# Image result for school of computing asia pacific collegeCarkila

# Vehicle Rental Mobile Application

Project Documentation Submitted

To the Faculty of School of

Computing and Information Technologies

Of

Asia Pacific College

In Partial Fulfillment of the Requirements for the subject

SYSADD 1 and 2

By

Martha Angela Salamat

Justin Jullian Omac

Brian Olores

John Ryan Pasaoa

January 9, 2018

# List of Figures

[Figure 1: Source of rentable vehicles 21](#_Toc503229957)

[Figure 2: Difficulty of finding rentable vehicles 22](#_Toc503229958)

[Figure 3: Time it takes to find rentable vehicles 23](#_Toc503229959)

[Figure 4: Usefulness of the proposed project for renters 24](#_Toc503229960)

[Figure 5: Ways of advertising rentable vehicles 25](#_Toc503229961)

[Figure 6: Pricing preference in renting vehicles 26](#_Toc503229962)

[Figure 7: Usefulness of proposed project for vehicle owners 27](#_Toc503229963)

[Figure 8: Functional Decomposition Diagram 31](#_Toc503229964)

[Figure 9: Context Diagram 32](#_Toc503229965)

[Figure 10: Diagram 0 33](#_Toc503229966)

[Figure 11: Entity Relationship Diagram 34](#_Toc503229967)

[Figure 12: Class Diagram 40](#_Toc503229968)

[Figure 13: Object Diagram 41](#_Toc503229969)

[Figure 14: Use Case Diagram 43](#_Toc503229970)

[Figure 15: Create Account Activity Diagram 51](#_Toc503229971)

[Figure 16: Rent Vehicle Activity Diagram 52](#_Toc503229972)

[Figure 17: Manage Listing Activity Diagram 53](#_Toc503229973)

[Figure 18: Approve Request Activity Diagram 54](#_Toc503229974)

[Figure 19: View Profile Activity Diagram 55](#_Toc503229975)

[Figure 20: View Feedback Activity Diagram 56](#_Toc503229976)

[Figure 21: Generate Reports Activity Diagram 57](#_Toc503229977)

[Figure 22: Create Account Sequence Diagram 58](#_Toc503229978)

[Figure 23: Rent Vehicle Sequence Diagram 59](#_Toc503229979)

[Figure 24: Manage Listing Sequence Diagram 60](#_Toc503229980)

[Figure 25: Approve Request Sequence Diagram 61](#_Toc503229981)

[Figure 26: View Profile Sequence Diagram 62](#_Toc503229982)

[Figure 27: View Feedback Sequence Diagram 62](#_Toc503229983)

[Figure 28: Generate Reports Sequence Diagram 63](#_Toc503229984)

[Figure 29: State Diagram of Create Account Use Case 64](#_Toc503229985)

[Figure 30: State Diagram of Rent Vehicle Use Case 64](#_Toc503229986)

[Figure 31: State Diagram of Manage Listing Use Case 65](#_Toc503229987)

[Figure 32: State Diagram of Approve Request Use Case 65](#_Toc503229988)

[Figure 33: State Diagram of View Profile and Write Feedback Use Case 66](#_Toc503229989)

[Figure 34: State Diagram of Generate Reports Use Case 66](#_Toc503229990)

[Figure 35: Timing Diagram of Create Account Use Case 67](#_Toc503229991)

[Figure 36: Timing Diagram of Rent Vehicle Use Case 67](#_Toc503229992)

[Figure 37: Timing Diagram of Manage Listing Use Case 68](#_Toc503229993)

[Figure 38: Timing Diagram of Approve Request Use Case 68](#_Toc503229994)

[Figure 39: Timing Diagram of View Profile and Feedback Use Case 69](#_Toc503229995)

[Figure 40: Timing Diagram of Generate Reports Use Case 69](#_Toc503229996)

[Figure 41: Package Diagram 70](#_Toc503229997)

[Figure 42: Communication Diagram 71](#_Toc503229998)

[Figure 43: Component Diagram 72](#_Toc503229999)

[Figure 44: Deployment Diagram 73](#_Toc503230000)

[Figure 45: Composite Structure Diagram 74](#_Toc503230001)

[Figure 46: Interaction Overview Diagram 75](#_Toc503230002)

# List of Tables

[Table 1: Carkila vs. TNVS apps 8](#_Toc503230003)

[Table 2: Carkila vs. Existing Car Rental Systems 8](#_Toc503230004)

[Table 3: SRS Output 28](#_Toc503230005)

[Table 4: SRS Input 29](#_Toc503230006)

[Table 5: SRS Process 30](#_Toc503230007)

[Table 6: SRS Performance 30](#_Toc503230008)

[Table 7: SRS Control 30](#_Toc503230009)

[Table 8: Data Dictionary of User Table 35](#_Toc503230010)

[Table 9: Data Dictionary of Vehicle Owner Table 35](#_Toc503230011)

[Table 10: Data Dictionary of Vehicle Renter Table 36](#_Toc503230012)

[Table 11: Data Dictionary of the Vehicle Table 36](#_Toc503230013)

[Table 12: Data Dictionary of Vehicle Type Table 37](#_Toc503230014)

[Table 13: Data Dictionary of Booking Table 37](#_Toc503230015)

[Table 14: Data Dictionary of Payment Table 38](#_Toc503230016)

[Table 15: Data Dictionary of Feedback Table 38](#_Toc503230017)

[Table 16: Data Dictionary of Reports Table 39](#_Toc503230018)

[Table 17: Event Table 42](#_Toc503230019)

[Table 18: Create Account Use Case Description 44](#_Toc503230020)

[Table 19: Rent Vehicle Case Description 45](#_Toc503230021)

[Table 20: Manage Listing Use Case Description 46](#_Toc503230022)

[Table 21: Approve Request Use Case Description 47](#_Toc503230023)

[Table 22: View Profile Use Case Description 48](#_Toc503230024)

[Table 23: Write Feedback Use Case Description 49](#_Toc503230025)

[Table 24: Generate Reports Use Case Description 50](#_Toc503230026)

**Table of Contents**

[List of Figures 2](#_Toc503230029)

[List of Tables 3](#_Toc503230030)

[I. Introduction 6](#_Toc503230031)

[**1.1** **Project Context** 6](#_Toc503230032)

[**1.2** **Purpose and Description** 7](#_Toc503230033)

[**1.3** **Objectives** 9](#_Toc503230034)

[**1.4** **Scope and Limitations** 9](#_Toc503230035)

[II. Review of Related Literature and Systems 10](#_Toc503230036)

[**2.1** **Traditional Car Rental Systems/Apps** 11](#_Toc503230037)

[**2.1.1** **Hertz Car Rental** 11](#_Toc503230038)

[**2.1.2** **Viking Car Rental** 13](#_Toc503230039)

[**2.2** **Similar System in the Philippines** 15](#_Toc503230040)

[III. Technical Background 15](#_Toc503230041)

[**3.1** **Android** 16](#_Toc503230042)

[***3.2*** **Integrated Development Environment: *Android Studio*** 16](#_Toc503230043)

[**3.3** **Java** 17](#_Toc503230044)

[**3.4** **Database: *SQLite*** 17](#_Toc503230045)

[**3.5** **Geolocation** 18](#_Toc503230046)

[IV. Methodology, Results and Discussion 19](#_Toc503230047)

[**4.1** **Requirements Analysis** 19](#_Toc503230048)

[**4.2** **Requirements Documentation** 20](#_Toc503230049)

[**4.3** **Software Requirements Specification** 28](#_Toc503230050)

[**4.4** **Design of Software, Systems, Product, and/or Processes** 31](#_Toc503230051)

[**4.4.1** **Functional Decomposition Diagram** 31](#_Toc503230052)

[**4.4.2** **Context Diagram and Data Flow Diagram** 32](#_Toc503230053)

[**4.4.3** **Entity Relationship Diagram** 34](#_Toc503230054)

[**Data Dictionary** 35](#_Toc503230055)

[**4.4.4** **Class Diagram** 39](#_Toc503230056)

[**4.4.5** **Object Diagram** 41](#_Toc503230057)

[**4.4.6** **Event Table** 42](#_Toc503230058)

[**4.4.7** **Use Case Diagram** 43](#_Toc503230059)

[**4.4.8** **Use Case Full Description** 44](#_Toc503230060)

[**4.4.9** **Activity Diagram** 50](#_Toc503230061)

[**4.4.10** **Sequence Diagram** 58](#_Toc503230062)

[**4.4.11** **State Diagram** 64](#_Toc503230063)

[**4.4.12** **Timing Diagram** 67](#_Toc503230067)

[**4.4.13** **Package Diagram** 70](#_Toc503230068)

[**4.4.14** **Communication Diagram** 71](#_Toc503230069)

[**4.4.15** **Component Diagram** 72](#_Toc503230070)

[**4.4.16** **Deployment Diagram** 73](#_Toc503230071)

[**4.4.17** **Composite Structure Diagram** 74](#_Toc503230072)

[**4.4.18** **Interaction Overview Diagram** 75](#_Toc503230073)

[Bibliography 76](#_Toc503230074)

[Appendix A 77](#_Toc503230075)

[Appendix B 78](#_Toc503230076)

# Introduction

## **Project Context**

Currently, social media, word of mouth, flyers and/or posters are the common means of how DIY (Do-It-Yourself) travelers, small production teams and families acquire vehicle rental services. This principle also applies to people who want their vehicles to be rented, all of this can be said in the grey economy. Finding rentable vehicles takes a lot of time according to the result of the survey conducted by the proponents. In order to make it easier for people to find rentable vehicles, the proponents proposed to develop an Android-based mobile application called *Carkila* that will provide a venue for users –vehicle owners and renters – to interact with each other.

This mobile application will allow users –vehicle owners and renters – to either list a vehicle for rent or rent one. This application will allow renters to look for a rentable vehicle according to their preferences like location, price range, seating capacity, etc. There will also be a geolocation feature that will help them find rentable vehicles near their location. Another feature will allow renters to make an offer or negotiate the renting price of the vehicle. There will also be a private chat feature where they can further discuss the important details of their transaction.

Thus, if everything goes well, this mobile application will alleviate the concerns of the target users that were found on the survey conducted.

## **Purpose and Description**

People look for rentable vehicles manually. A survey was conducted by the group with 50 respondents and the information that the group gathered from the survey shows that there is a limitation as to where people can find vehicles for rent; the majority being from referrals from other parties or by searching through social media. Also based on the data gathered, 44% of the respondents described that finding rentable vehicles was quite on the difficult side.

Another question on the survey asked the respondents if having a mobile app would be useful for finding private rentable vehicles and 98% of the respondents answered yes. Now with the data that the survey provided, the proponents have deduced that a mobile app would indeed be useful for people who need to find rentable vehicles.

Moreover, the group also conducted a survey for the drivers. Based the results of the survey, there is also a limitation as to where drivers can advertise their vehicles to get clients. Majority of the respondents (64 out of 80) answered that if they were to rent out their vehicle, it would be through social media. 70 out of 80 respondents thinks that an app would be useful for advertising their vehicles and getting more passengers. With the data, the proponents deduced that a mobile application would be helpful for both vehicle owners and renters.

*Carkila* is an Android-based mobile application that connects people who need to find rentable vehicles and people who want their cars to be rented. It includes a chat feature wherein both parties may offer a reasonable price. The owner can choose to accept the offer or decline it. This mobile application would also ensure that the transaction is secure for both end-users as both parties will have each other’s personal information.

There are already existing vehicle rental websites and mobile applications such as *Hertz, Viking Vehicle Rentals,* and *Manila Rent-A-Vehicle.* All these systems offer the same type of services; they have options like self-drive and chauffeured drive. The difference of these systems from the proposed project is that, they are all owned by a company; meaning they own fleet vehicles, unlike in *Carkila* where the vehicles are owned by local and private vehicle owners. Table 1 shows the difference between TNVS (Transport Network Vehicle Services) apps and the proposed project. Table 2 on the other hand, shows the difference between existing car rental systems and *Carkila.*

|  |  |  |  |
| --- | --- | --- | --- |
| **Features** | **Grab** | **Uber** | **Carkila** |
| Out-of-town transport | ✕ | ✕ | ✓ |
| Long-term rental | ✕ | ✕ | ✓ |

Table 1: Carkila vs. TNVS apps

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Features** | **Viking** | **Hertz** | **Manila Rent-a Car** | **Avis** | **Carkila** |
| Mobile-based | ✕ | ✓ | ✕ | ✓ | ✓ |
| Peer-to-Peer | ✕ | ✕ | ✕ | ✕ | ✓ |
| Offers self-drive | ✓ | ✓ | ✓ | ✓ | ✓ |
| Offers chauffeured drive | ✓ | ✓ | ✓ | ✓ | ✓ |
| Shows driver information | ✕ | ✕ | ✕ | ✕ | ✓ |
| Requires users to upload valid IDs | ✕ | ✕ | ✕ | ✕ | ✓ |
| Negotiable price | ✕ | ✕ | ✕ | ✕ | ✓ |
| Locates nearest vehicle available | ✕ | ✕ | ✕ | ✕ | ✓ |

Table 2: Carkila vs. Existing Car Rental Systems

## **Objectives**

**General Objectives:**

* To be able to develop an app that will provide a venue for the users – vehicle owners and car renters – to interact with each other and discuss necessary terms in renting vehicles

**Specific Objectives:**

* To automate the transactions of the users which includes the details of the trip, owner, and vehicle.
* To be able to implement a private chat feature where the owner and renter can discuss their terms for the transaction
* To be able to give users a list of available rentable vehicles that includes the owner and vehicle information, and images that is based on their specifications such as vehicle type, location, price range, and, availability (search filter).

## **Scope and Limitations**

The project is a vehicle rental mobile application that will run in Android. The scope of the mobile application is from the registration up until a transaction has been made between end-users. For vehicle owners, they are required to submit legal documents like their driver’s license details, NBI clearance, the vehicles’ Official Receipt (OR) and Certificate of Registration (CR), and insurance policy of the vehicle. Both users should upload a copy of their valid government ID. Background checks will also be done for every driver and for additional security and all the information will be open for law enforcement and government to view. For the renters, they are required to register their name, e-mail address, and contact number. They can also register through their social media accounts such as Facebook and Google. The renters must verify their accounts via SMS code verification.

The app will record transactions done by the vehicle owner and renter, and the cost of each transaction may be negotiated via the chat feature prior to accepting or declining the rental request. The app will not also be involved in doing necessary background checks when self-drive option is chosen; vehicle owners must be aware of the risk of renting a vehicle. Therefore, they must comply with what is stated in the terms and conditions that will be provided, since the main goal of the app is to provide an environment for vehicle owners and renters who need to find rentable vehicles in order to make it easier to find rentable vehicles.

# Review of Related Literature and Systems

The proposed project is a kind of consumer-to-consumer rental mobile application where both end-users are meant to benefit from each other. In order to determine the business process and techniques of car renting, the proponents researched about two different traditional car rental that exist today like Hertz and Viking Car Rental. Each system has different policies, techniques and features. This will help the proponents have a better understanding about car rental policies and therefore develop a more reliable and safe system also.

With these research, the proponents were also able to determine differences between the existing car rental systems and the proposed project. Firstly, Carkila will locate the nearest available vehicle to the user; therefore, making it more accessible and easier for users to get a vehicle wherever they are. In Hertz and Viking, there is a limitation to its branches. In Hertz, there are only 7 available pickup location; 5 which are airports, while the other two are offices located in Cebu and Makati only. Second, in Viking Car Rental, the process is still manual as you would still have to contact them for the rates. In Carkila, the rental rates are displayed in the app and the vehicles can be rented instantly. Also, the driver information can also be checked upon renting a vehicle. Uploading of valid IDs in the app will also be required for the renters for security purposes.

## **Traditional Car Rental Systems/Apps**

## **Hertz Car Rental**

Hertz car rental system provides customers a list of available cars that can be rented. These cars are from the company itself and not from other customers. What makes Hertz Car Rental System effective is also their own rental qualifications and requirements that should be taken into consideration before anything else to make more secured and efficient transactions. Here are some of their important qualifications and requirements:

* Filters – When reserving a car for rent, users are prompted to select/input filters such as pickup locations, pickup date and time, return date and time, age, car type, and then asks if a user is a guest or a member.
* Driver’s License – In order for a customer to rent a car, he/she should have a driver’s license and is valid for the duration of the desired rental period. The driver’s license should not expire on the range of the period. Moreover, if the customer’s driver’s license is registered in a foreign country, but wants to rent a car for example, in United States, the customer may need to issue an International Driver’s Permit to be qualified in processing the rental request.
* Insurances – Accidents may happen during the renting period. Therefore, different sets of insurances are also viable. For example, Personal Accident Insurance (PAI) can be applied when an accidental death and accidental medical expense happens. For an accidental death, the immediate family receives $175,000, while each passenger receives $17,500. For an accidental medical expense, the renter receives $2,500 and each passenger receives $2500. There are also a lot of additional insurances that can be applied depending on every country and the customer affected as well.
* Age restrictions and exceptions – the minimum age for a customer to be a candidate for the services of Hertz is 18. Of course the driver’s license requirement is also applied. However, there are also some things to be noted such as the type of car to be rented by the customer. For the Adrenaline, Dream and Prestige Collections, the minimum age is 25. All other cars can be rented for age lower than 25. Also, corporate accounts also state that young renters can also be allowed to rent the 3 collections for age 25. However, agreements should be made between the two parties.

Now, taking all these sample rental qualifications and requirements, the proponents can adapt these things in making policies for the application. For example, filters. Filters are recommended, so that the system knows what to process from the vast amount of data in itself. Filters help so it will be easier for the system and for the customers to find their desired car. The most important thing of all requirement is the driver’s license. In developing policies in the app, driver’s license should be the most significant requirement for a user to be registered in the system.

Complexities such as International Driver’s Permit should also be a requirement if a renter is foreign. This will ensure that the user is capable of driving even in foreign locations. In making features and policies in the system, insurances are a must also. And because it deals with cars and trips, insurances should be accounted by the system.

Lastly age restrictions should also be studied very well. Not all young renters can drive all types of vehicles. There should be some type of vehicles that should be restricted for some certain reasons just like in Hertz Car Rental System (Hertz, n.d.).

## **Viking Car Rental**

Viking Car Rental is another existing company that lends cars to the public. It from different kind of cars up to vans, and even buses. Viking Car Rental System also offers some of the same features of Hertz Car Rental System. However, Viking offers some features that Hertz do not, such as rental on buses and vans, tips and tricks on renting, detailed car specifications, and many more. Viking Car Rental System directly stated options that users can pick according to their own preferences:

* Self-Drive – Viking gives the costumers an option of self-drive meaning, they rent and drive the rented car. They will also be the ones to submit the car in their location after usage.
* Chauffeured Drive – The customer rents the vehicle, but is accompanied by a driver. This option is more expensive than self-drive. It’ll be safe for the customers because the drivers are trained and licensed by the company.
* Leasing/Long-Term rental – Viking recommend this option for users who still cannot decide whether to buy a certain vehicle or not. This can help them decide. Also, corporate use belongs here.
* Special Events: Conventions, Summit, Trade Shows etc. – Users can avail bigger discount or special rates if large number of vehicles are rented for special occasions.
* One-way Rental Pick Up or Drop Off – shares the same method with taxis, Uber, and Grab
* Promos – Viking Car Rental System also offers marketing promotions which the users can avail.
* Detailed Specifications of Available Cars – All available cars are rendered in a user friendly interface, where users can see the detailed description of vehicles such as number of doors, number of persons that can fit in the car, number of luggage that can fit in the car, and if the car is air conditioned or not. A picture of the car is also included (Viking Rent-A-Car, n.d.).

## **Similar System in the Philippines**

In order to confirm the need for a mobile app, the proponents researched about peer-to-peer car rental in the Philippines and they have only found one similar system called *Arkila.ph.* According to the owner, *Arkila.ph* removes the inconvenience of searching the internet and rentable vehicles and then contacting the drivers if they are available. With this information, the researchers concluded that the proposed project would indeed be useful.

In this website, a user has to post the trip details in the website and then the system will automatically send an SMS notification to all drivers that are registered in the system. After posting, a bidding process will come next, wherein the drivers will post the rates of their service. The user can choose among the bids posted by the drivers by clicking on the *Book* button (Arkila, n.d.).

# Technical Background

The proponents researched about (1) *Android*, where the app will be based on. The proponents plan to develop the project by using the (2) *Android Studio IDE* – which would be the main tool for the project – and by using the (3) *Java* programming language. (4) *SQLite* is also the proposed database to be used in developing the app. Furthermore, the group also research about (5) Geolocation to have a better understanding on how it works since there would be a geolocation feature in the proposed project.

## **Android**

Android is an operating system for mobile phones developed by Google. This platform’s kernel is based on the Linux operating system. It is designed specifically for touchscreen devices such as smartphones and tablet. Since it is designed for touchscreens, it is manipulated directly through touch gestures like swiping, tapping, etc. The Android OS was written using Java and C/C++ programming languages.

Android is an important part of the proposed project for this is where the proposed mobile application will be based. It is ideal to use Android as it is the most used and preferred OS in the Philippines. Research shows that 91% of Filipinos prefer the Android mobile operating system (Peebles, 2013).

## **Integrated Development Environment: *Android Studio***

An IDE (Integrated Development Environment) is a software that provides programmers basic tools that they need for developing a software such as a text editor, compiler, and debugger which can be accessed in a single GUI (Graphical User Interface). In simple terms, it is a software that allows developers to write and test/run their code. IDE is very helpful for programmers because all the tools necessary for development is just in a single software.

Android Studio is the official IDE (Integrated Development Environment) developed by Google for developing Android-based mobile application. Android Studio is based on the *IntelliJ IDEA* software – an IDE for Java. Android Studio supports Google App Engine, which can be used to integrate APIs and features (David, 2015). It is available on Windows, macOS, and Linux operating systems for free. Android Studio is ideal to use for the proposed project since the mobile application would be Android-based.

## **Java**

Java is a general-purpose and high-level programming language created by James Gosling. It was first released by Sun Microsystems in 1995. The compiled code – which is also called bytecode – can run on Windows, Linux, and macOS. According to techopedia.com, it can produce software for different platforms as long as it supports Java. Java is an object-oriented programming language that is based on the syntax of C and C++ programming languages.

Java would be the ideal programming language to use for the proposed project since it can run on any platforms, meaning that it can also run on Android. Moreover, the Android OS is written in Java and C++. Therefore, most mobile applications for Android are often developed using the Java programming language. Also, based on research, the proponents found out that it was one of the most used programming language for Android development.

## **Database: *SQLite***

A database is a collection of related records or data. The Database Management System (DBMS) is the software that allows users to interact with the databases. Basically, this is where one can create, update, or simply manage the databases.

An example of a database that can be used for the proposed project is *SQLite*. It is a relational database management system (DBMS) that can be embedded into an end program. It was also said to be the popular mobile database as it can also be used in Android and iOS. Database would play a huge part in the proposed project as this is where all the data would be kept, data which are all relevant such as information about the users and many more.

## **Geolocation**

According to techopedia.com, geolocation is the process of finding, determining and providing the location of a computer or any networking device. Basically, geolocation finds the real-world location of an Internet-connected device with the help of geolocation data. Geolocation data are any information that can be found from electronic devices that could be used to identify its physical location. There are two-types of geolocation data collection: (1) *Device-based data collection* relies mostly on GPS and cellular networks (McCarthy, 2017). It acquires any data from an application that a user has on his/her electronic device; and (2) *Server-based data collection* gathers any data connected to a device’s IP or MAC address either through Wi-Fi or Ethernet connection.

Geolocation would be a part of the proposed project for there would be a geolocation feature in the mobile application that will allow users to find nearby cars.

# Methodology, Results and Discussion

## **Requirements Analysis**

Vehicle rental or “arkila” is known to be one of the common means of transportation especially for out-of-town or vacation trips here in the Philippines. The proponents went to Facebook and observed that many people are posting about car rentals, specifically about where to find one or who knows anyone who offers car rentals. With this observation, the proponents thought that it is a problem as there is no mobile application for it yet.

The proponents researched about vehicle rentals here in the Philippines and they have found an existing website called *Arkila.ph,* which addresses the same issue that the proponents would like to target. The proponents considered this as a proof that there is indeed a problem regarding vehicle rentals.

In order to confirm the need for the mobile application, the proponents conducted an online survey that consisted of (4) four questions. The survey (*See Appendix A*) was posted in travel groups on social media and it was also answered by those who have experience in renting a vehicle. Moreover, the proponents also conducted surveys to ask for the opinions of drivers. The survey (*See Appendix B*) was also posted in car rental groups on social media and it was also given out to drivers as anyone who owns a car can be a potential user of the mobile application.

## **Requirements Documentation**

The following data support the proponents’ claim about the problems that the proposed application might mitigate. The respondents usually look for rentable vehicles through social media or through other people. Therefore, the mobile application will serve as a platform that will connect renters to vehicle owners who rents out their cars. According to the respondents, finding rentable vehicles can take up to more than an hour. This will be resolved by the mobile application for it will provide the users a list or choices of rentable vehicles. The proponents were also able to find a system/website that addresses the same problem that the group claims. Furthermore, the proponents also conducted a survey for potential car owners who may want to rent out their car in the future. This is to determine whether an app could also be useful for the drivers and also to determine whether to opt for a variable pricing or put bidding system feature in the app. Based on the results, the majority of the respondents are open to negotiation of prices and therefore, a bidding system could be used in the mobile application.

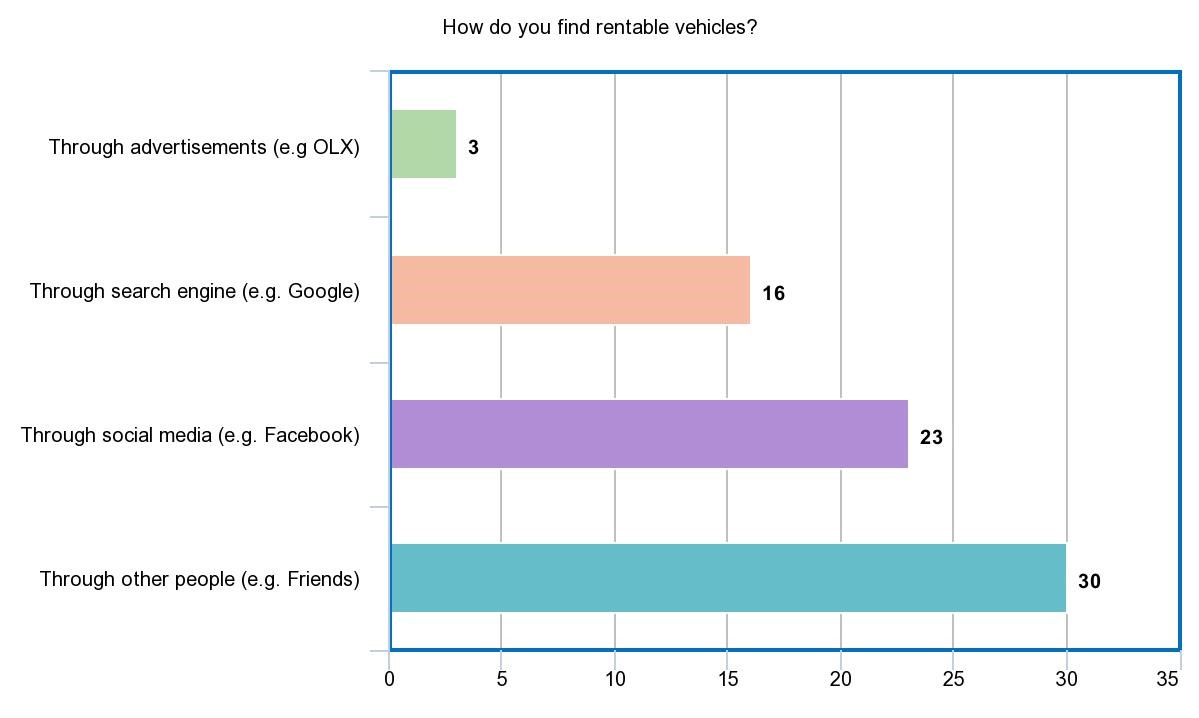


Figure 1: Source of rentable vehicles

Figure 1 shows that most of the respondents (30 out 50) search for rentable vehicles through other people. With this data, the proponents can foresee that not one of them had used a mobile application that is specifically for renting vehicles.

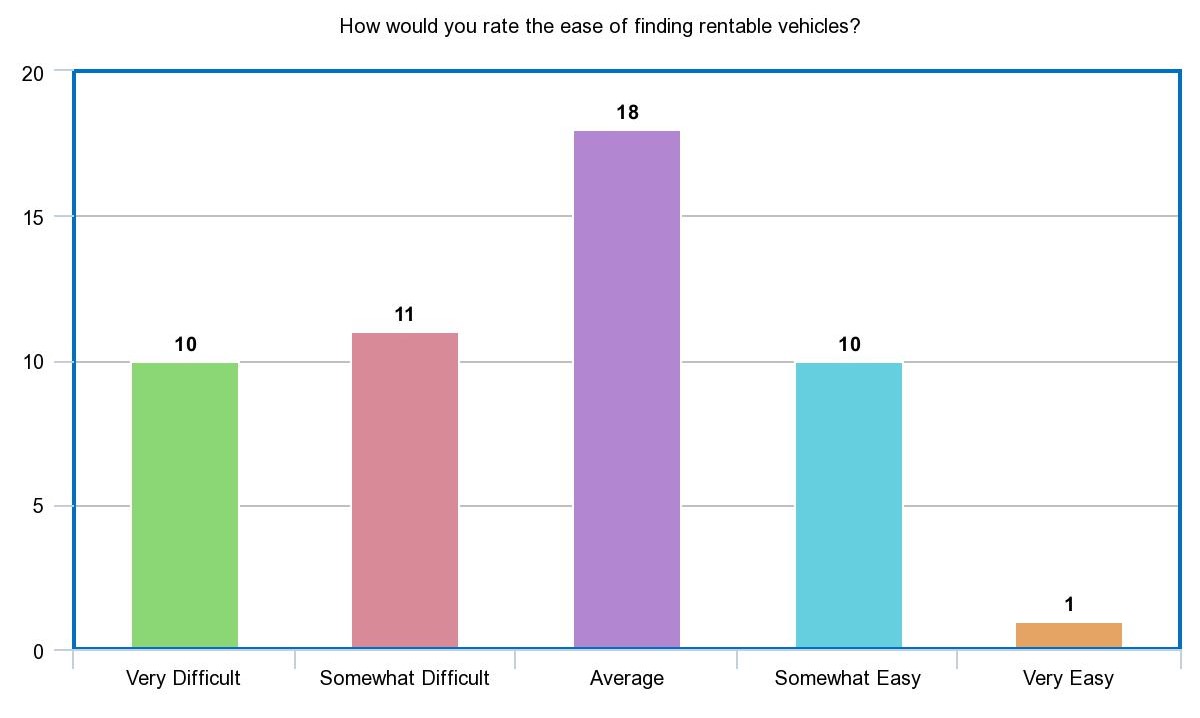


Figure 2: Difficulty of finding rentable vehicles

The proponents want to create a vehicle rental mobile application to make it easier for the people to look for rentable vehicles. Figure 2 shows that 36% (18 out 50) of the respondents rate the ease of finding rentable vehicles as average. Although it got the highest number of respondents, the majority of the result (21 out of 50) was quite on the difficult side. This data confirmed that there is indeed a difficulty when finding rentable vehicles.

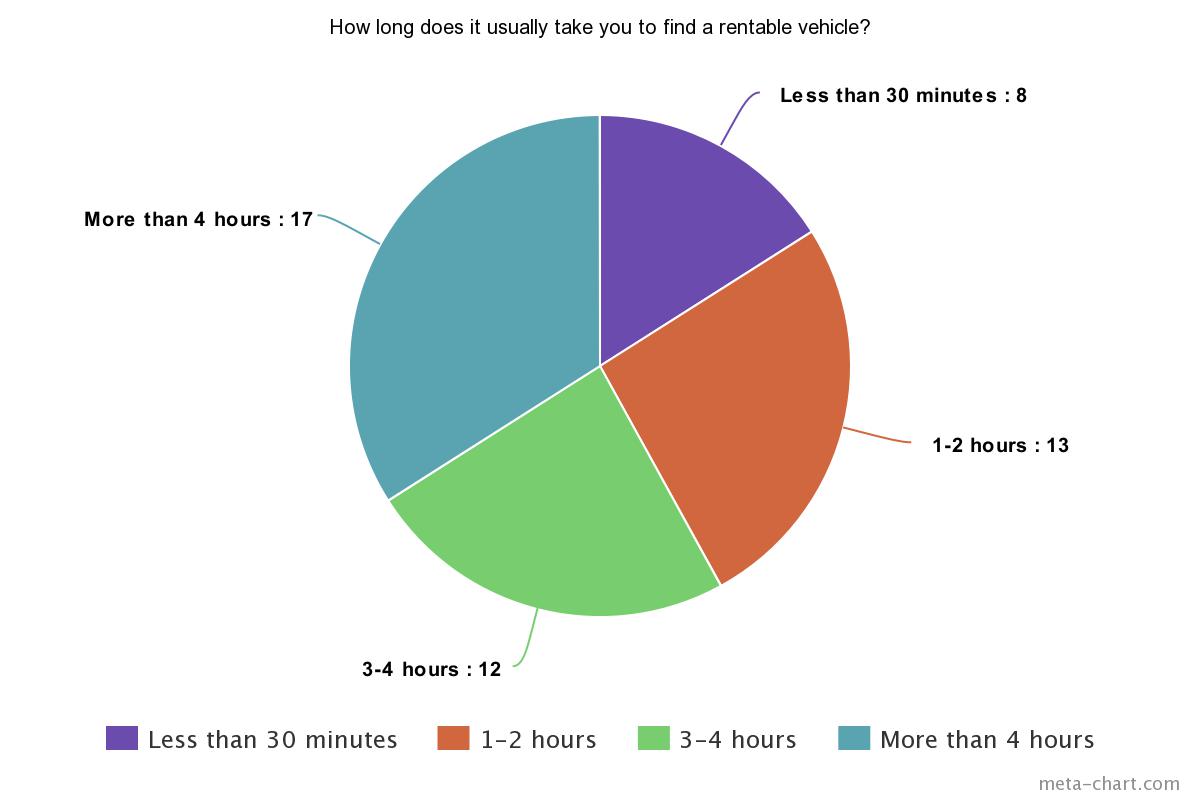


Figure 3: Time it takes to find rentable vehicles

Figure 3 shows the usual time it takes when finding rentable vehicles. Based on the results, 84% (42 out of 50) of the respondents took couple of hours to find rentable vehicles, and only 16% of them claim to have found a rentable vehicle in less than 30 minutes. As shown above, 34% (17 out of 50) respondents claim that it took them more than 4 hours to find a rentable vehicle. This is another issue that the proponents want to mitigate. The proponents would like to make it faster and easier for the people to find rentable vehicles with the help of a mobile application.

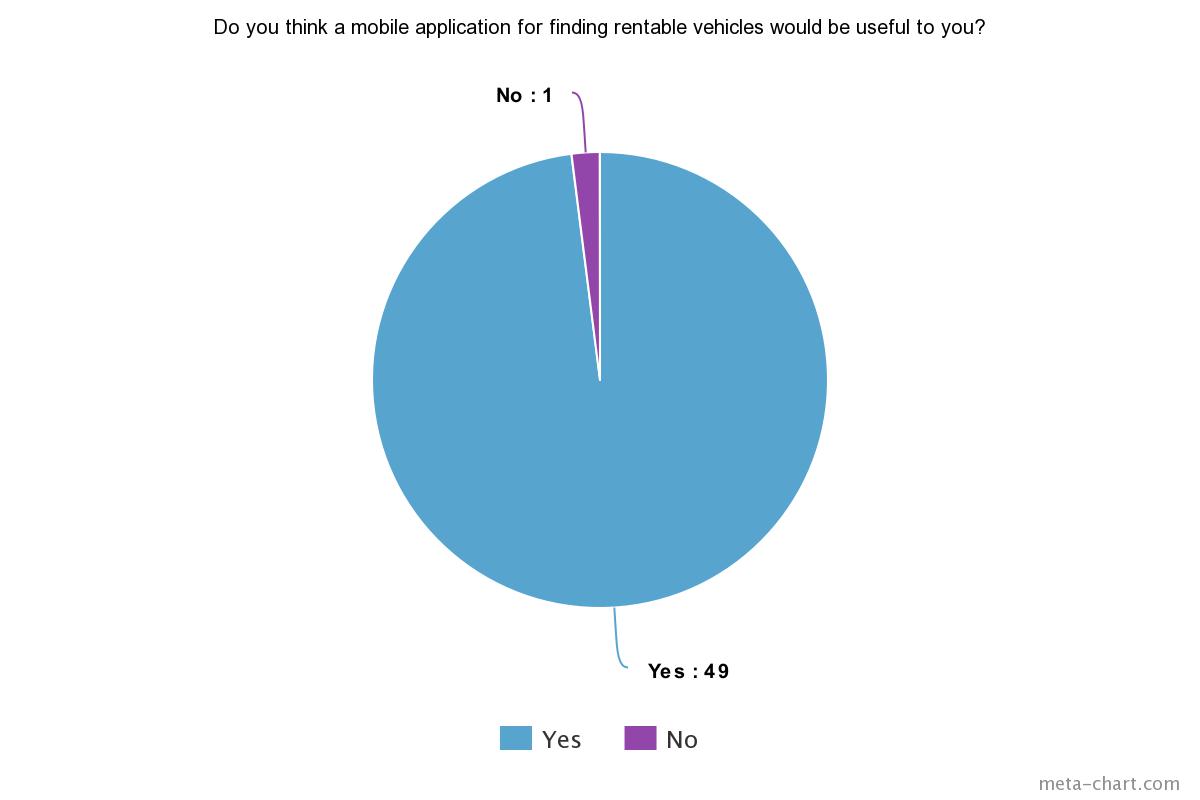


Figure 4: Usefulness of the proposed project for renters

The proponents asked if a mobile application would be useful. Based on the results, 98% thinks it would indeed be useful. Therefore, with this data, the proponents can foresee that a mobile application would indeed aid the problem.

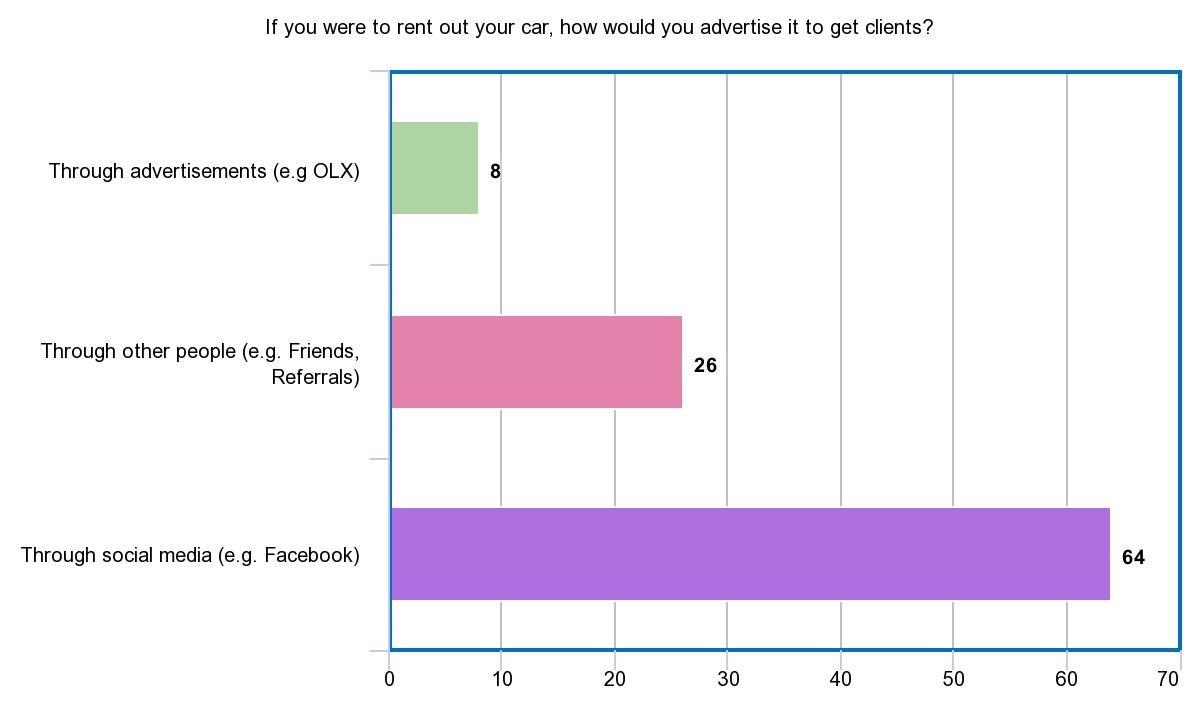


Figure 5: Ways of advertising rentable vehicles

The drivers were asked how they would advertise their cars if ever they were to rent it out. Figure 5 shows that 80% (64 out of 80) of the respondents consider using social media to advertise their cars. 32.5% (26 out of 80) the respondents considered advertising through other people. Meanwhile, 10% (8 out of 80) considered using advertisements such as OLX, flyers, etc.

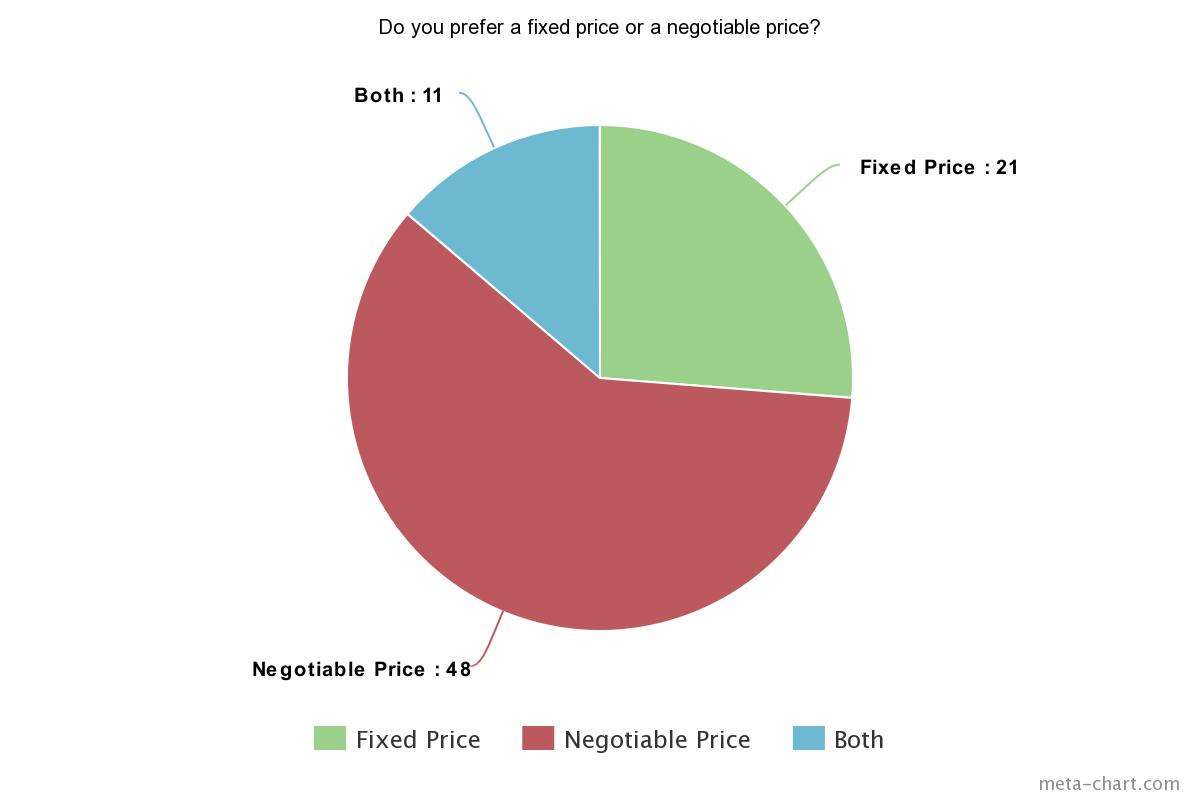


Figure 6: Pricing preference in renting vehicles

The proponents would like to determine whether car owners would agree with a bidding system wherein car renters could make offers or negotiate the renting price of a vehicle. Therefore, the proponents asked whether they prefer a fixed price or a negotiable price. Figure 6 shows that 60% (48 out of 80) of the respondents prefer a negotiable price, while 26.3% (21 out of 80) of them prefer a fixed price. Meanwhile, 13.7% (11 out of 80) of the respondents are okay with both. Based on the data gathered, the proponents concluded that majority of the respondents are open to negotiation of prices. Thus, it would be ideal to apply a bidding system wherein the users can negotiate the renting price of a vehicle.

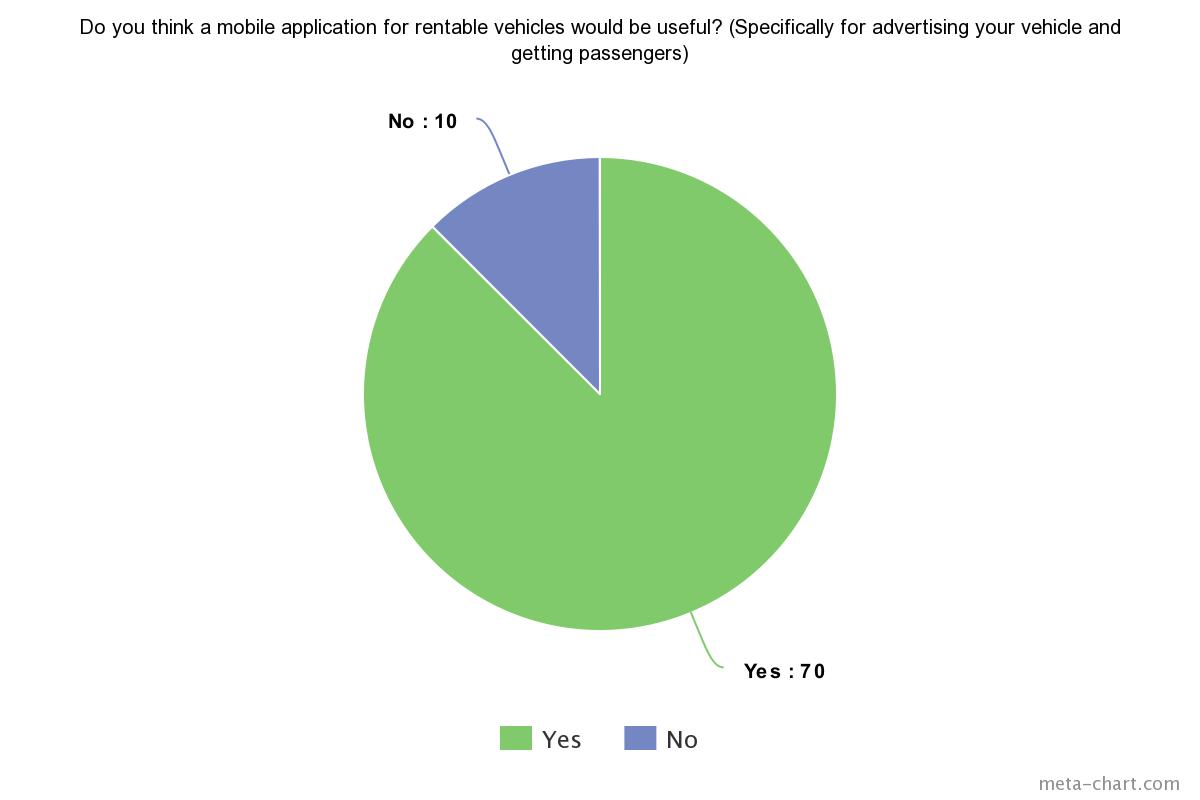


Figure 7: Usefulness of proposed project for vehicle owners

The proponents also asked if a mobile application could also be useful for advertising their vehicles 87.5% (70 out of 80) respondents answered yes while 12.5% (10 out of 80) answered no. Therefore, the proponents concluded that a mobile would also be indeed useful for the car owners.

## **Software Requirements Specification**

The following table shows the features and functionalities expected from the system.

Table 3: SRS Output

|  |
| --- |
| **OUTPUT** |
| * The mobile application must provide a report on the following: frequently requested vehicle, days of the week when vehicle is commonly requested. * The mobile app must generate a review on the renter and on the car owner with a review and rating from 0 to 5 stars. * The mobile application will provide a detailed list of the vehicle owner, the year and make of the vehicle, a recommended vehicle. * The app will display only vehicles that are available within the rent date/time specified by the car renter. * The mobile application must show the car details including: * images * make * model * year * seating capacity * color * The app must show the contact information of the car owner and the car renter. * The application must display to the car owner the chosen rent date and time to the car renter. * The application must display to the vehicle owner the chosen location of the renter. |

|  |
| --- |
| **INPUT** |
| * Car owners and car renters can sign up/sign in in the app by using their email or Facebook login credentials. * Car owners and car renters must input their personal information: * Full name * Address * Sex * Age * Birth date * Mobile Number * Email * License number (if driver) * Car owners must input their car details including: * Images of the vehicle (front, back, both sides, left, and right) * Brand * Year and model * Manufacturers recommended seating capacity * Color * Plate number * OR and CR * License * Car owners must set their availability schedule. * Car owners must upload the necessary documents via a clear picture or a clear scan. These documents include: * Driver’s License * The Official Receipt/Certification of Registration of the vehicle * Car renters must include their trip details: * Pick-up location * Destination * Time (from-to) * Date (from-to) * Number of Passengers |

Table 4: SRS Input

|  |
| --- |
| **PROCESS** |
| * The system must be able to store all vehicle information. * The system must be able to record the transactions. * The system must be able to generate reports. * The system must be able to locate nearest cars to the user. * The system must be able to find cars based on the preferences of the car renters. |

Table 5: SRS Process

Table 6: SRS Performance

|  |
| --- |
| **PERFORMANCE** |
| * The mobile application must be available 24/7. * The system must notify the owner if someone requests his/her vehicle. * The system must notify the renter if the owner accepted his/her request. |

Table 7: SRS Control

|  |
| --- |
| **CONTROL** |
| * Only an administrator can generate reports. * Only an admin can access the transaction records. |

## **Design of Software, Systems, Product, and/or Processes**

## **Functional Decomposition Diagram**

Figure 8 shows the broken-down processes and functions of the proposed system.

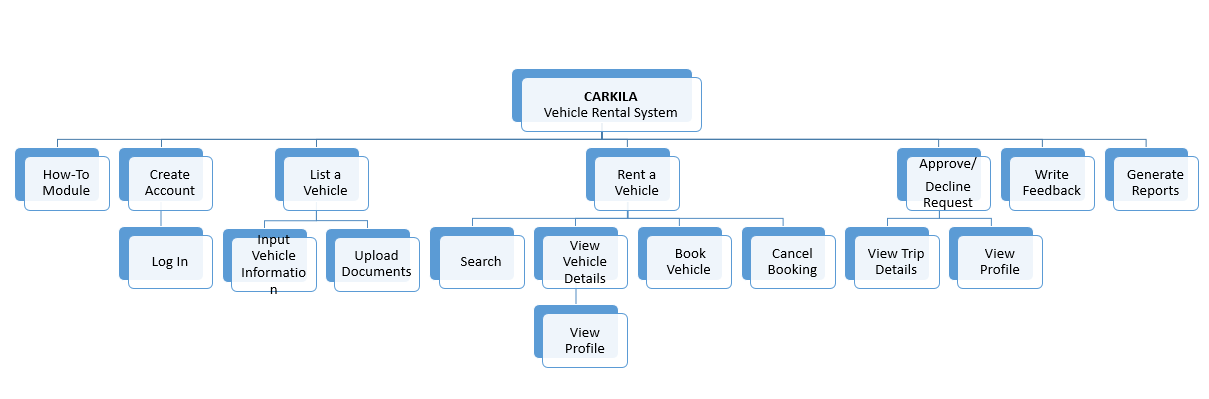


Figure 8: Functional Decomposition Diagram

## **Context Diagram and Data Flow Diagram**

Figure 9 and 10 shows the graphical representation of the proposed project without going over into detail of how the system works, but instead, it focuses on how all the data/information will move through the system.

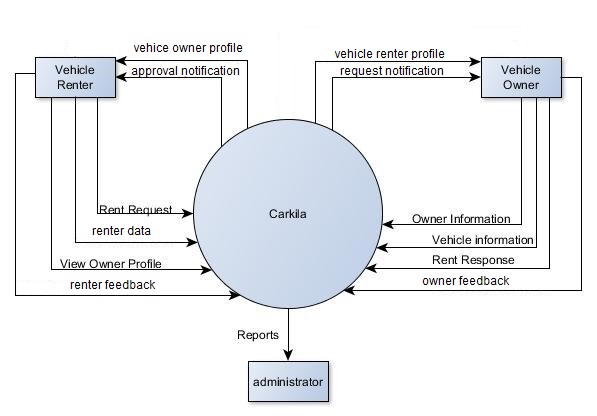


Figure 9: Context Diagram

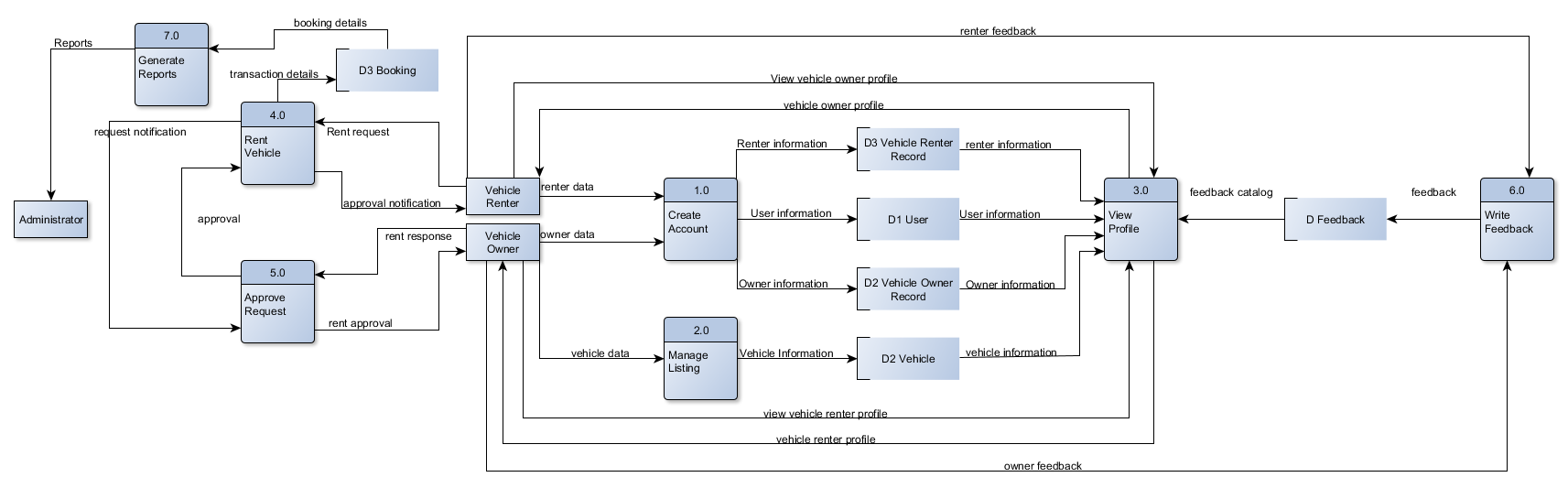


Figure 10: Diagram 0

## **Entity Relationship Diagram**

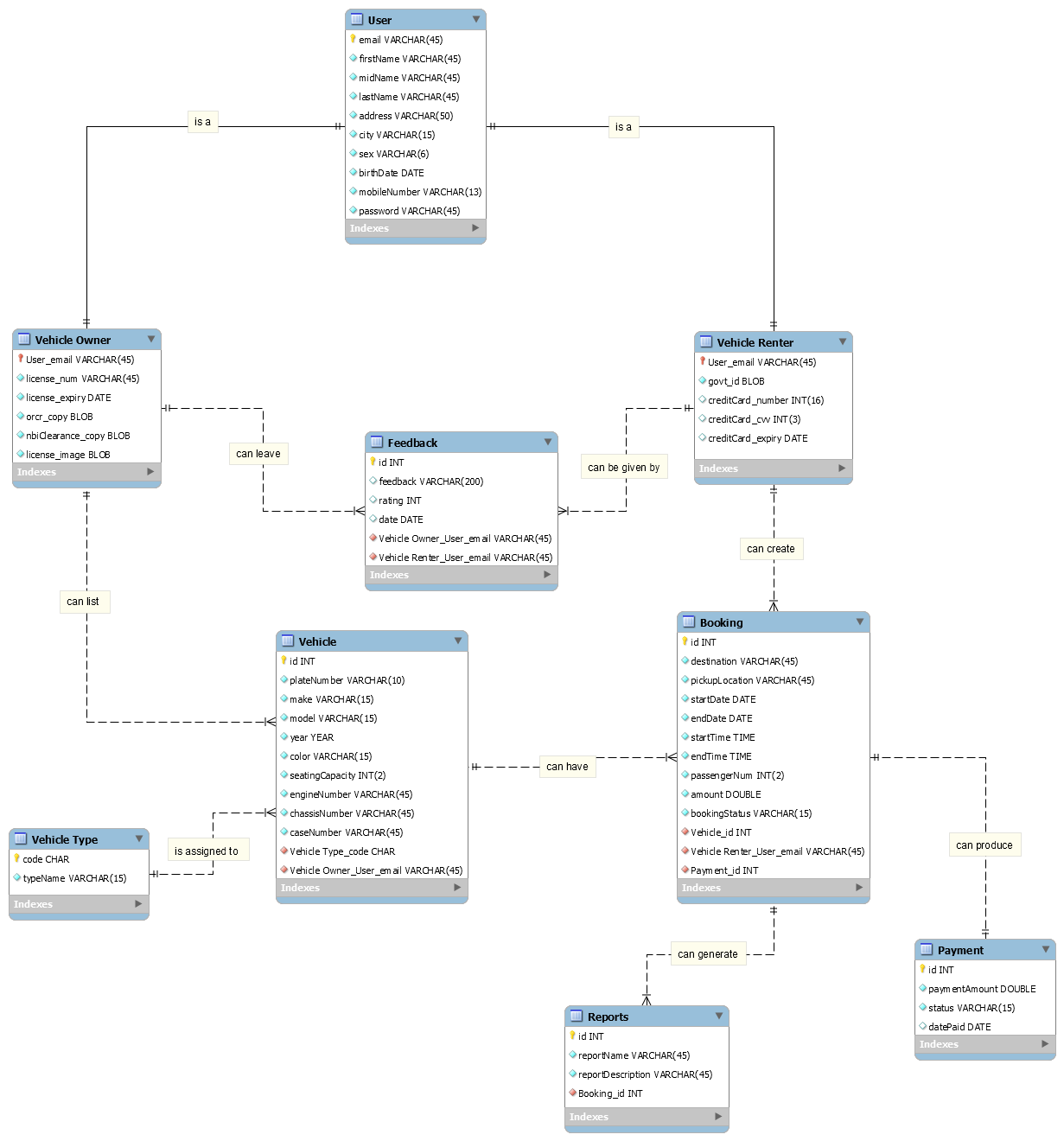
Figure 11 shows the proposed database design of the system. It shows the relationship between entities and where all the incoming data will be stored.

Figure 11: Entity Relationship Diagram

### **Data Dictionary**

The following tables show the description of each data items that will be placed in the database.

Table 8: Data Dictionary of User Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE: **User** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| email | INT |  | Primary Key | Email Address |
| firstName | VARCHAR | 45 | Not Null | First Name |
| midName | VARCHAR | 45 | Not Null | Middle Name |
| lastName | VARCHAR | 45 | Not Null | Last Name |
| address | VARCHAR | 45 | Not Null | Address |
| city | VARCHAR | 2 | Not Null | City |
| sex | VARCHAR | 6 | Not Null | Sex |
| age | INT | 2 | Not Null | Age |
| birthdate | DATE |  | Not Null | Birth Date |
| mobileNumber | VARCHAR | 13 | Not Null | Mobile Number |
| password | VARCHAR | 45 | Not Null | Password |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE: **Vehicle Owner** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| User\_email | VARCHAR | 45 | Primary Key, Foreign Key | Email Address |
| license\_num | VARCHAR | 45 | Not Null | License Number |
| license\_expiry | DATE |  | Not Null | License Expiry |
| orcr\_copy | BLOB |  | Not Null | Photocopy of OR/CR |
| nbiClearance\_copy | BLOB |  | Not Null | Photocopy of NBI Clearance |
| license\_copy | BLOB |  | Not Null | Photocopy of License |

Table 9: Data Dictionary of Vehicle Owner Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE: **Vehicle Renter** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| User\_email | VARCHAR | 45 | Primary Key, Foreign Key | Email Address |
| govt\_id | BLOB |  | Not Null | Valid Government ID |
| creditCard\_number | INT | 16 | Null | Credit Card Number |
| creditCard\_cvv | INT | 3 | Null | Credit Card CVV |
| creditCard\_expiry | DATE |  | Null | Credit Card Expiry |

Table 10: Data Dictionary of Vehicle Renter Table

Table 11: Data Dictionary of the Vehicle Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE: **Vehicle** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| id | INT |  | Primary Key | Vehicle ID, Auto-generated |
| plateNum | VARCHAR | 45 | Not Null | Vehicle Plate Number |
| make | VARCHAR | 45 | Not Null | Vehicle Make |
| model | VARCHAR | 45 | Not Null | Vehicle Model |
| year | YEAR |  | Not Null | Vehicle Year |
| color | VARCHAR | 15 | Not Null | Vehicle Color |
| seatingCapacity | INT | 2 | Not Null | Vehicle Seating Capacity |
| engineNumber | VARCHAR | 45 | Not Null | Vehicle Engine Number |
| chassisNumber | VARCHAR | 45 | Not Null | Vehicle Chassis Number |
| Vehicle Type\_code | CHAR |  | Foreign Key | Vehicle Type ID |
| Vehicle Owner\_User\_email | VARCHAR | 45 | Foreign Key | Vehicle Owner E-mail Address |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE: **Vehicle Type** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| code | CHAR |  | Primary Key | Vehicle Type Code |
| typeName | VARCHAR | 15 | Not Null | Vehicle Type Name |

Table 12: Data Dictionary of Vehicle Type Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE: **Booking** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| id | INT | 11 | Primary Key | Booking ID, Auto-generated |
| destination | VARCHAR | 45 | Not Null | Destination |
| pickupLocation | VARCHAR | 45 | Not Null | Pickup Location |
| startDate | DATE | 45 | Not Null | Trip Start Date |
| endDate | DATE |  | Not Null | Trip End Date |
| startTime | TIME |  | Not Null | Trip Start Time |
| endTime | TIME |  | Not Null | Trip End Time |
| passengerNum | INT | 2 | Not Null | Number of Passengers |
| amount | DOUBLE | 11 | Not Null | Amount of Rental |
| bookingStatus | VARCHAR |  | Not Null | Booking Status |
| Vehicle\_id | INT |  | Foreign Key | Vehicle ID |
| Vehicle Renter\_User\_email | VARCHAR | 45 | Foreign Key | Vehicle Renter E-mail Address |
| Payment\_id | INT |  | Foreign Key | Payment ID |

Table 13: Data Dictionary of Booking Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE: **Payment** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| id | INT |  | Primary Key | Payment ID, Auto-generated |
| paymentAmount | DOUBLE |  | Not Null | Payment Amount |
| paymentStatus | VARCHAR | 15 | Not Null | Payment Status |
| datePaid | DATE |  | Null | Date Paid |

Table 14: Data Dictionary of Payment Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE: **Feedback** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| id | INT |  | Primary Key | Report ID, Auto-generated |
| feedback | VARCHAR | 200 | Null | Feedback |
| rating | INT |  | Null | Rating |
| date | DATE |  | Null | Feedback Date |
| Vehicle\_Owner\_User\_email | VARCHAR | 45 | Foreign Key | Vehicle Owner E-mail Address |
| Vehicle Renter\_User\_email | VARCHAR | 45 | Foreign Key | Vehicle Renter E-mail Address |

Table 15: Data Dictionary of Feedback Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE: **Reports** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| id | INT |  | Primary Key | Report ID, Auto-generated |
| reportName | VARCHAR | 45 | Not Null | Report Name |
| reportDescription | VARCHAR | 45 | Not Null | Report Description |
| Booking\_id | INT |  | Foreign Key | Booking ID |

Table 16: Data Dictionary of Reports Table

## **Class Diagram**

Figure 12 shows the classes, their attributes, methods, and their relationship with one another which will later be used in developing the system.

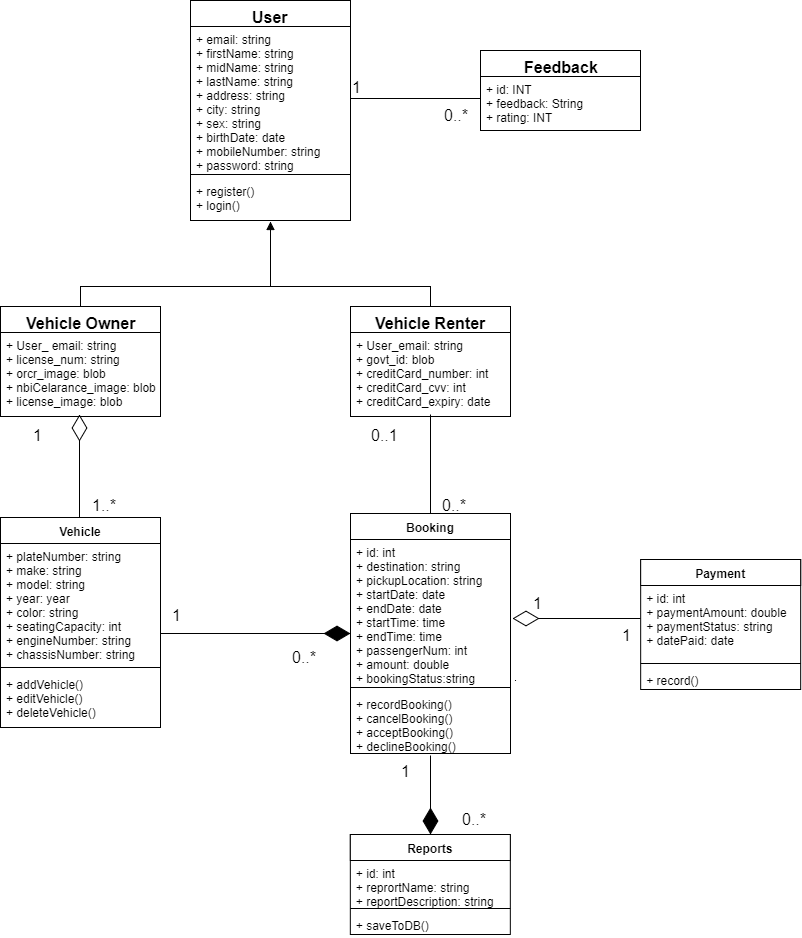


Figure 12: Class Diagram

## **Object Diagram**

Figure 13 shows the relationship between objects. This diagram will also help in understanding the relationship between class instances.

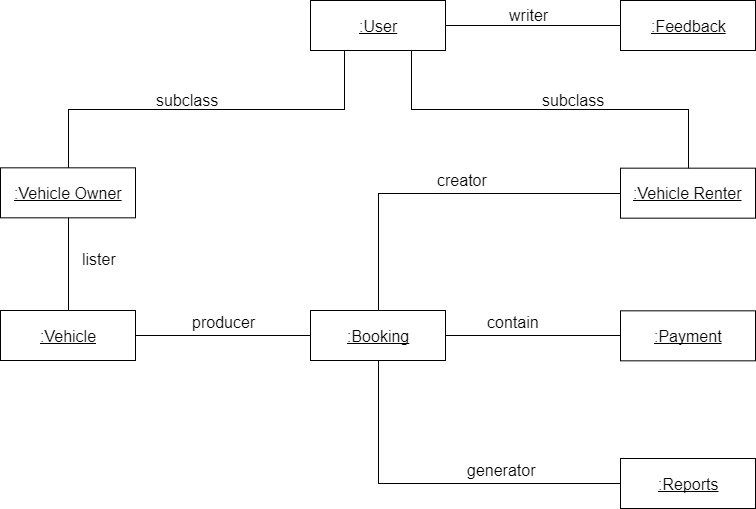


Figure 13: Object Diagram

## **Event Table**

Table 6 shows the events where the system must respond and how it is supposed to respond. This table will also help in identifying and creating use cases for the proposed project.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Event** | **Trigger** | **Source** | **Use Case** | **Response** | **Destination** |
| 1. Vehicle renter wants to create an account. | New account request | Vehicle renter | Create Account | New Account/  Account details | System  Vehicle renter |
| 1. Vehicle renter wants to rent a vehicle. | Request a vehicle | Vehicle renter | Rent Vehicle | Trip details | Vehicle owner |
| 1. Vehicle owner wants to create, view, update, or delete listing. | Request to manage vehicle listing | Vehicle owner | Manage Listing | Vehicle listing  details | System Vehicle owner |
| 1. Vehicle owner wants to accept a rent request. | Rent request notification | Vehicle owner | Approve Request | Approval | Vehicle renter |
| 1. Vehicle renter wants to view profile | Profile request | Vehicle renter | View Profile | Vehicle Owner Profile | Vehicle renter |
| 1. Vehicle renter wants to give feedback. | Request to give user feedback | Vehicle renter | Give Feedback | Feedback form | Vehicle renter |
| 1. Admin requests the system to generate reports. | Report request | Admin | Generate Reports | Report details | System  Admin |

Table 17: Event Table

## **Use Case Diagram**

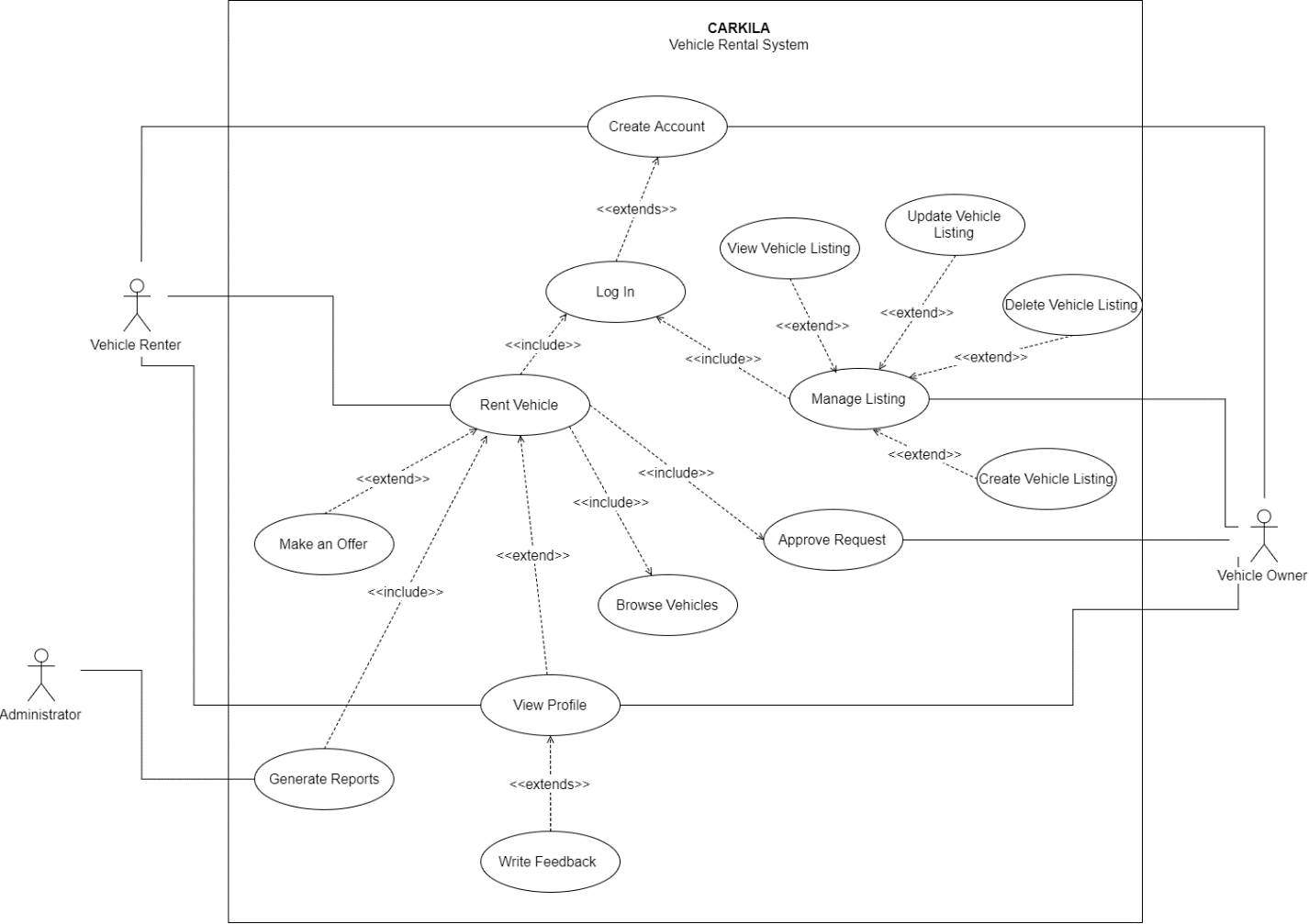
Figure 14 shows how the users can interact with the system. This diagram also contains the step-by-step activities that occurs in the system.

Figure 14: Use Case Diagram

## **Use Case Full Description**

The following tables shows the description of each use cases.

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Create Account | |
| **Scenario:** | User wants to create an account. | |
| **Triggering Event:** | New account request | |
| **Brief Description:** | This will allow the user to create an account. | |
| **Actors:** | Vehicle owner, vehicle renter | |
| **Related Use Case:** | Rent Vehicle, Manage Listing | |
| **Stakeholders:** | Vehicle owner, vehicle renter | |
| **Preconditions:** | The user must have the mobile application installed. | |
| **Post conditions:** | The user must be able to access the app features. | |
| **Flow of Activities:** | **Actor** | **System** |
| 1. The user opens the mobile application. 2. The user taps **Create Account**. 3. The user enters the required information. 4. The user enters the verification code. | * 1. The system displays the create account/log in page.   2. The system prompts the user to enter the required information.   3.1 The system validates the data entered.  3.2 The system sends a verification code.  3.3 The system displays the verification page.  4.1 The system directs the user to the home page. |
| **Exception Conditions:** | 1. If the user entered invalid data. 2. If the user entered invalid verification code. 3. If the user cancels the creation of account. | |

Table 18: Create Account Use Case Description

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Rent Vehicle | |
| **Scenario:** | Vehicle renter wants to rent a vehicle. | |
| **Triggering Event:** | Request a vehicle | |
| **Brief Description:** | This will allow the user to rent a vehicle. | |
| **Actors:** | Vehicle renter | |
| **Related Use Case:** | Create Account | |
| **Stakeholders:** | Vehicle renter | |
| **Preconditions:** | The vehicle renter must be logged in. | |
| **Post conditions:** | The vehicle must be able to book a vehicle.  A notification must be sent to the vehicle owner. | |
| **Assumptions:** | The vehicle renter is already logged in. | |
| **Flow of Activities:** | **Actor** | **System** |
| 1. The renter taps **Browse Vehicles.** 2. The renter clicks on a vehicle listing. 3. The renter clicks **Book Now**. 4. The renter enters the trip details. 5. The renter taps **Send Request**. | * 1. The system displays the list of vehicles.   2.1 The system displays the vehicle information page.  3.1 The system prompts the renter to enter trip details.  5.1 The system sends a rent request notification to the vehicle owner. |
| **Exception Conditions:** | 1. If required data is missing. 2. If booking is cancelled. | |

Table 19: Rent Vehicle Case Description

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Manage Listing | |
| **Scenario:** | Vehicle owner wants to create, view, update, or delete a listing. | |
| **Triggering Event:** | Request to add, edit, or delete a listing | |
| **Brief Description:** | This will allow the vehicle owner to create, view, update, or delete a vehicle listing. | |
| **Actors:** | Vehicle owner | |
| **Related Use Case:** | Create Account | |
| **Stakeholders:** | Vehicle owner | |
| **Preconditions:** | The vehicle owner must be logged in. | |
| **Post conditions:** | The listing must be published in the mobile application.  The vehicle owner must be able to create, view, update, or delete a listing. | |
| **Assumptions:** | The vehicle owner is already logged in. | |
| **Flow of Activities:** | **Actor** | **System** |
| 1. The vehicle owner taps onthe **Listing** tab**.**   2.A. The vehicle owner taps **Create Vehicle Listing.**  2.B. The vehicle owner enters all required information.  3. The vehicle owner views an existing listing.  4. The vehicle owner edits the vehicle listing  5. The vehicle owner deletes an existing listing. | * 1. The system displays the listing page.   2.A.1 The system prompts the user to enter required vehicle information and upload required documents.  2.B.1 The system will store the information and publish the vehicle listing in the app.  3.1 The system displays the vehicle listing information.  4.1 The system displays the vehicle listing in edit mode.  5.1 The system will remove the listing from the app. |
| **Exception Conditions:** | 1. If the user entered invalid data. 2. If there is no existing listing, user must create one first to view, update, or delete. | |

Table 20: Manage Listing Use Case Description

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Approve Request | |
| **Scenario:** | Vehicle owner wants to accept a rent request. | |
| **Triggering Event:** | Request to manage listing | |
| **Brief Description:** | This will allow the vehicle owner to accept a rent request. | |
| **Actors:** | Vehicle owner | |
| **Related Use Case:** | Create Account, Rent Vehicle | |
| **Stakeholders:** | Vehicle owner | |
| **Preconditions:** | The vehicle owner must be logged in.  The vehicle owner must receive a rent request notification. | |
| **Post conditions:** | The vehicle owner must be able to accept a request.  The vehicle renter must receive a rent request. | |
| **Assumptions:** | The vehicle owner is already logged in. | |
| **Flow of Activities:** | **Actor** | **System** |
| 1. The vehicle owner taps on **Notifications**. 2. The vehicle owner taps on a request. 3. The vehicle renter taps **Accept.** | * 1. The system displays the notification page.   2.1 The system displays the trip details sent by the renter.  3.1 The system notifies the renter that the rent request has been accepted. |
| **Exception Conditions:** | 1. If the owner declines the rent request. | |

Table 21: Approve Request Use Case Description

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | View Profile | |
| **Scenario:** | The vehicle renter wants to view the profile of the vehicle owner. | |
| **Triggering Event:** | Profile request | |
| **Brief Description:** | This will allow the vehicle renter to view the profile of the vehicle owner or vice versa. | |
| **Actors:** | Vehicle renter | |
| **Related Use Case:** | Rent a Vehicle | |
| **Stakeholders:** | Vehicle renter | |
| **Preconditions:** | The vehicle renter must select a vehicle listing. | |
| **Post conditions:** | The vehicle renter must be able to see the vehicle owner’s profile. | |
| **Assumptions:** | The vehicle renter is already logged in.  The vehicle renter is already on a vehicle listing page. | |
| **Flow of Activities:** | **Actor** | **System** |
| 1. The vehicle renter clicks on the icon of the vehicle owner. | 1.1 The system displays the vehicle information.  2.1 The system displays the profile of the owner. |
| **Exception Conditions:** | 2. If the vehicle renter wants to view the feedbacks on the owner, initiate *View Feedback* use case. | |

Table 22: View Profile Use Case Description

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Write Feedback | |
| **Scenario:** | The vehicle renter wants to give feedbacks on the owner. | |
| **Triggering Event:** | Request to give user feedback | |
| **Brief Description:** | This will allow the vehicle renter to give feedbacks on the owner. | |
| **Actors:** | Vehicle renter | |
| **Related Use Case:** | View Profile, Rent Vehicle | |
| **Stakeholders:** | Vehicle renter | |
| **Preconditions:** | The vehicle renter must select a vehicle listing. | |
| **Post conditions:** | The vehicle renter must be able to write feedback. | |
| **Assumptions:** | The vehicle renter is already on the profile of the vehicle owner. | |
| **Flow of Activities:** | **Actor** | **System** |
| 1. The vehicle renter wants to give feedback**.** 2. The renter will give feedback. | * 1. The system displays the feedback form.   2.1 The system will save the feedback. |
| **Exception Conditions:** | - | |

Table 23: Write Feedback Use Case Description

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Generate Reports | |
| **Scenario:** | The admin requests the system to generate reports. | |
| **Triggering Event:** | Report request | |
| **Brief Description:** | This will allow the administrator to generate a report. | |
| **Actors:** | Administrator | |
| **Related Use Case:** |  | |
| **Stakeholders:** | Administrator | |
| **Preconditions:** | There must be existing bookings. | |
| **Post conditions:** | A report must be created.  The report must be stored in the database. | |
| **Flow of Activities:** | **Actor** | **System** |
| 1. The admin wants to generate a report**.** 2. The admin receives the report. | * 1. The system retrieves data from the database.   2. The system analyzes the data.   3. The system displays the generated report.   4. The system stores the report in the database. |
| **Exception Conditions:** | 1. If there are no existing bookings. | |

Table 24: Generate Reports Use Case Description

## **Activity Diagram**

The following diagrams graphically represents the flow of activities of each use cases of the system.

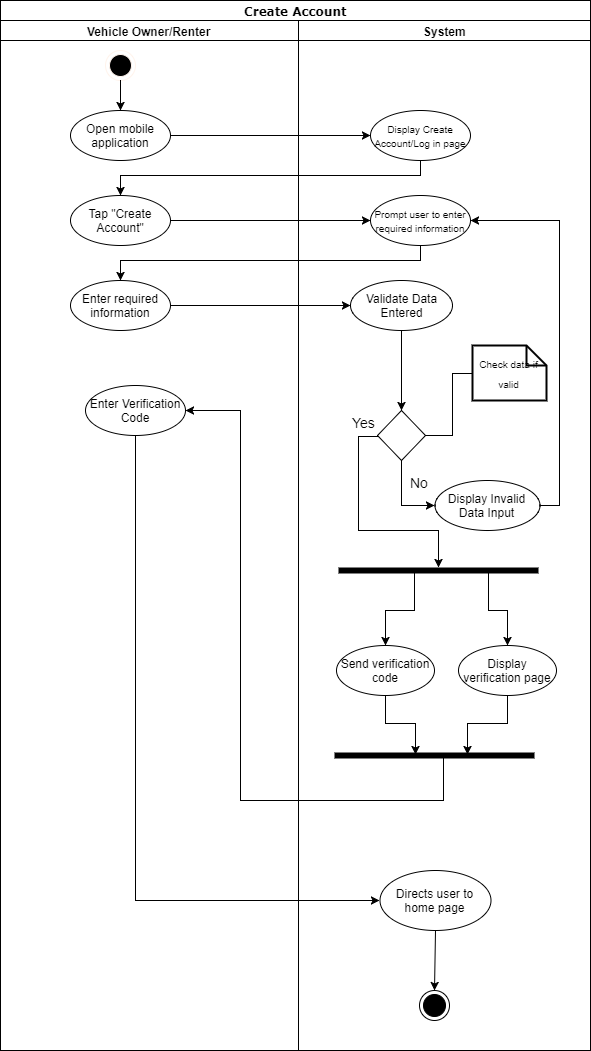
****

Figure 15: Create Account Activity Diagram

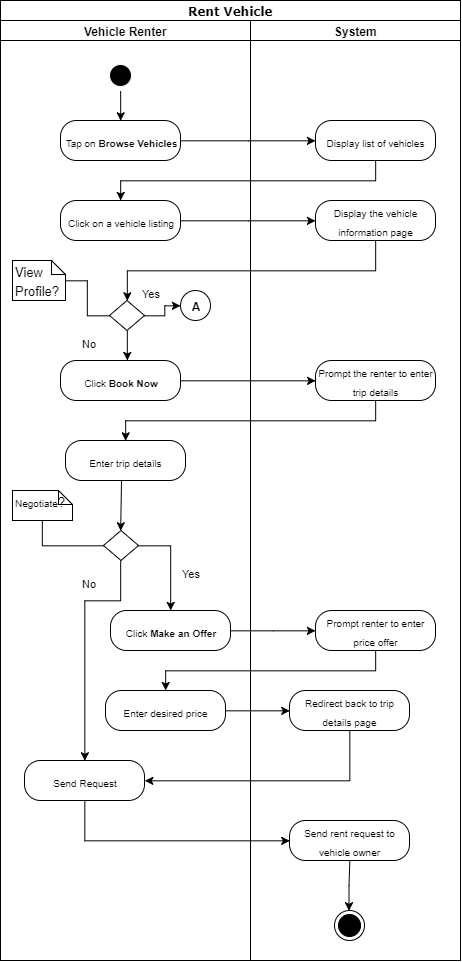
****

Figure 16: Rent Vehicle Activity Diagram

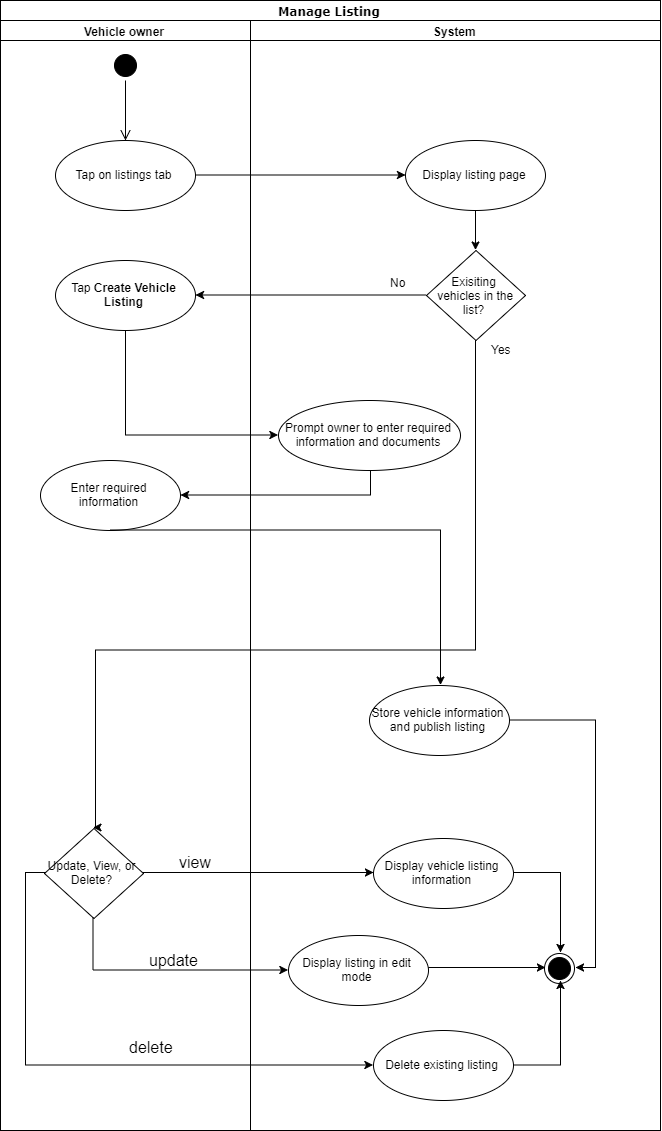


Figure 17: Manage Listing Activity Diagram

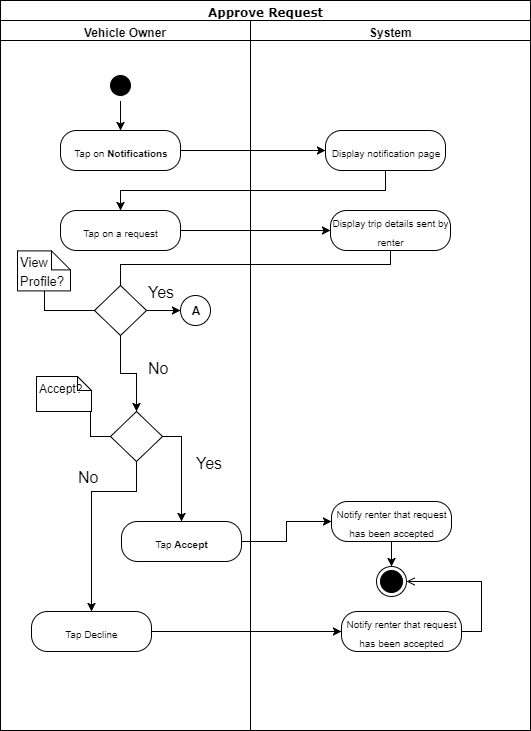
****

Figure 18: Approve Request Activity Diagram

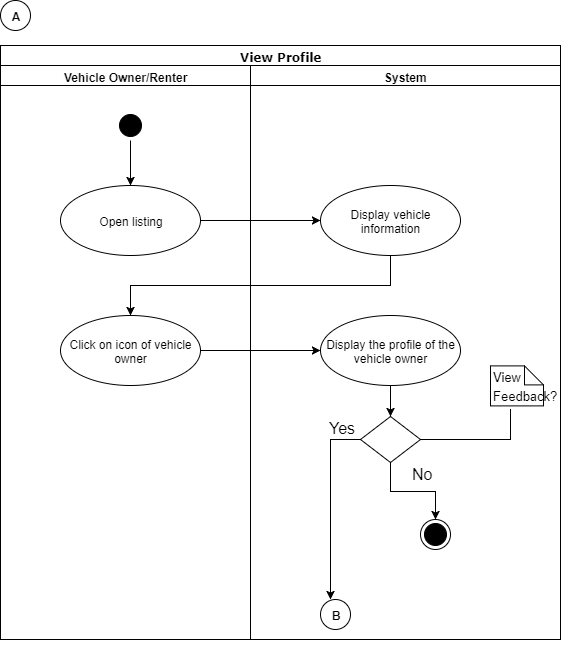
****

Figure 19: View Profile Activity Diagram

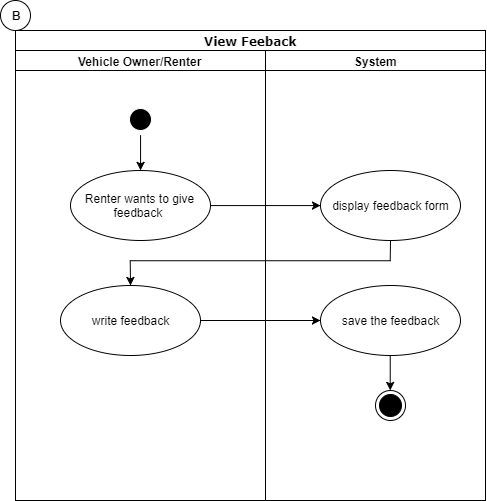
****

Figure 20: View Feedback Activity Diagram

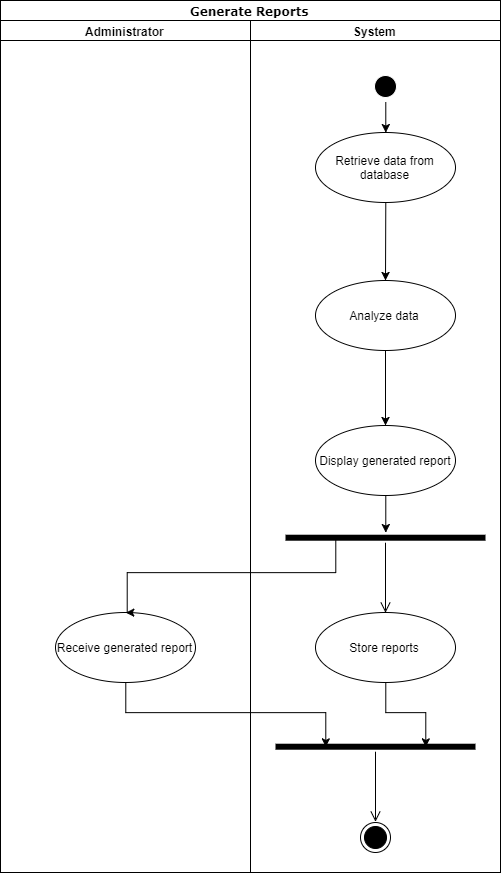
****

Figure 21: Generate Reports Activity Diagram

## **Sequence Diagram**

The following diagrams shows a graphical representation of how the objects of the system interact with each other

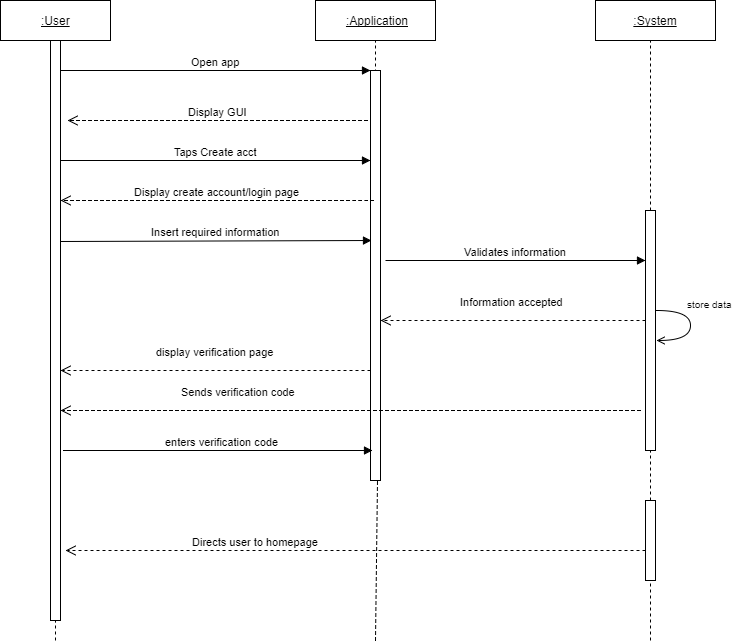
****

Figure 22: Create Account Sequence Diagram

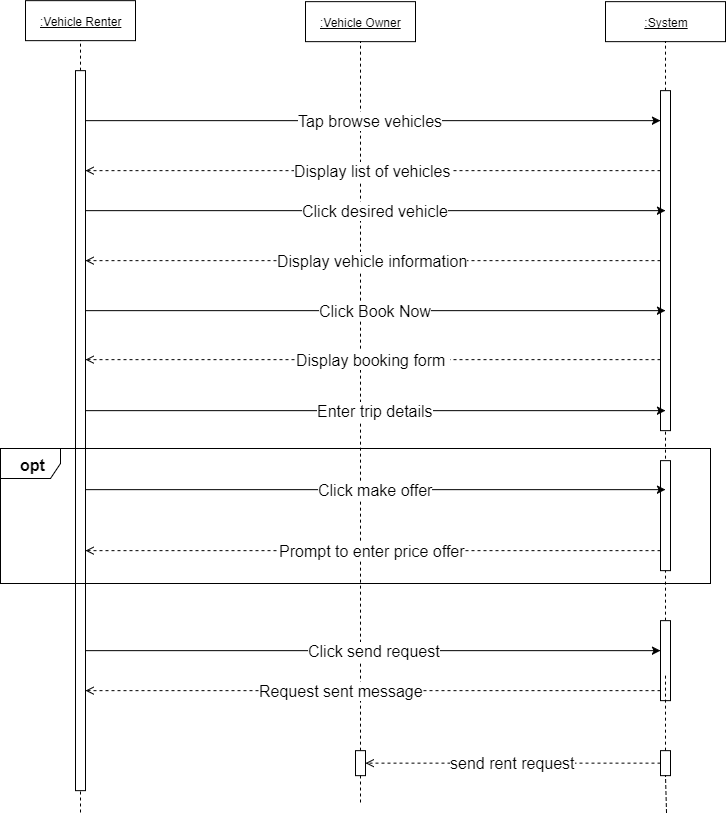


Figure 23: Rent Vehicle Sequence Diagram

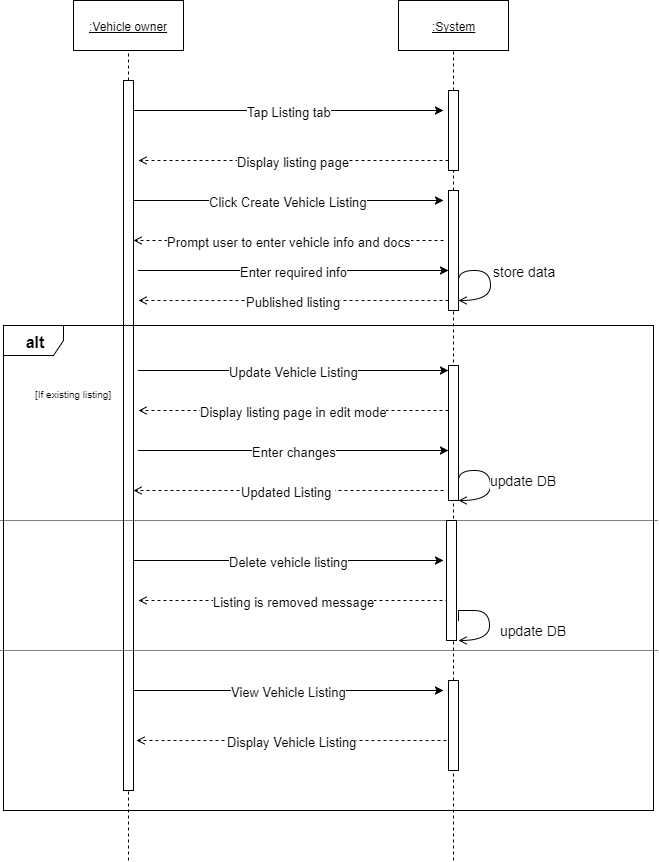


Figure 24: Manage Listing Sequence Diagram

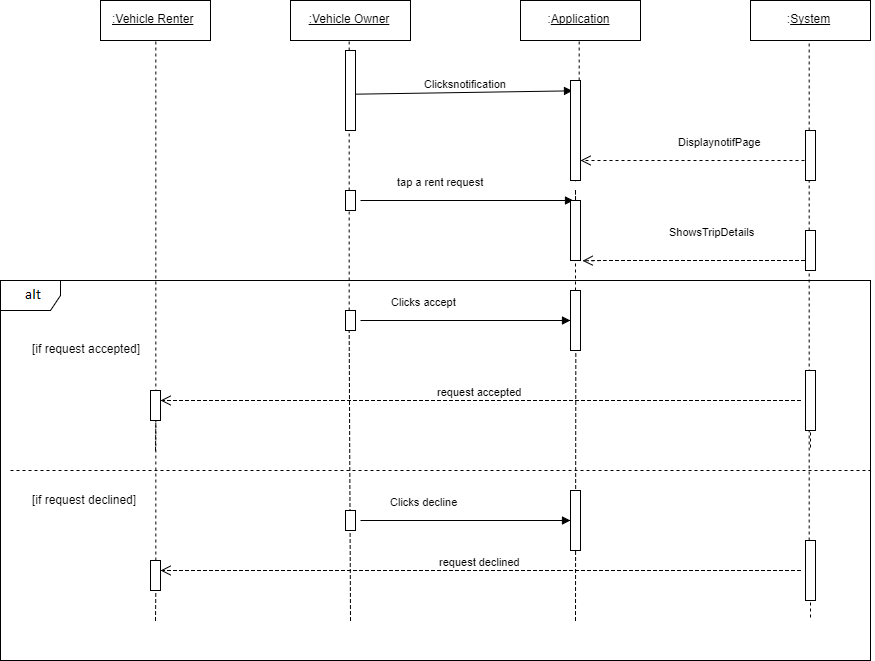


Figure 25: Approve Request Sequence Diagram

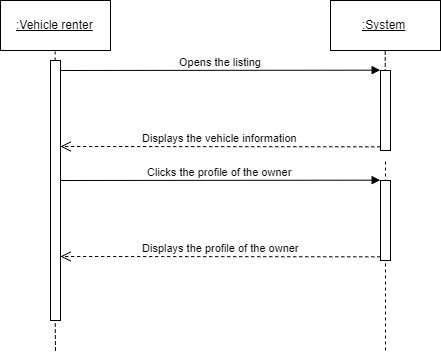


Figure 26: View Profile Sequence Diagram

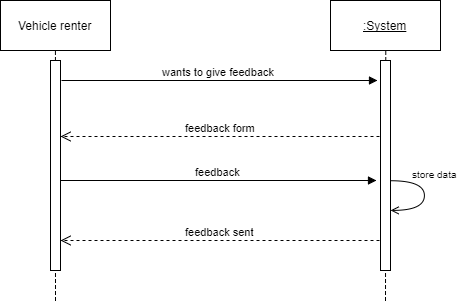


Figure 27: View Feedback Sequence Diagram

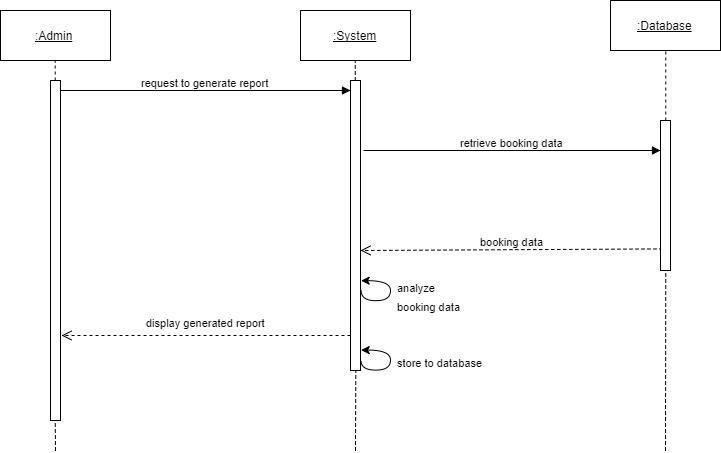


Figure 28: Generate Reports Sequence Diagram

## **State Diagram**

The following diagrams show the state or behavior of the system in response to the action executed.

### C:\Users\Martha\Downloads\SD-Create (1).png

Figure 29: State Diagram of Create Account Use Case

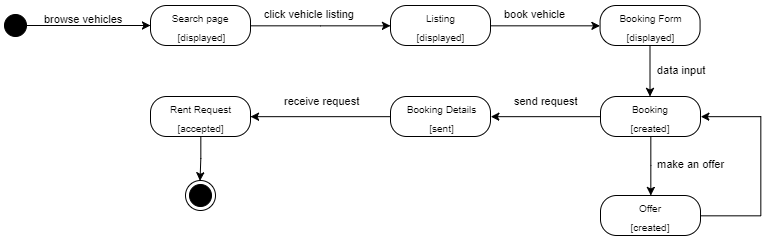


Figure 30: State Diagram of Rent Vehicle Use Case

### C:\Users\Martha\Downloads\SD-manage.png

Figure 31: State Diagram of Manage Listing Use Case

### C:\Users\Martha\Downloads\SD-approve.png

Figure 32: State Diagram of Approve Request Use Case

C:\Users\Martha\Downloads\SD-viewprof (1).png

C:\Users\Martha\Downloads\SD-writefeed.png

Figure 33: State Diagram of View Profile and Write Feedback Use Case

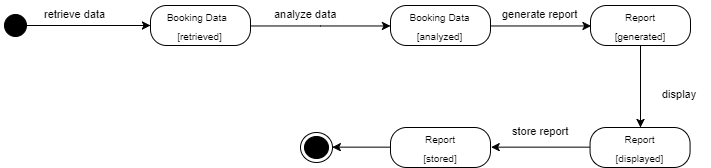


Figure 34: State Diagram of Generate Reports Use Case

## C:\Users\Martha\Downloads\TD-Create.UPDATED.png**Timing Diagram**

Figure 35: Timing Diagram of Create Account Use Case

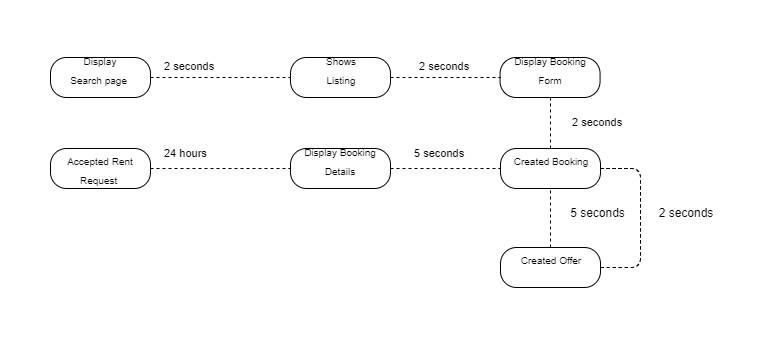


Figure 36: Timing Diagram of Rent Vehicle Use Case

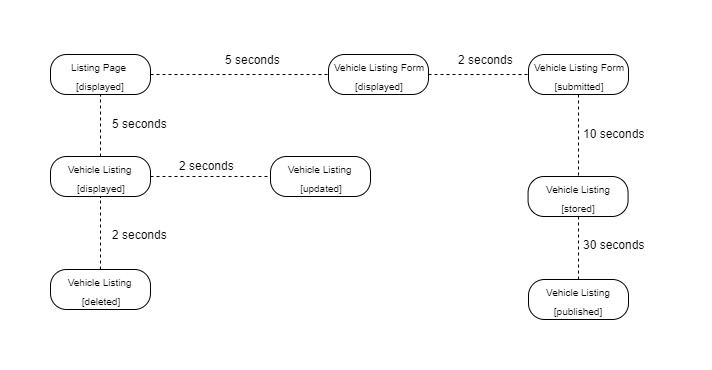


Figure 37: Timing Diagram of Manage Listing Use Case

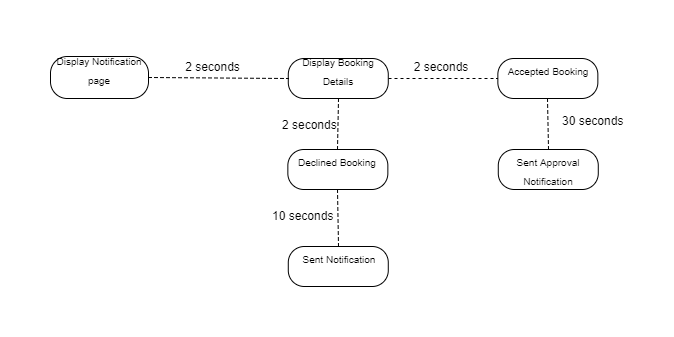
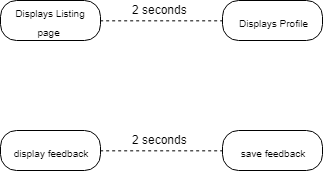


Figure 38: Timing Diagram of Approve Request Use Case

Figure 39: Timing Diagram of View Profile and Feedback Use Case



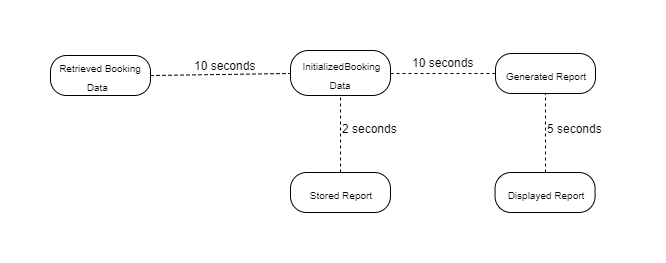


Figure 40: Timing Diagram of Generate Reports Use Case

## **Package Diagram**

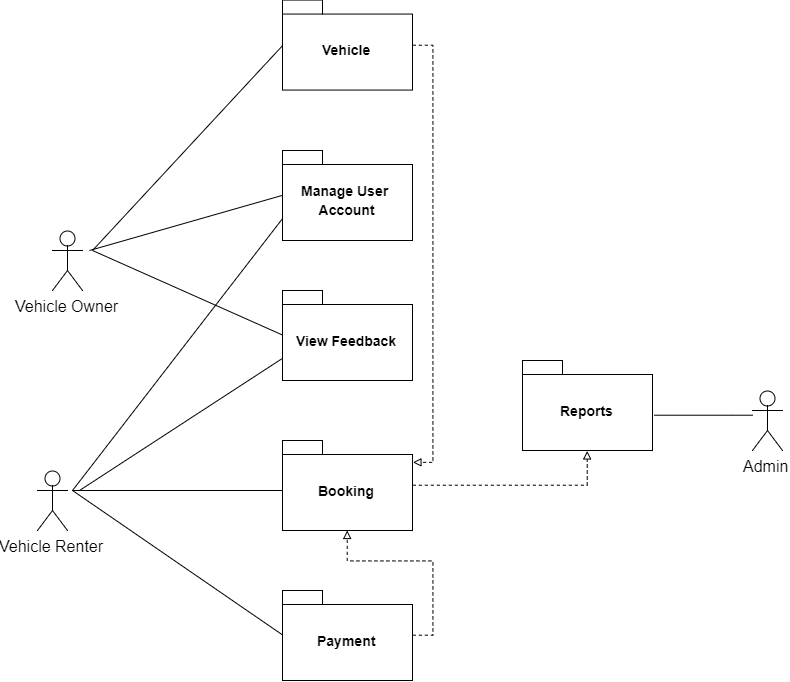


Figure 41: Package Diagram

## **Communication Diagram**

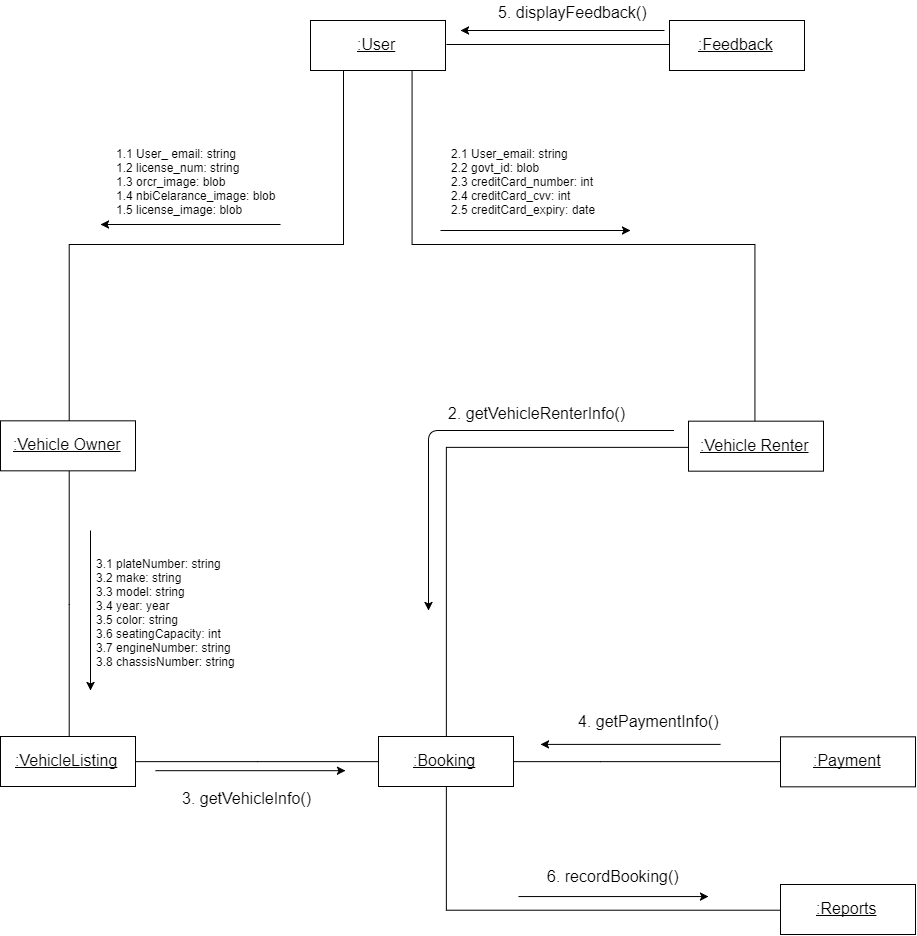
****

Figure 42: Communication Diagram

## **Component Diagram**

Figure 43: Component Diagram

## **Deployment Diagram**

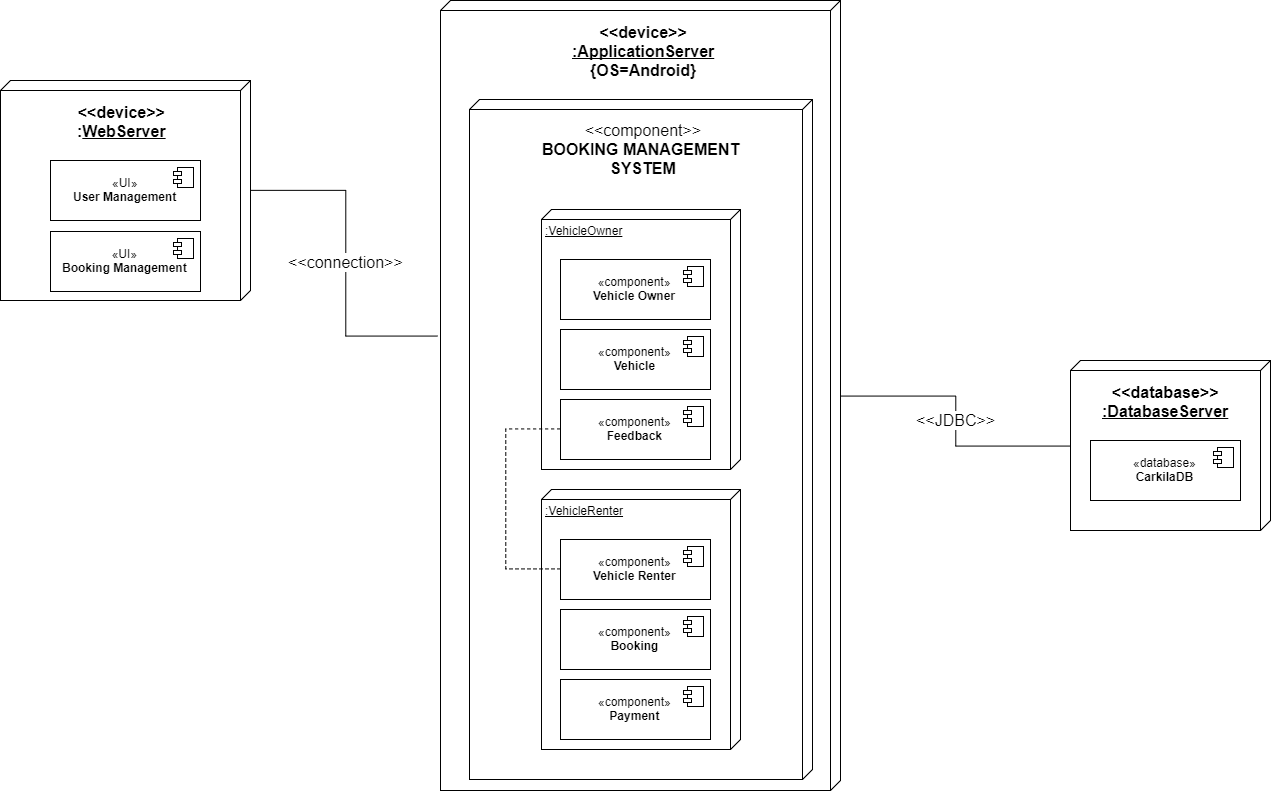
****

Figure 44: Deployment Diagram

## C:\Users\Martha\Downloads\CompositeStructure2 (1).png**Composite Structure Diagram**

Figure 45: Composite Structure Diagram

## C:\Users\Martha\Downloads\Interaction (2).png**Interaction Overview Diagram**

Figure 46: Interaction Overview Diagram

# Bibliography

*Arkila*. (n.d.). Retrieved from Arkila.ph: http://arkila.ph/

Arkila.ph. (2015, February 25). *Arkila.PH: The fastest and easiest way to rent a vehicle in the Philippines*. Retrieved from IdeaSquare: https://ideasquare.io/t/arkila-ph-the-fastest-and-easiest-way-to-rent-a-vehicle-in-the-philippines/83

Ballus-Armet, I., Shaheen, S. A., Clonts, K., & Weinzimmer, D. (2014). *Peer-to-Peer Carsharing*. Retrieved from http://www.carsharing.org/research/TSRC/P2Pcarsharing\_TRR2014.pdf

Covington, P., Adams, J., & Sagrin, E. (2016). Deep Neural Networks for YouTube Recommendations.

Deepak. (2015, September 25). In-depth Feature Analysis of P2P Car Renting Marketplace. *Takeover the Peer-to-Peer Car Sharing Marketplace with These Essential Website Features*. Retrieved from https://www.fatbit.com/fab/peer-to-peer-car-sharing-marketplace-features-business-model/

Dishman, L. (2010, June 21). *Brother, Can You Spare a Blender? The Scoop on Peer-to-Peer Equipment Rentals*. Retrieved from Fast Company: https://www.fastcompany.com/1662305/brother-can-you-spare-blender-scoop-peer-peer-equipment-rentals

Einav, L., Farronato, C., & Levin, J. (2015, August). *Peer-to-Peer Markets*. Retrieved from http://www.nber.org/papers/w21496.pdf

FATbit Technologies. (2015, July 29). Facilitating Car Renting-Sharing with Internet: Business Mdel & Site Feature. *Planning to Launch Car Renting Marketplace? Script Features to Succeed!* Retrieved from https://www.fatbit.com/fab/start-peer-to-peer-car-renting-portal-with-these-advanced-features/

Greenrobot. (n.d.). Mobile Databases. *SQLite and SQLite alternatives for Android and iOS*. Retrieved from http://greenrobot.org/news/mobile-databases-sqlite-alternatives-and-nosql-for-android-and-ios/

Hertz. (n.d.). Retrieved from https://www.hertz.com

*iRenta*. (n.d.). Retrieved from iRenta Philippines: https://www.irenta.net/

Java. (n.d.). *What is Java technology and why do I need it?* Retrieved from https://www.java.com/en/download/faq/whatis\_java.xml

McCarthy, A. (2017, March 1). *Geolocating Carmen Sandiego*. Retrieved from https://www.gravitatedesign.com/blog/what-is-geolocation/

Peebles, C. (2013, December 8). *Android Becomes Top Mobile Operating System in the Philippines*. Retrieved from http://mobileadvertisingwatch.com/android-becomes-top-mobile-operating-system-philippines-2246

SQLite. (n.d.). *Features of SQLite*. Retrieved from https://www.sqlite.org/features.html

Techopedia. (n.d.). *Geolocation*. Retrieved from https://www.techopedia.com/definition/1935/geolocation

Viking Rent-A-Car. (n.d.). Retrieved from http://vikingcarrentals.com.ph/

Wikipedia. (n.d.). SQLite. Retrieved from https://en.wikipedia.org/wiki/SQLite

# Appendix A

# Appendix B