoid catastrophic effects on the system and the people. Systems engineering provides the framework to build large, complex elements of infrastructure by integrating the engineering disciplines into a structured development process [2]. It is covering the entire lifecycle of the system, starting with the identification of the need, the conceptual and detailed design processes, development, deployment, operation, maintenance, and disposal of the system [6].

2. Background of Study

In the sequel of activities leading to the development of the Taxi-booking Mobile application project, it was noticeable that there is a public outcry or rather need for a secure and efficient way a person could get a taxi without long waiting at the taxi stops or bus stops, or the illegal haggling perpetrated by the taxi driver who charge unknowing customers exorbitantly, thus giving birth to the notion of creating a solution that will harness the potency of information technology to facilitate and ensure a smooth transaction between taxi drivers and customers.

Many organizations have risen to develop such applications, but all or most lack the required and necessary coverage needed to make the app usable on a wide range and that is where the TaxiHero comes in. The TaxiHero Company is a venture nurtured and managed by Nova Founders Limited [7]. The company's directive and goal is to create a mobile solution that will enhance and ensure a smooth connection, transaction, and relationship between a taxi driver and a passenger. Their services shall be used in Hong Kong, Philippines, and sooner in Malaysia. The App is required to cover a wide range of users and support different languages, more specifically Chinese and English.

In order to see the development of the TaxiHero system, the systems engineering process is implemented where the systems engineering process involves 3 phases which are concept development, engineering development and product post development. For concept development, it consists of need analysis, concept exploration and concept definition stages. Meanwhile, for engineering development, it consists into 3 stages which are advance development, engineering design and integration and evaluation. In product post development phase, it consists stages of the production and operation and support. Figure 1 shows the stages involved in engineering development phase and methods used in each stage. Further explanation of the phases, stages and methods involved in the TaxiHero development process is discussed in section 5, 6 and 7.

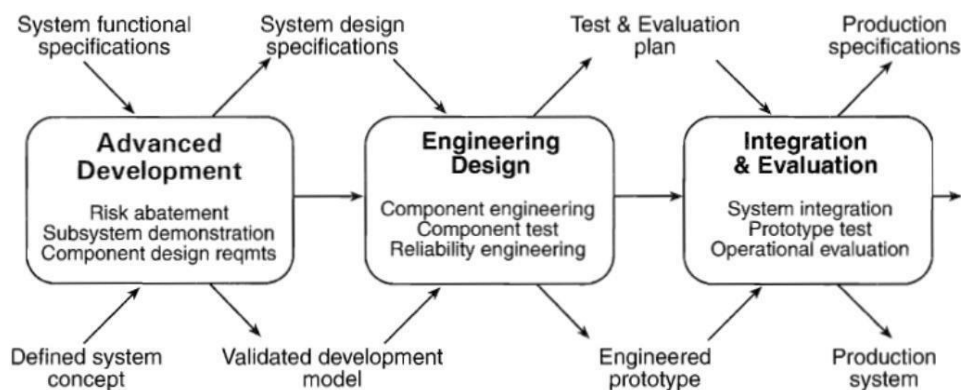


Figure 1. Engineering Development Phase [8]

3. Problem Statement

The major problem identified that led to the needs of taxi app is the difficulties and the unfavorable conditions that are experienced in the cause of trying to get a taxi, especially during unfavorable weather conditions such as a heavy rainfall or a very intense sunlight. It becomes a necessity to get a taxi to transport oneself from one place to the other. Another problem that could be identified include the following:

- a. Illegal haggles done by taxi drivers to cheat unknowing customers of their money with regard to directly billing customers exorbitantly instead of using the billing meter.

- b. Insecurity, with regards to trusting the chosen taxi driver.
- c. Inability of drivers to get to the drop off location in time using the shortest possible route.
- d. Inability of customers to report illegal activities or miscreant behavior that might have been perpetrated by the driver directly or indirectly to him/her.

4. Objectives

The followings are the objectives for this project:

- a. To develop a mobile-based application that would facilitate the hiring of a taxi for the movement of people from one place to the other.
- b. To expound certain complexities and difficulties that may arise in the cause of hiring a taxi.
- c. To reduce the problem of illegalities usually perpetrated by the taxi drivers, with regards to illegal haggling and what not.
- d. To ensure customer satisfaction and safety when they hire or ride in a taxi.
- e. To provide a digital map as a solution to the need of clients to get to wherever they need to get to in time.
- f. To provide a medium of communication or reporting that will allow passengers report any issues they might have had with the taxi drivers.

5. Engineering Process

In order to see the process [9] of Taxi-Booking Mobile application, it needs to go through the system of engineering [10], which involve the concept development, engineering development and post development. In this section, all the phases will be discussed in depth.

5.1. Need Analysis

The Need Analysis phase of the concept development includes the requirements analysis, functional definition (in this case, the user requirements), and design validation.

5.1.1. Requirements Analysis: As the project is to follow an iterative or incremental methodology, expected features in the application are segregated into phases, or rather version release control. The first iteration is expected to have the basic function which is afforded signing up for all users in order to able passengers to book a taxi via the app, and a driver may accept or reject, ability for booking cancellation, and reason for it. Also to be present in the first iteration is a Geo coordinated mapping system that tracks and indicate driver's movement and passenger's position.

Second iteration is expected to include the ability for users to switch locale languages, view trip details such as estimated cost of commuting, add extra notes indicating where a passenger's at, give driver tips. Also expected in the 2nd iteration is the ability of the passengers to make advance bookings, a feedback and rating system is to be implemented. Also added is the ability for the users to transact trips within the app.

5.1.2. User Requirements: The users of the app will include just two types of users, the drivers and the passengers. The passengers will be able to download the app for free on their respective mobile platform application stores and will be able to make booking as soon as they sign up and log in into the app. On a successful login or launch, the app locates the current location of the passenger and indicates it as the pick-up location, after which a passenger may then proceed to select a drop off location, and then make his or her booking, which could then be accepted by any willing driver.

The driver users on the other hand, can also download the app and required to log in into the app. If the driver does not have an account on the app, the driver required to sign up first. On successful login or launch, the app will navigate the driver to the driver map screen which just includes a map showing the current location of the driver. When a passenger makes a booking, the app alerts drivers within 2km radius of the passenger making the booking. The driver is required to either accept or decline a trip request. If a driver accepts the booking, the driver information is shared with the passenger and on the driver side, a GPS location of the driver and passenger appears on the driver map screen. The driver is required to follow the path laid on the map screen until the driver gets to the prescribed locations.

Moreover, in the case of any cancellations either done by the driver for some reason or the other, or by the passenger, both users are required to give a reason why such actions have been performed. Another feature for which both users can both access is the language selection. The users are allowed to select between Chinese Mandarin and English languages. Mandarin in the sense that the app will be released in Hong Kong and for the app to be popular among the locals, it must cater for the needs of the locals.

5.1.3. Design Validation: There were no specific data collection method employed, however, most data the development team will work with is to be supplied by the client company. A little bit of research carried out indicated that most of the data collected to formulate the requirement were based on studying legacy app in such field. First movers such as MyTeksi and EasyTaxi were studied closely. Other taxibooking mobile application such as an Automated Taxi Booking and Scheduling System [1] and mobile application for taxi service company in Nigeria [11]. Other formulated requirements came through brain storming among the staffs of the client company.

Furthermore, a bit deeper research was carried out by some members of the development team. Taxi drivers were approached and sometimes a trip is taken just to be able to ask the taxi drivers how they feel about the app they may be currently using and what would they have preferred to see as features if an app were to be built to replace the current traditional dispatch by radio system.

6. Engineering Design

In this phase, emphasis will be placed mainly on the application decomposition into different units. A well detailed design is the key to a successfully developed and implemented system. A number of guiding principles are outlined which dominates or embodies the design of the application to ensure a good design quality is achieved. The major problem of design is the ineffectual way of which requirements and module interaction are depicted. So for easy understandability, as the app involves a large number of users who may not be tech savvy, the design will be endeared in line with the Keep It Super Simple (KISS) principle [12-14]. Reliability is achieved through emphasizing simplicity in the design and through extensive feedback from operational testing [15].

6.1. Architectural Design

Figure 2 depicts the architectural design of the Taxi-Booking Mobile Application. It clearly outlines the users of the app, shows the relationship between the passenger and the driver and how they interact with each other. The app accesses the API (developed using C#.NET) over an asynchronous HTTP request. The Node.JS serves as the server platform from which the apps are launched, monitored and managed. The Socket.IO, web socket client, connects the driver and passenger app to the Node.JS server, basically what a web socket does is keep the connection between two nodes open as long as the information keeps passing through. The Database is accessed by both the API and the Node.JS server. The database is a NoSql database created in MongoDB. A NoSql database basically stores data as series of objects, kind of like JavaScript Object Notation, JSON for short.

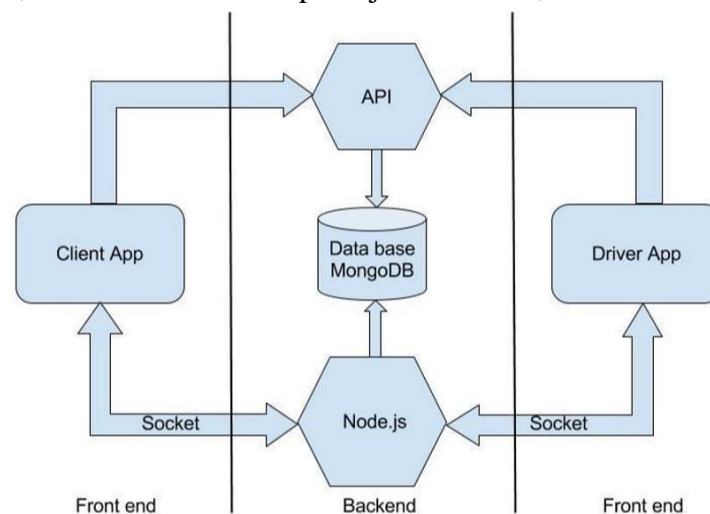


Figure 2. Architectural design of the application

6.2. Database Design [14]

The concept of the database structure would be displayed and expatiated in this paper. Figure 3 shows a conceptual entity relationship diagram of Taxi-Booking Mobile application database.

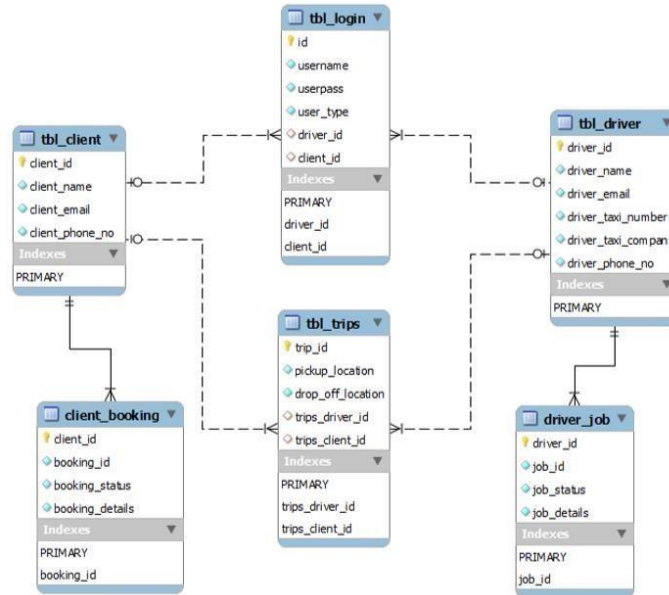


Figure 3. Conceptual entity relationship diagram of the Taxi-Booking Mobile app database

6.3. Interface Design

The mock design for the interface were given by the client company, the designs were given as both PNGs, PSDs, and JPEGs. The PSDs will be sliced and diced into smaller bits of PNG images to be fit into the application as it is being designed and the JPEG images will be used as a model for how the design structure of the app is meant to look. These are some of the images for which the app will be modeled after. The following screens are depicted for the users in the passenger side; they include the account screen, language selection screen, the map screen, the pick-up location selection screen, the sign-up screens, the password retrieval screen, the profile update and then the driver info screen. Figure 4 show sample of mock interface design of taxi-booking mobile passenger app.

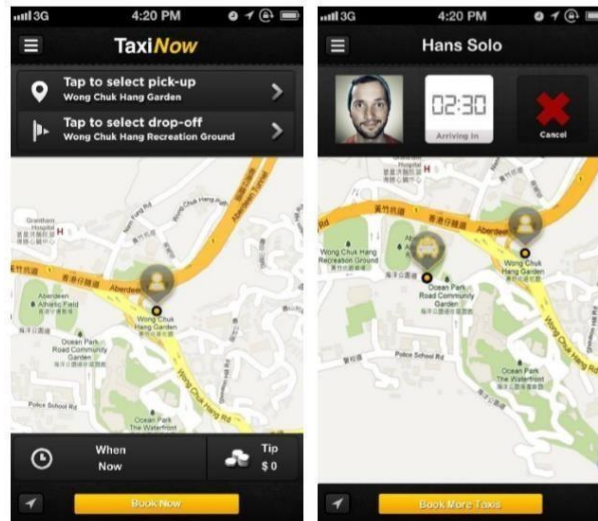
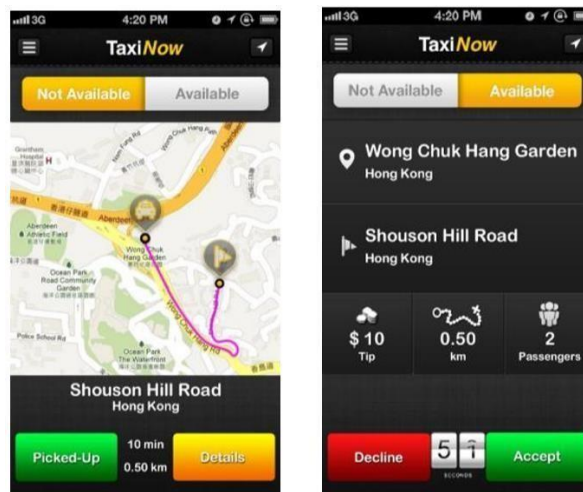


Figure 4. Sample of mock interface design of the taxi-booking mobile passenger app

The following interface designs were used for the driver app screen. The screen includes, but not limited to following: driver account screen, password change screen, driver map screen, countdown screen, the driver arrived screen, picked up the screen, dropped-off screen, driver profiles screen and jobs screen.



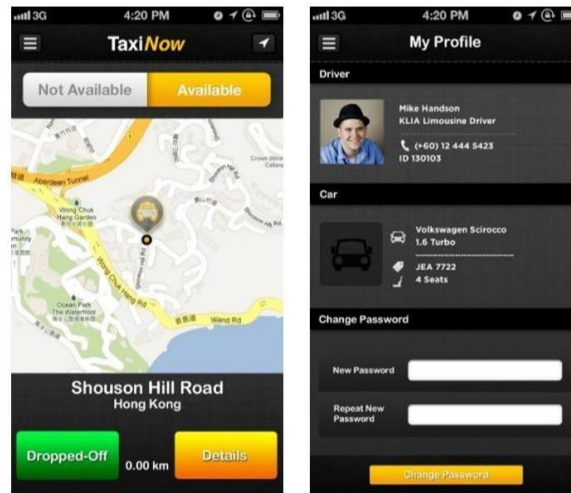


Figure 5. Samples of mock interface design of the taxi-booking mobile driver app

6.4. Implementation

Implementation refers to the manifestation of the analysis and designs with regards to specified requirements of the application. This subsection explains the implementation process, tools and technology employed during the development of the application, and other implementation issues.

6.4.1. Development Tools: The development tools are basically software tools and programming languages employed during the development of the project. The tools range in usage and complexity. The different tools, plus their description are explained upon in the following clauses. The hardware tools were all vendor based, so less emphasis will be placed on them in this paper.

For software application tools, Table 1 shows the software application implement in this project.

Table 1. Software Application Tools Implement in Taxi-Booking Mobile Application

Software application tools	Description
Eclipse IDE (Android Developer Tools)	Used for developing the android executable app of the taxi app. It was also used for the debugging procedures that ensued during development
Adobe Photoshop	Used for slicing and dicing PSDs and edit graphic images to be used in the app
Sublime Text	Used to write code and structure code files
Visual Studio 11	Used as a catalyst to hosting the API on the windows azure platform

Cocoa IDE (IOS)	Used to compile the source code of the taxi app into an IOS executable
-----------------	--

For servers, libraries, and plugins, Table 2 shows the deployment as follows:

Table 2. Servers, Libraries and Plugin deployment

Servers/ Libraries/ Plugins	Description
PhoneGap	This a technology that enables mobile app developer develop hybrid applications. Hybrid apps are apps that can release on multiplatform operation systems. The PhoneGap technology contains several JavaScript libraries and other platform-based development libraries, like Android, IOS, Windows 8, Windows Phone 7 & 8, and Blackberry
AngularJS Framework	This is a JavaScript framework that basically allows for development of single page applications. Through the use of various techniques, such as page routing, page controllers, writing directives and modules. It became a complete framework used for developing high-end web or mobile based applications
Syntactically Awesome Style Sheet (SASS)	It is used for writing CSS programmatically, by allowing programmers focus on actual coding and worry less on structuring and formatting CSS source files
Socket.IO	This is a JavaScript library written to enable open cross browser communication. By manipulating dome element, this library, with the idea behind the W3C web socket API implemented newly in HTML5, allows of an opened gateway between two nodes or browser or application for data streaming and communication
Node.JS	This is a server developed using JavaScript, that enables JavaScript driven applications run on servers, that is, imitating the usual server side languages driven applications like PHP, JSP and all that
MongoDB	This is a database that stores data as series of objects in collections, it is used in juxtaposition with Node.JS server
Ripple Emulator	This is a Google Chrome browser plugin that allows simulation of apps developed for mobile platforms on the chrome browser
Grunt Server	This is a notable Node.JS resources, it basically serves as a server that enables app to be run on the client browser

6.4.2. Implementation issues: This subsection will discuss on the implementation issues that possibly happen during application implementation. Table 3 will show the issues, cause and edifications for taxi-booking mobile application.

Table 3. Implementation issues

Issues	Cause	Edifications
Performance	The recurrent pattern of which object is being stored and manipulated with the use of several classes with anonymous methods and functions	The interface uses a single page with multiple boxes, boxes with accordions and tabs, for all activities, thereby limiting the problem on overall app performance. Also, only the initial performance on page loading is affected, performance is better than normal afterwards.
Consistency	The use of too many modules and functions, makes the structuring looks a bit confusing	The use of the AngularJS framework modules and directives to modularize services and controllers for each app page
Scalability	Performance failure along with increase of users	Dealt pre-emptively by the use of highly cohesive and less coupled modules
Security	Access to the app without proper authorization	The app requires a user login into the app before it is being used. Details of user log in are then cached in the phone browser storage services for a period of time before automatically refreshing and requesting the user to perform the login operation again
Real-time Issues	Data not being delivered in real-time till page refresh	The use of AJAX requests and Socket.IO to asynchronously interact with the server and update the page with server updates without page refresh
Concurrency Control	Data integrity affected by unsafe concurrency operations	Class-level variables were not used, hence two objects accessing the same data would not conflict
Flexibility	The system being stiff to changes and increments of functionality	A strict cohesive modular structure with loose coupling making the addition of more functional modules much easier

Adaptability	The system could fail to work on a different platform	The app was developed as hybrid apps with the supports of the PhoneGap libraries, plus calling the navigator user agent to determine the device platform brand would solve any adaptability issue
Fault Tolerance	The occurrence of invalid input or bugs which could crop up while the system is in operation	The apps contains strict validation rules and has been unit tested to minimize the effects of such faults

6.4.3. Testing: Regarding the testing process, the taxi app underwent some levels of testing. The major testing activities included system testing and user acceptance testing (UAT). The other levels of testing included integration testing and unit testing which were integral to the development process. a. Unit Testing

Involve testing different module unit independently. This level of test is usually carried out at the source or code level to check for language specific programming errors or bugs, such as syntax error, logical error, and to test particular functions to ascertain their functionality and correctness.

In the case of white-box testing, which is performed at the unit testing level, the UI is ignored and inputs and outputs are tested directly at the code level, mostly with the use of the system command console, and the results are compared against specifications. This level of testing ignores the function of the program under the test and will focus only on its code and structure of the code. b. Integration Testing

There are three primary modules involve in the integrated testing of the application which are the driver GUI module, the Logic module and the Backend Module. These three components were all integrated and tested to ascertain compatibility The three components infuse together will form most of the functionality within the test domain. An incremental testing strategy was used to complete the integration. Table 4 shows the integrated testing of taxi-booking mobile application. c. System Testing

The goals of system testing are to detect faults that can only be exposed by testing the entire application as a whole or some major part of it. Generally, system testing is mainly concerned with areas such as performance, security, validation, load or stress, and configuration sensitivity. Testing exercise was carried with regards to those criteria in mind and the black box method of testing will be applied. d. User Acceptance Testing

The user acceptance testing shall be carried out by the potential users and passenger of the application, in this testing level, emphasis is placed more on testing the application against the requirements defined in the list requirements given by the client company. Corrections shall be made as required, and documented in the test logs.

The primary tester, which is the client company and appointed tester of the application will however test the system based on the test case plan outlined by the client company.

Table 4. Integrated testing

Integrated testing	Description
GUI Module	Provides a simple GUI where the user can perform different actions (function). This module will be initially tested separately from the logic that guides it, to check for responsive effects with regards to different screen sizes, and to ascertain interface design correlates with the design given by the client company. The test of the GUI was carried out on both the passenger taxi app and the driver taxi app together, as they both have seeming similar designs and aesthetics.
Logic Module	Provides the logic that drives the flow of design and program of the interface in the GUI module of the apps. The logic was developed using JavaScript framework known as AngularJS with other JavaScript libraries. The logic module serves as the controller between the interface and the Backend. The Logic module is used to get all required data from the backend module, connect to the backend socket, basically, it serves as the medium through which data is transmitted from the backend to the front end and vice versa. The Logic Module and the GUI module together make the front end of the application.

Backend Module (Node.JS, API, MongoDB, Socket.IO).	<p>The backend module is where all the ‘magic’ happens. The backend module basically consists of the database, the server and the API. The API was written in C#.NET and initially hosted on a windows azure cloud platform before being moved to a very ‘discrete’ location. The logic module makes call to the API whenever it needs to perform some specific types of functions. If the functions require access to the database, the API makes the call to the database to retrieve the information or rather save the sent data.</p> <p>The Node.JS server serves as the bed layer for which the database is being hosted and also the socket connection gateway which is use for open connection and communication between the two apps i.e. the client app and the driver app. And as for the database, MongoDB was used, it’s a NoSql database that saves data as series of Object.</p>
--	---

7. Post Development

With the app fully developed and successfully tested. The application will be hosted on various mobile operating system application stores, more specifically iTunes store for the iOS power phones, and the Play store for the Android powered devices.

Additionally, the business unit of the client company will handle the commercialization for the app, i.e. getting the targeted users to adopt the application and give feedback to identify areas of improvement. Subsequently, more iterations of the application will be released as technology and user demand advances and changes.

8. Conclusion

In conclusion, the project as embarked upon was successful, the application is now being launched in Hong Kong and in the Philippines, and was available on the Apple’s iTunes stores for the IOS versions and on the Google play store for the Android range of mobiles. The App name was changed from TaxiNow to TaxiHero, so for normal passengers, the app is named as TaxiHero and for the drivers the app is named as TH driver.

The appropriate recommendation that could be suggested, as the app is close to usable perfection, is that the company keeps it intact and not be complacent on the feedbacks they get back from the users. A good suggestion would be for the company to go into making custom taxi-booking applications for private Taxi Fleet operators that would enhance the market strength of the company and give it an advantage over its market rivals.

Another suggestion, that could be given, is for the company to embrace the notion of enhancing the security details of the app and put in measures to really ensure user

safety, like for example, a feature that could allow a user the option of letting the company listen in on the trip activities could be incorporated into the app.

Acknowledgments

The authors wish to thank Advanced Informatics Department of Razak Faculty of Informatics, Universiti Teknologi Malaysia (UTM) and the Ministry of Education (MOE) for providing the Research Grant (Vote No:15H97).

11. References

- [1] Awajan, A. (2013). An Automated Taxi Booking and Scheduling System. EUROSIM Congress on Modelling and Simulation. 10 – 13 September 2013, Cardiff, UK. [2] Rechtin, E. (1991). *Systems Architecting: Creating & Building Complex Systems* (1st Edition). Southern California: Prentice Hall.
- [3] Esposito, C., Castiglione, A., Pop, F., and Choo, K.-K. R. (2017). Challenges of Connecting Edge and Cloud Computing: A Security and Forensic Perspective. *IEEE Cloud Computing* 4 : 2 (2017) 13-17.
- [4] Jing, W., Lu, X., Zhao, D., and Wu, J. (2010). Research on the Technology of Common-View Based on the GEO Positioning System. *International Conference on Information Engineering and Computer Science*. 25 – 26 December 2010, Wuhan, China.
- [5] Jiang, F.-Y. and Duan, H.-C. (2012). Application research of WebSocket technology on Web tree component. *International Symposium on Information Technologies in Medicine and Education*. 3 – 5 August 2012, Hokodate, Hokkaido, Japan.
- [6] Babiceanu, R. F. (2010). Systems engineering life-cycle modeling approach to wireless sensor networks. *IEEE International Systems Conference*. 5 – 8 April 2010, San Diego, CA, USA.
- [7] Nova Founders Capital. (2017). Retrieved October 18, 2018, from Nova Founders Capital: <http://www.novafounders.com/>
- [8] Haskins, C. (2006). *Systems Engineering Handbook: A Guide For System Life Cycle Processes And Activities*. International Council on Systems Engineering (INCOSE). 2006. INCOSE-TP-2003-002-03. [9] Laudon, J. P., and Laudon, K. C. (1996). *Essentials of Management Information Systems: Organization and Technology*. Saddle River, NJ, USA: *Essentials of Management Information Systems: Organization and Technology* (2nd Edition).
- [10] Sommerville, I. (2011). *Software Engineering* (9th Edition). Massachusetts: Pearson Education.
- [11] Adewumi, A., Odunjo, V., and Misra, S. (2015). Developing a mobile application for taxi service company in Nigeria. *International Conference on Computing, Communication and Security (ICCCS)*. 4 – 5 December 2015, Pamplemousses, Mauritius.
- [12] Fulcher, E., and Patil, S. (1994). A Four Asic Mcm-c, the "Kiss" Principle, and the Next Generation Silicon. *Proceedings of the International Conference on Multichip Modules*. 13 – 15 April 1994, Denver, CO, USA.
- [13] Smith, B. (2012). How to KISS: The Art of Keeping it Simple, Stupid. X Double Dot, LLC, 1-6.
- [14] Terano, T. (2008). Beyond the KISS Principle for Agent-Based Social Simulation. *Journal of Socio-Informatics* 1 : 1 (2008) 175-187.
- [15] Seder, G. (1989). An Application Of The Kiss Principle In The Design Of A Human-Powered Submarine. *Proceedings OCEANS*. 18 – 21 September 1989, Seattle, WA, USA.