# 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

SYSADD1 and 2

By

Martha Angela Salamat

Justin Jullian Omac

Brian Olores

John Ryan Pasaoa

November 7, 2017

# List of Figures

Figure 1 21

Figure 2 22

Figure 3 23

Figure 4 24

Figure 5 25

Figure 6 26

Figure 7 27

Table of Contents

[Carkila 1](#_Toc497790770)

[Vehicle Rental Mobile Application 1](#_Toc497790771)

[List of Figures 2](#_Toc497790772)

[I. Introduction 4](#_Toc497790773)

[**1.1** **Project Context** 4](#_Toc497790774)

[**1.2** **Purpose and Description** 5](#_Toc497790775)

[**1.3** **Objectives** 6](#_Toc497790776)

[**1.4** **Scope and Limitations** 7](#_Toc497790777)

[II. Review of Related Literature 8](#_Toc497790778)

[**2.1** **Traditional Car Rental Systems/Apps** 8](#_Toc497790779)

[**2.1.1** **Hertz Car Rental** 8](#_Toc497790780)

[**2.1.2** **Viking Car Rental** 10](#_Toc497790781)

[**2.2** **Peer-to-Peer Rentals** 12](#_Toc497790782)

[**2.3** **Similar Systems in the Philippines** 13](#_Toc497790783)

[**2.3.1** **Arkila.ph** 13](#_Toc497790784)

[**2.3.2** **iRenta.net** 13](#_Toc497790785)

[**2.4** **Artificial Intelligence in Recommendation Services** 14](#_Toc497790786)

[**2.5** **Artificial Intelligence in Matchmaking** 16](#_Toc497790787)

[III. Technical Background 16](#_Toc497790788)

[**3.1.1** **Android** 17](#_Toc497790789)

[**3.2** **Integrated Development Environment: *Android Studio*** 17](#_Toc497790790)

[**3.3** **Java** 18](#_Toc497790791)

[**3.4** **Database: *SQLite*** 19](#_Toc497790792)

[**3.5** **Geolocation** 19](#_Toc497790793)

[IV. Methodology, Results and Discussion 20](#_Toc497790794)

[**4.1** **Requirements Analysis** 20](#_Toc497790795)

[**4.2** **Requirements Documentation** 21](#_Toc497790796)

[**4.3** **Software Requirements Specification** 29](#_Toc497790797)

[**4.4** **Design of Software, Systems, Product, and/or Processes** 31](#_Toc497790798)

[**4.4.1** **Functional Decomposition Diagram** 31](#_Toc497790799)

[**4.4.2** **Functional Decomposition Diagram** 32](#_Toc497790800)

[Context Diagram 32](#_Toc497790801)

[Level 0 33](#_Toc497790802)

[**4.4.3** **Entiy Relationship Diagram** 34](#_Toc497790803)

[**4.4.4** **Class Diagram** 38](#_Toc497790804)

[**4.4.5** **Use Case Diagram** 39](#_Toc497790805)

[**4.4.6** **Use Case Full Description** 40](#_Toc497790806)

[**4.4.7** **Event Table** 49](#_Toc497790807)

[**4.4.8** **Activity Diagram** 50](#_Toc497790808)

[**4.4.9** **Sequence Diagram** 59](#_Toc497790809)

[Appendix A 60](#_Toc497790810)

[Appendix B 61](#_Toc497790811)

[Bibliography 62](#_Toc497790812)

# 

# Introduction

## **Project Context**

Currently, social media, word of mouth, flyers and/or posters are the common means of how people look for rentable vehicles. This principle also applies to people who want their cars to be rented. 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 cars, the proponents proposed to develop an Android-based mobile application called *Carkila* that will provide a venue for users – car owners and car renters – to interact with each other.

This mobile application will allow users – car owners and car renters – to either list a car or rent a car. This application will allow car 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 car renters find rentable vehicles near their location. Another feature is a bidding method wherein car renters can make an offer or negotiate the renting price of the vehicle. There will also be a private chat feature where users 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 limitations as to where drivers can advertise their vehicles to get clients. The majority of the respondents (64 out of 80) answered that if they were to rent out their car, 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 these data, the proponents concluded that a mobile application would be helpful for both car owners and car 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 bidding method in which the car renter may offer a reasonable price. The car 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.

## **Objectives**

**General Objectives:**

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

**Specific Objectives:**

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

## **Scope and Limitations**

The project is a vehicle rental mobile application that will run in Android. The scope of this project from the registration up until when a customer has finally rented a vehicle or when a transaction has been made between end-users. For car owners, they are required to submit some 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, and their fingerprints. They are also required to have a franchise in order to join. Therefore, they will also have to submit a DTI business name registration, provisional authority to operate a TNVS, and certificate for public convenience. The vehicle that a driver can register should not be older than 2014. Background checks will also be done for every driver and for additional security, all the information will be open for law enforcement and government to view. For the car 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. The car renters must verify their accounts through email or SMS code verification.

The app is not involved in the payment between the users. Payment is made completely between the driver and the customer. The app will not also be involved in doing necessary background checks when self-drive option is chosen since the main goal of the app is to provide an environment for car owners and renters who need to find rentable vehicles, and to make it easier to find rentable vehicles.

# Review of Related Literature

## **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 our 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 is 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.

## **Viking Car Rental**

Viking Car Rental System 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.

What makes this system different from other car rental system such as *Hertz* is that, it has a very friendly graphical user interface. It gives user an environment where it is not as technical as it should be. For example: car specifications. It only gives user the information he/she will be needing. Also, it has a wide variety of options to choose from; starting from self-drive up to one-way pickup.

## **Peer-to-Peer Rentals**

According to investopedia.com, Peer-to-Peer or P2P service is a decentralized platform where two individuals interact directly with each other, without the intermediation by a third-party, or without the use of a company of business selling a product or service. The past years, peer-to-peer e-commerce have been emerging. According to a study, peer-to-peer marketplaces have now expanded to provide short-term rental of products. There are different categories of peer-to-peer rental marketplaces. There are such marketplaces that offer car rentals, home/apartment rentals, and clothing rentals. These marketplaces differ from the usual e-commerce apps that have B2C or Business-to-Consumer models. In peer-to-peer rentals, the transaction is between two individuals rather than between a firm/company and an individual (Fraiberger & Sundararajan, 2016). Examples of these includes Airbnb, an online marketplace where one (guests) can rent hotel rooms, apartments, etc. from hosts (owners). Another example or P2P rental is *StyleLend*, which allows users to rent clothing items from other people.

According to an article written by Lydia Dishman, the founder of Rentalic.com – a website where people can rent to/from each other – said that “the whole idea is to build a community-based rental market place where individuals or businesses in a community can share goods and services by renting to/from each other”. Based on this article, there are a few number of growing peer-to-peer rental sites such as *Rentalic, Zilok,* and *Rent-Instead.*

## **Similar Systems in the Philippines**

## **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. 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.

## **iRenta.net**

*iRenta.net* is the first P2P or peer-to-peer car sharing platform where car owners can list their vehicles to earn money when their car is idle and where people can rent a nearby car at an hourly or daily rate. The website was said to be founded in 2016 and is based in Ormoc City.

## **Artificial Intelligence in Recommendation Services**

Machine learning powers the web searches, content filtering, and recommendation services. It was said to be used to identify objects in images, match items or products with users’ interests, and finds relevant search results with the use of techniques called deep learning. Deep learning is a form of artificial intelligence that allows machines to understand abstract data sets such as speech recognition, visual representations, objects detection, etc. It finds structure in large data sets with the use of backpropagation algorithm. Deep learning makes machines rewire the layers of its processing in order to learn the best path for processing abstract data. (LeCun, Bengio, Hinton, 2015).

YouTube is the world’s leading platform for sharing and discovering video content. In this regard, how is video content discovered? In a study entitled “Deep Neural Networks for YouTube Recommendation”, Paul Covington, Jay Adams, and Emre Sargin defined how deep learning affects the recommendations of every *youtuber* with content discovery. YouTube has many dynamic contents as a new video can be uploaded at any given time, which also means that the content is not static and the basic search engine with crawlers may not be quite as effective. The basic system overview on how YouTube provides its recommendation is through the context in which the user views contents on YouTube, and from the database of millions of videos on YouTube, the system gathers hundreds of candidates for recommendations. It then balances old and latest videos from its ranking and relevance to narrow down the search from hundreds and to dozens of recommendations. This is done at a constant rate to refresh recommendations every now and then (2016).

According to an article, an Artificial Intelligence recommendation engine can understand the customers’ preferences by using factors like how long they view a content, what device they use, what they read most, and many more. It was stated that using AI, it can connect the many patterns on user interactions, measure, analyze, and then make predictions. AI recommendation engines can also work to customize ads or sponsored content based on a user preferences and display these at the right time to the audience (Thomas, 2016).

Another article stated that Amazon had unveiled DSSTNE (“destiny”). DSSTNE is an open-source artificial intelligence framework used to power the company’s recommendation system which, according to them, can now be used by any company or researchers for their own AI applications. Large tech companies such as Google, Facebook, and Microsoft had focused on the branch of AI, which is called deep learning, and mainly uses it for image and speech recognition. However, Amazon is using it to sell merchandise. DSSTNE is a system that can make predictions based on less data. It was developed in order to recommend products that customers are more likely to click on and buy. (Finley, 2016)

Facebook had launched an AI-powered suggestions called *M.* According to Messenger’s product manager, “it is an automated assistant which is powered by artificial intelligence that detects intent and offers suggestions to enrich the way you communicate to get things done” (Wallbank, 2017). It appears within conversation and suggests actions such as sending a sticker and sharing a location. It was said that this will help the buried features resurface.

## **Artificial Intelligence in Matchmaking**

By definition, intelligent matching is a type of data management technique in which data is searched, indexed, and retrieved from a database through a series of artificial intelligence with data sorting and matching algorithms.

Machine learning is being used for a lot of things nowadays such as the type of content shown on the Facebook feed, suggested movies on Netflix, and matches suggested on dating apps. Now it is being used in the hiring process. The candidates are being filtered by using strings and keywords in resumes with matching algorithm. (Gutierrez, 2016)

According to an article, two former MIT students launched a website called *Beansprock* that is powered by AI. It uses natural language processing and machine learning to match people with suitable jobs. The system works by asking users to give it some information about their preferred jobs. On the other hand, it finds thousands of new jobs from public job listings and job sites. The system then compares the information with the users’ personal info and then the service’s algorithm match the users to a job suitable for them (Alba, 2016).

# Technical Background

The proposed project is a mobile application. Therefore, the proponents researched about (1) *Android*, where the app will be based on. The proponents plan to develop the project by using (2) *IDE or Integrated Development Environment* – which would be the main tool for the project – and by using the (3) *Java* programming language. (4) *Database* will also be a part of the proposed project as it would be needed for keeping all the data and records of the users. 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 customers to car 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.

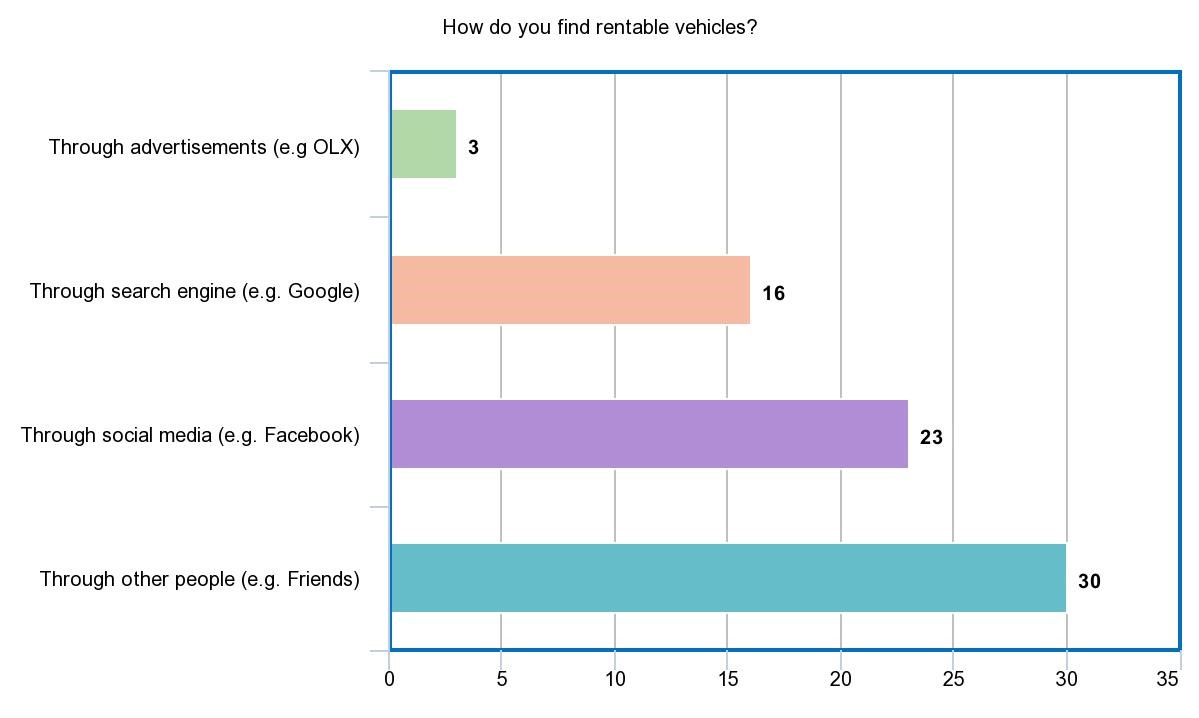


Figure 1

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.

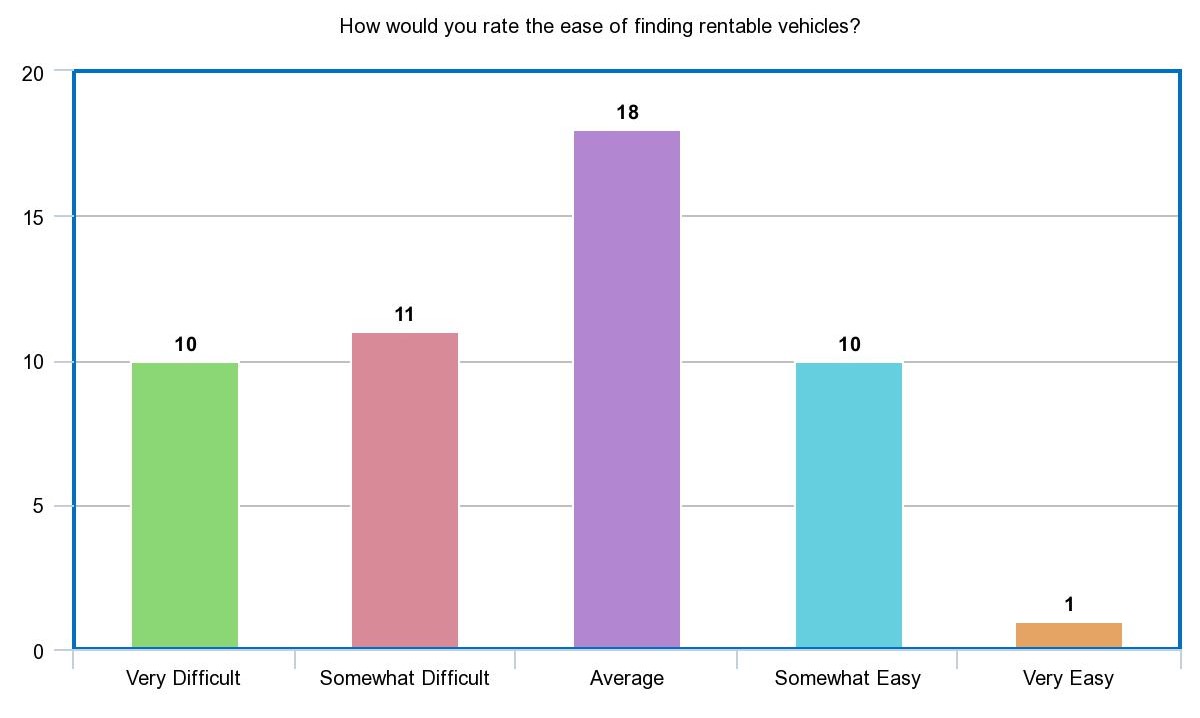


Figure 2

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.

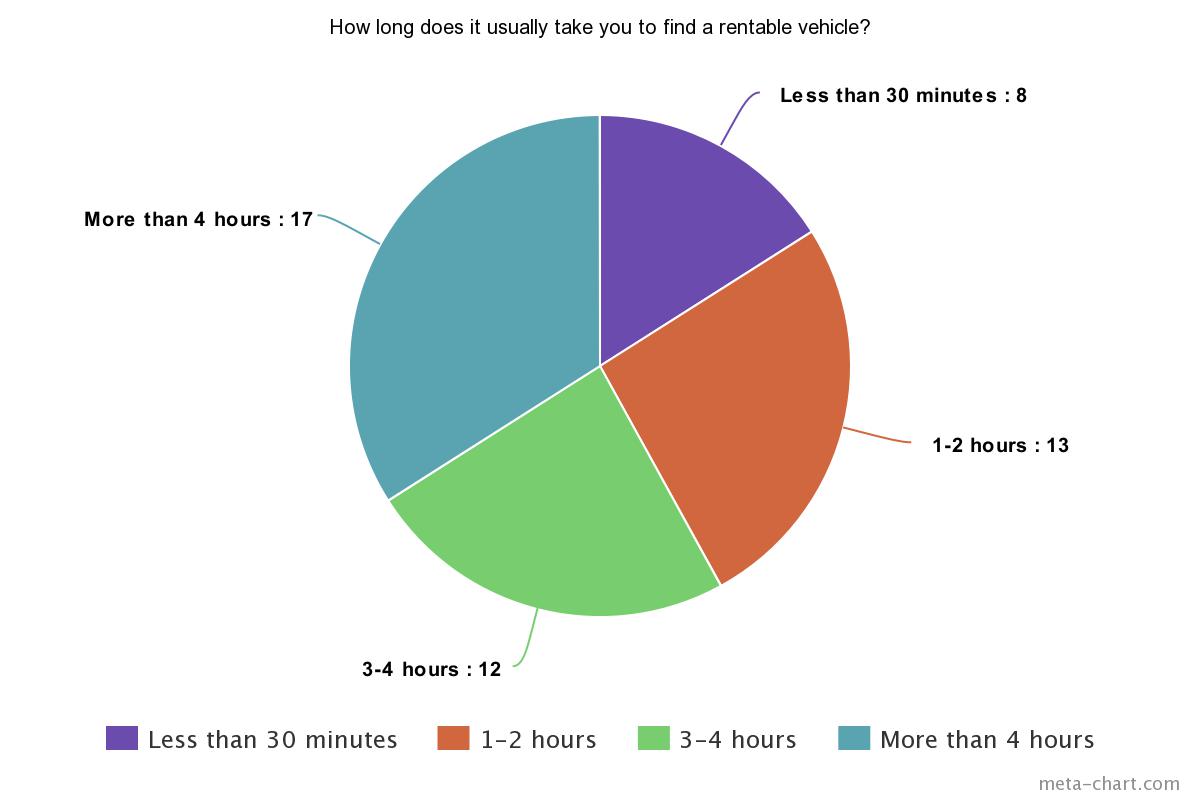


Figure 3

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.

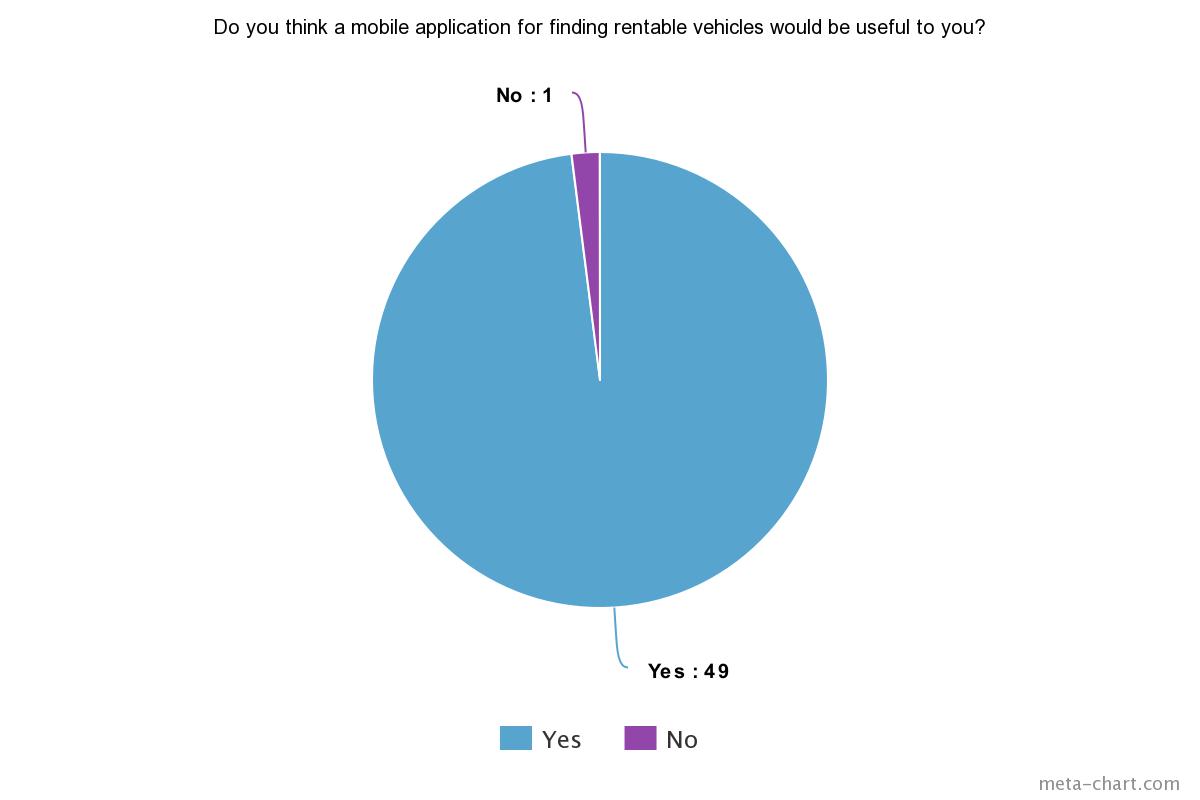


Figure 4

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.

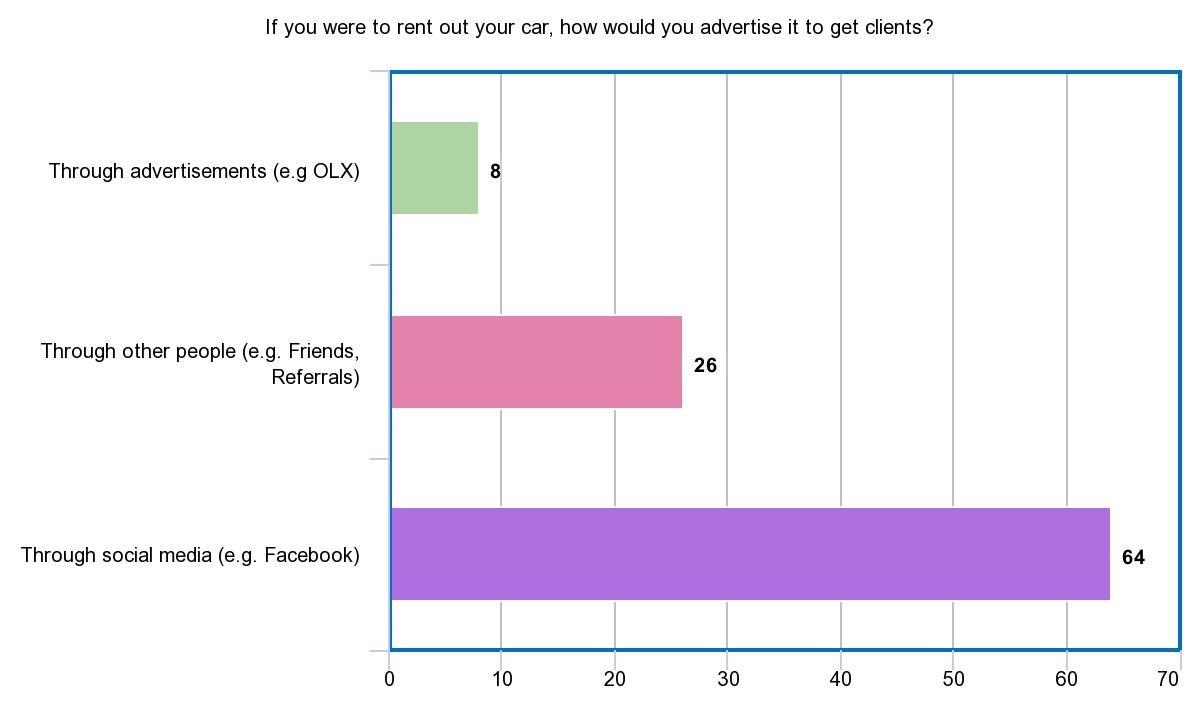


Figure 5

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.

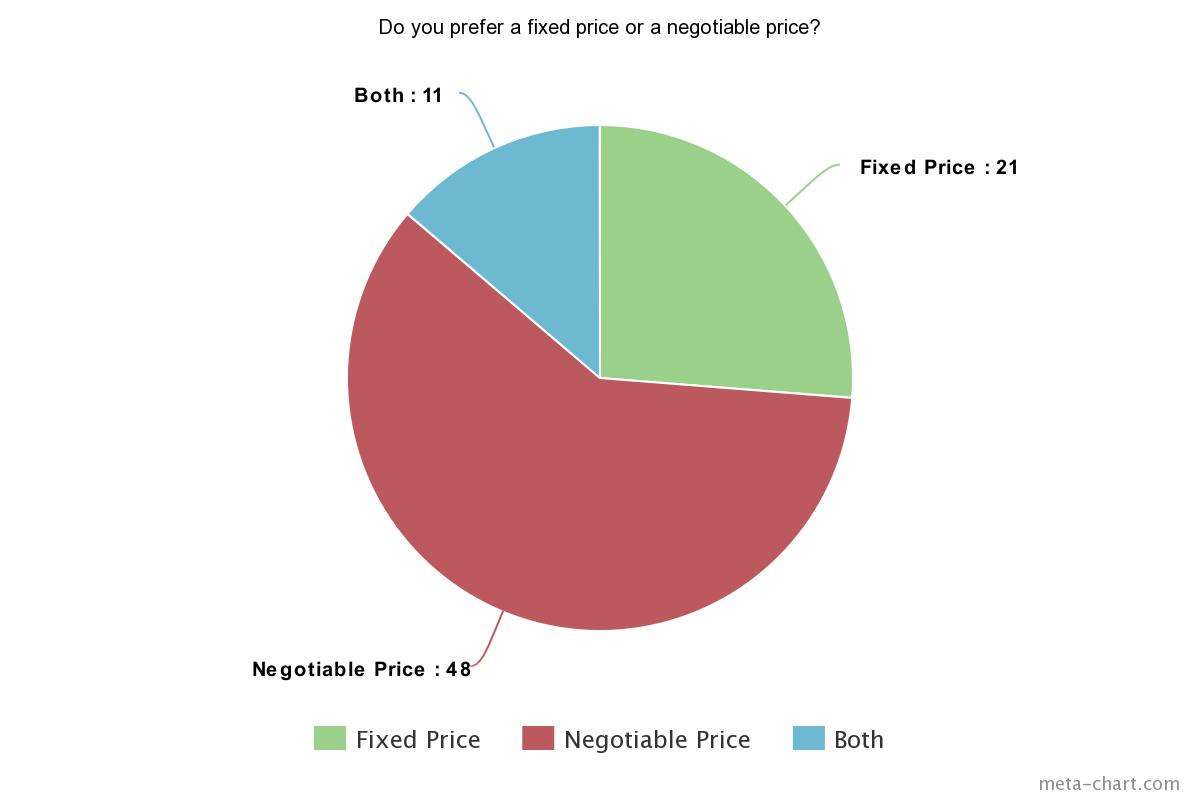


Figure 6

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.

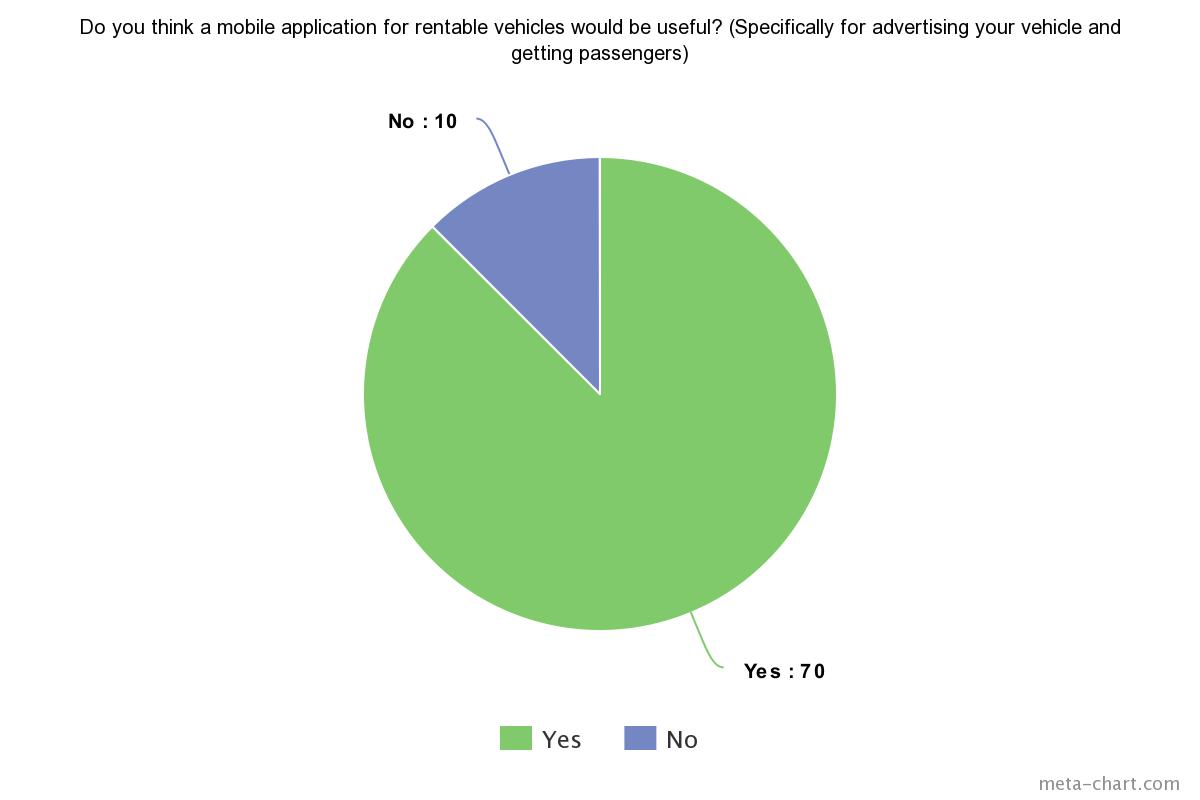


Figure 7

Figure 8

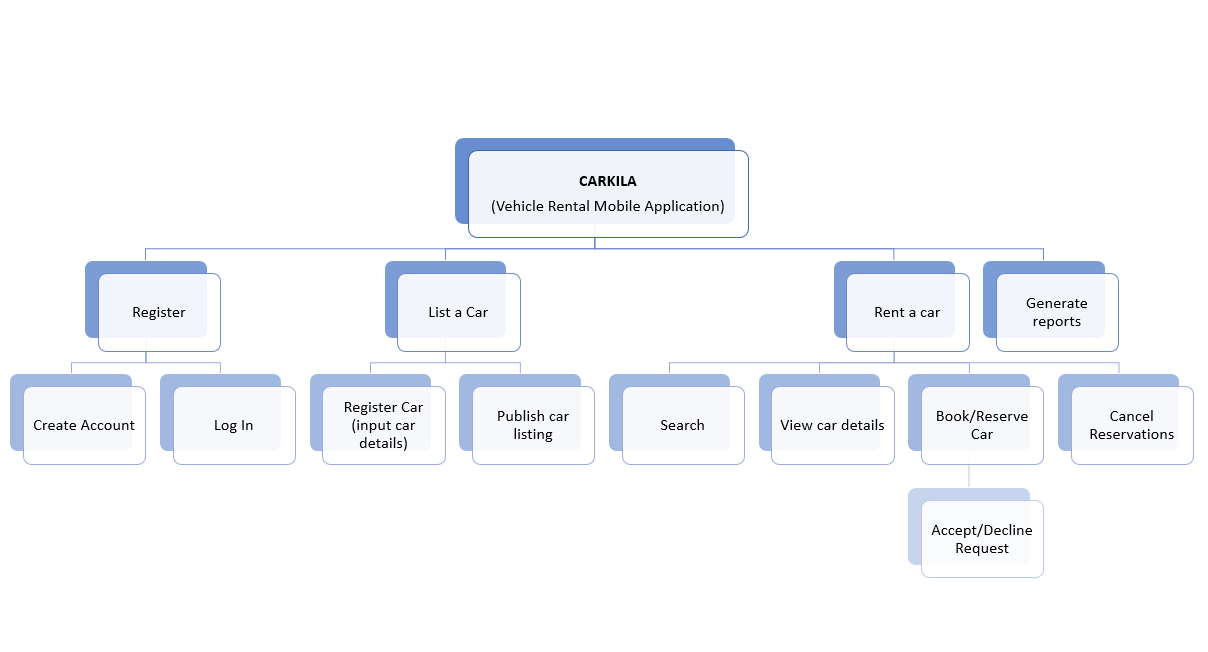
The proponents also asked if a mobile application could also be useful for advertising their vehicles 87.5% (70 out of 80) respondents answered yes while 12.5% (10 out of 80) answered no. Therefore, the proponents concluded that a mobile would also be indeed useful for the car owners.

## **Software Requirements Specification**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | ASSIGNED TO | YES | NO | REMARKS |
| **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 * LTFRB accreditation (franchise) * Car renters must include their trip details: * Pick-up location * Destination * Time (from-to) * Date (from-to) * Number of Passengers |  |  |  |  |
| **Process**   * The system must notify the car owner if someone requests his/her car. * The system must notify the car renter if the car owner accepted his/her request. * 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. |  |  |  |  |
| **Performance**   * The mobile application must be available 24/7. * The system must notify the car renters within 10 minutes after the car owner accepted/declined the request * The response time or car renters to the notification must not exceed 5 seconds. * The mobile application must work on all Android phones.   **Control**   * Only the administrator can generate the reports. |  |  |  |  |

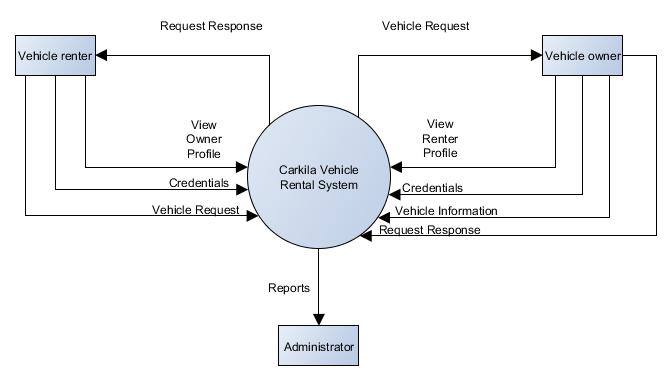
## **Design of Software, Systems, Product, and/or Processes**

## **Functional Decomposition Diagram**

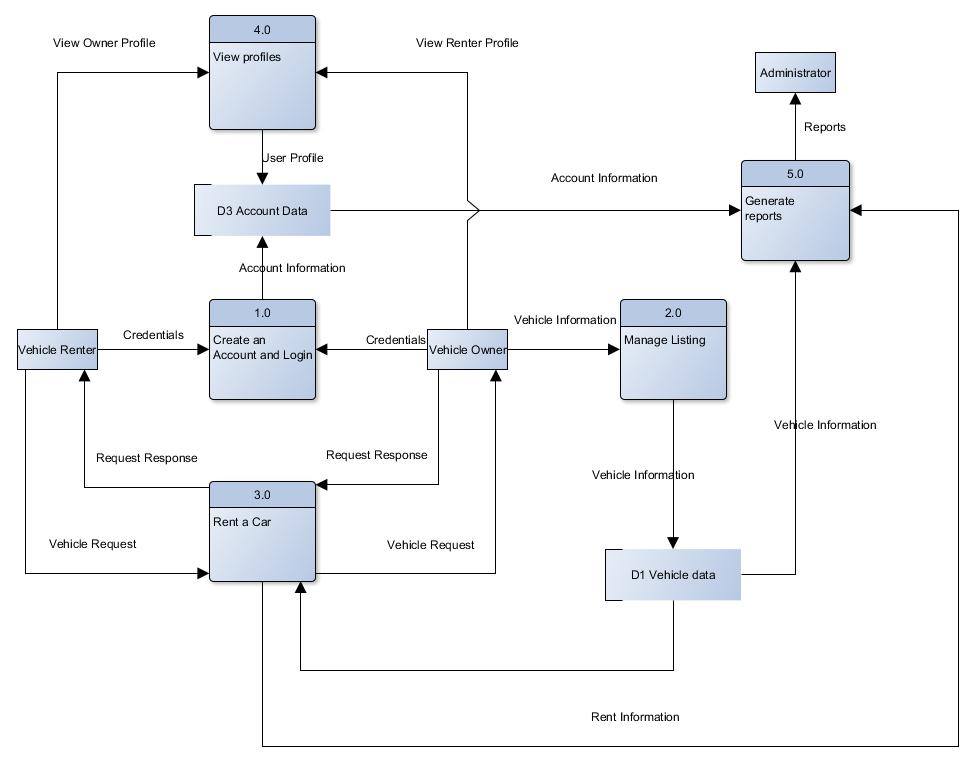


## **Functional Decomposition Diagram**

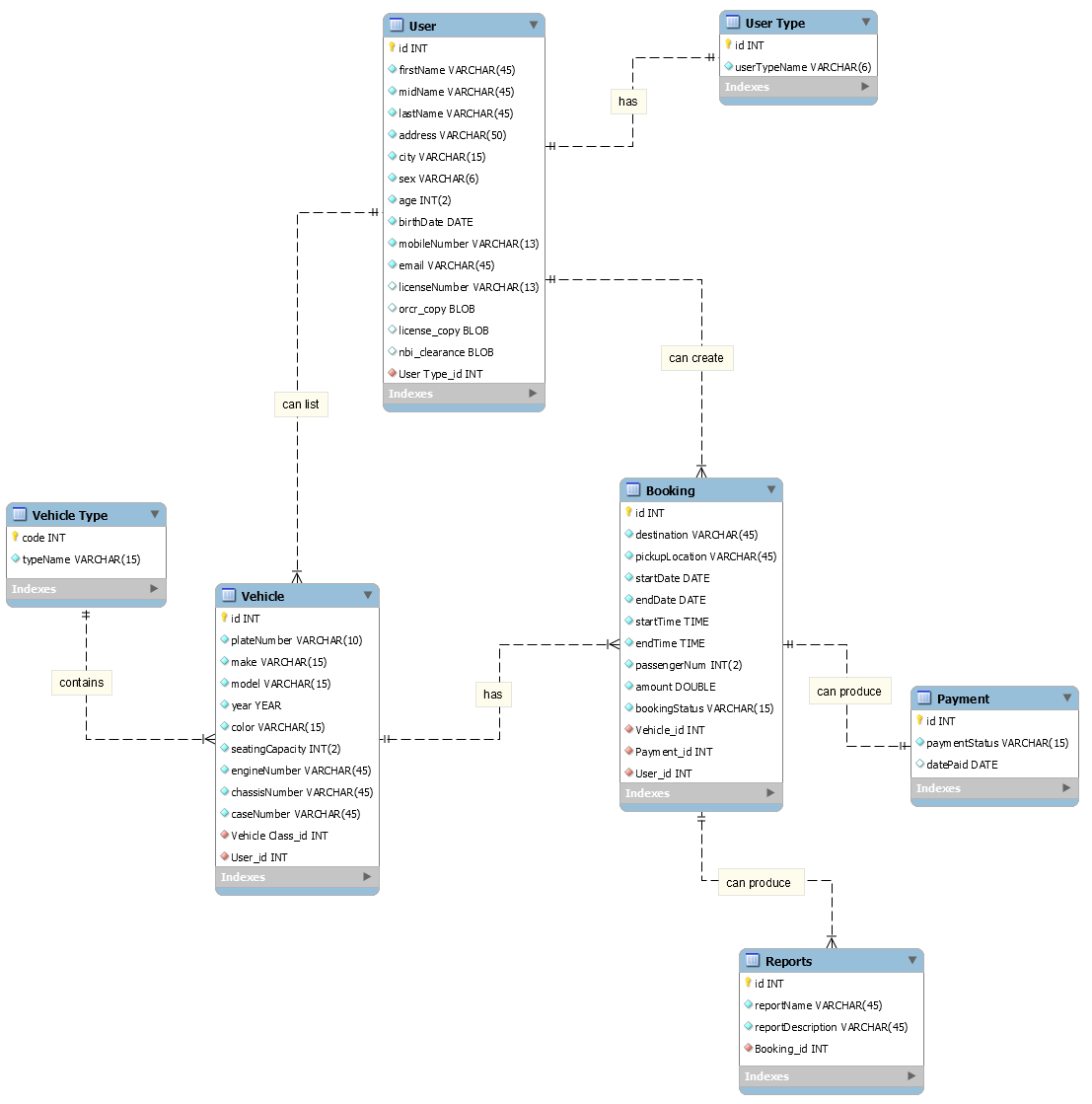
# Context Diagram



# Level 0



## **Entiy Relationship Diagram**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE: **User** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| id | INT |  | Primary Key | User ID, Auto-generated |
| 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 |
| email | VARCHAR | 45 | Not Null | Email Address |
| licenseNumber | VARCHAR | 13 | Null | License Number |
| orcr\_copy | BLOB |  | Null | Photocopy of Vehicle OR/CR |
| license\_copy | BLOB |  | Null | Photocopy of Driver’s License |
| nbi\_clearance | BLOB |  | Null | Photocopy of NBI Clearance |
| User Type\_id | INT |  | Foreign Key | User Type ID |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE: **User Type** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| id | INT |  | Primary Key | Payment ID |
| userTypeName | VARCHAR | 6 | Not Null | Payment Status |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |
| caseNumber | VARCHAR | 45 | Not Null | Franchise Case Number |
| rentalRate | DOUBLE |  | Not Null | Vehicle Rental Rate |
| Vehicle Type\_id | INT |  | Foreign Key | Vehicle Type ID |
| User\_id | INT |  | Foreign Key | User ID |

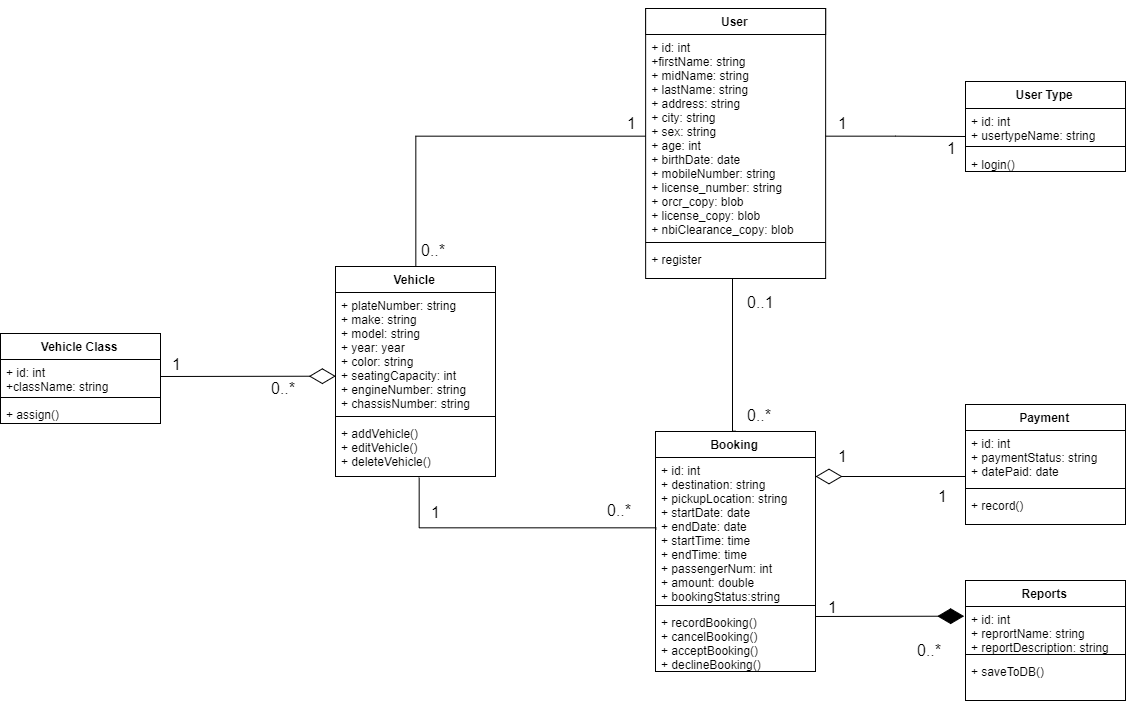
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE: **Vehicle Type** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| Code | INT |  | Primary Key | Vehicle Type Code |
| typeName | VARCHAR | 15 | Not Null | Vehicle Type Name |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 |
| Payment\_id | INT |  | Foreign Key | Payment ID |
| User\_id | INT |  | Foreign Key | User ID |

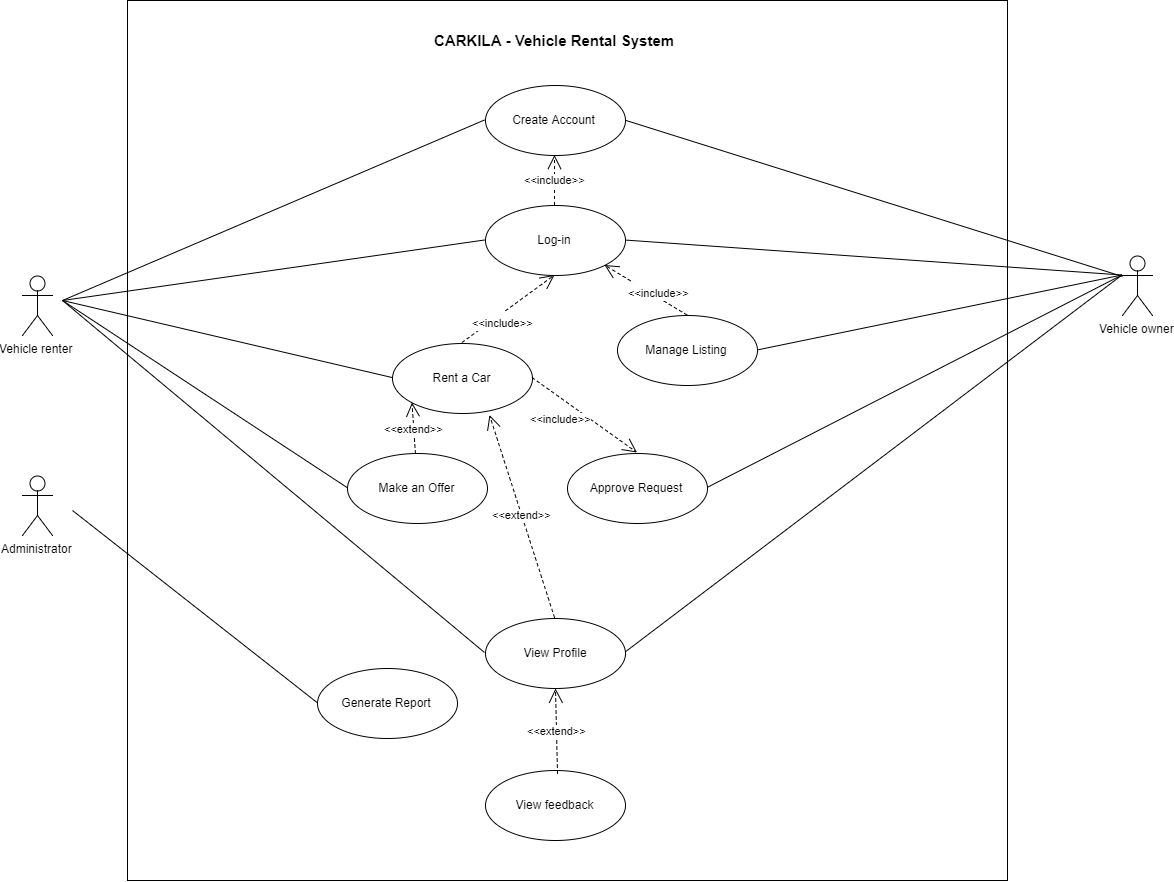
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TABLE: **Reports** | | | | |
| **Field Name** | **Data Type** | **Field Length** | **Constraint** | **Description** |
| id | INT |  | Primary Key | Report ID, Auto-generated |
| reportName | VARCHAR | 45 | Not Null | Report Name |
| reportDescription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

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

## **Class Diagram**



## **Use Case Diagram**



## **Use Case Full Description**

|  |  |  |
| --- | --- | --- |
| **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:** | Log In | |
| **Stakeholders:** | Vehicle owner, vehicle renter | |
| **Preconditions:** | The user must have the mobile application installed. | |
| **Postconditions:** | The data entered must be stored to the database.  Account verification must be sent to the user. | |
| **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 and displays the verification page.  4.1 The system validates the verification code.  4.2 The system stores all the data entered.  4.3 The system directs the user to the home page. |
| **Exception Conditions:** | If the user entered invalid data.  If the user entered invalid verification code.  If the user cancels the request. | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Log In | |
| **Scenario:** | User wants log in. | |
| **Triggering Event:** | Log in request | |
| **Brief Description:** | This will allow the user to log in. | |
| **Actors:** | Vehicle owner, vehicle renter | |
| **Related Use Case:** | Create Account | |
| **Stakeholders:** | Vehicle owner, vehicle renter | |
| **Preconditions:** | The user must create an account. | |
| **Postconditions:** | 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 **Log In**. 3. The user enters his credentials and clicks on Log In button. | * 1. The system displays the create account/log in page.   2. The system prompts the user to enter email and password.   3. The system validates credentials.   3.2 The system directs the user to the home page. |
| **Exception Conditions:** | If the user entered an invalid email/password.  If the user does not have an account yet, initiate *Create Account* use case. | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Rent a Car | |
| **Scenario:** | Vehicle renter wants to rent a car. | |
| **Triggering Event:** | Request a car | |
| **Brief Description:** | This will allow the user to rent a car. | |
| **Actors:** | Vehicle renter | |
| **Related Use Case:** | Make an Offer, Approve Request, View Profile | |
| **Stakeholders:** | Vehicle renter | |
| **Preconditions:** | The vehicle renter must be logged in. | |
| **Postconditions:** | The vehicle must be able to book a car.  A notification must be sent to the vehicle owner. | |
| **Assumptions:** | The vehicle renter is already logged in. | |
| **Flow of Activities:** | **Actor** | **System** |
| 1. The vehicle renter taps on **Browse Cars.** 2. The vehicle renter clicks on a vehicle listing. 3. The vehicle renter clicks **Book Now**. 4. The user enters the trip details. 5. Click **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 the renter wants to check the profile of owner, initiate *View Profile* use case.   4.1 If the user wants to negotiate the renting price of the vehicle, initiate *Make an Offer* use case. | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Manage Listing | |
| **Scenario:** | Vehicle owner wants to list his car. | |
| **Triggering Event:** | Request to manage listing | |
| **Brief Description:** | This will allow the vehicle owner to create, view, update, or delete a vehicle listing. | |
| **Actors:** | Vehicle owner | |
| **Related Use Case:** |  | |
| **Stakeholders:** | Vehicle owner | |
| **Preconditions:** | The vehicle owner must be logged in. | |
| **Postconditions:** | The listing must be published in the mobile application. | |
| **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 **Add Vehicle Listing.**  3. The vehicle owner enters all required information. | * 1. The system displays the listing page.   2.1 The system prompts the user to enter required vehicle information and upload required documents.  3.1 The system will store the information and publish the vehicle listing in the app. |
| **Exception Conditions:** | 2.b If the vehicle owner wants to view, the system displays the vehicle listing details.  2.c If the vehicle owner wants to edit the listing, the system displays the listing in edit mode.  2.d If the vehicle owner wants to delete an existing listing, The system removes the listing from the app. | |

|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Make an offer | |
| **Scenario:** | The vehicle renter wants to make an offer. | |
| **Triggering Event:** | Offer request | |
| **Brief Description:** | This will allow the vehicle renter to negotiate the renting price of the vehicle. | |
| **Actors:** | Vehicle renter | |
| **Related Use Case:** | Rent a car, Create Account, Log In, Approve Request | |
| **Stakeholders:** | Vehicle renter | |
| **Preconditions:** | The vehicle renter must be logged in. | |
| **Postconditions:** | The offer request must be included in the trip details that will be sent to the vehicle owner. | |
| **Assumptions:** | The vehicle renter already clicked on a vehicle listing tapped the **Book Now** button. | |
| **Flow of Activities:** | **Actor** | **System** |
| 2. The vehicle renter enters the trip details.  3. The vehicle renter clicks on **Make an offer**.  4. The vehicle renter enters the desired price.  5. The vehicle renter clicks **Send Request.** | 1. The system prompts the renter to enter trip details.  3.1 The system prompts the renter to enter the price offer.  4. 1 The system redirects the page back to Booking page.  5.1 The system will send the rent request to the vehicle owner. |
| **Exception Conditions:** | - | |

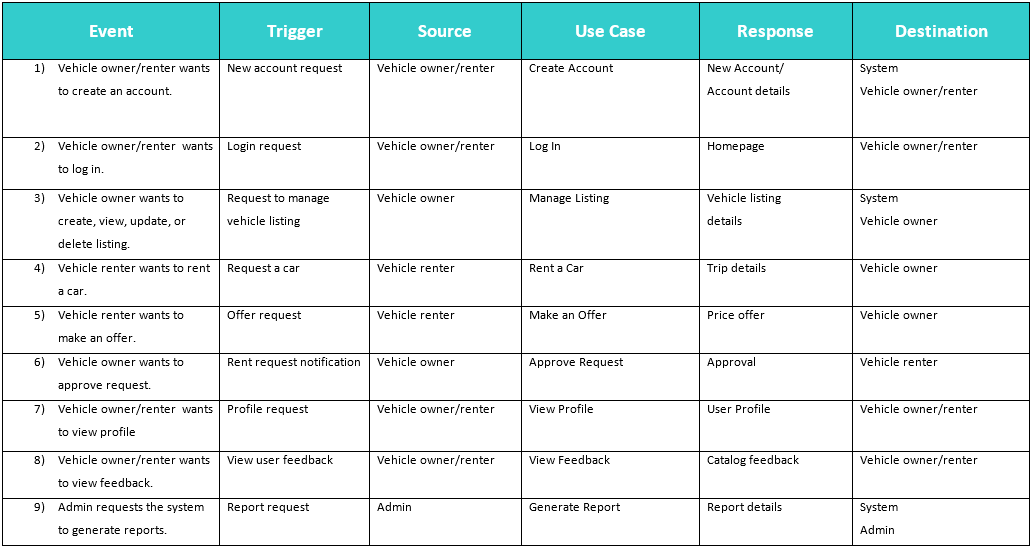
|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Approve Request | |
| **Scenario:** | Vehicle owner wants to accept a rent request. | |
| **Triggering Event:** | Request to manage listing | |
| **Brief Description:** | This will allow the vehicle owner to create, view, update, or delete a vehicle listing. | |
| **Actors:** | Vehicle owner | |
| **Related Use Case:** | Rent a Car | |
| **Stakeholders:** | Vehicle owner | |
| **Preconditions:** | The vehicle owner must be logged in.  The vehicle owner must receive a rent request notification. | |
| **Postconditions:** | 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 renter 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 systems notifies the renter that the rent request has been accepted. |
| **Exception Conditions:** | 3. If the vehicle owner wants to view profile, initiate *View Profile* use case.  4.1 If the vehicle owner declines the request, the system notifies the vehicle renter that the rent request had been declined. | |

|  |  |  |
| --- | --- | --- |
| **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. | |
| **Actors:** | Vehicle renter | |
| **Related Use Case:** | Rent a Car | |
| **Stakeholders:** | Vehicle renter | |
| **Preconditions:** | The vehicle renter must select a vehicle listing. | |
| **Postconditions:** | The vehicle renter must be able to see the vehicle owner’s profile. | |
| **Assumptions:** | The vehicle renter is already logged in. | |
| **Flow of Activities:** | **Actor** | **System** |
| 1. The vehicle renter opens a listing. 2. 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:** | 3. If the vehicle renter wants to view the feedbacks on the owner, initiate *View Feedback* use case. | |

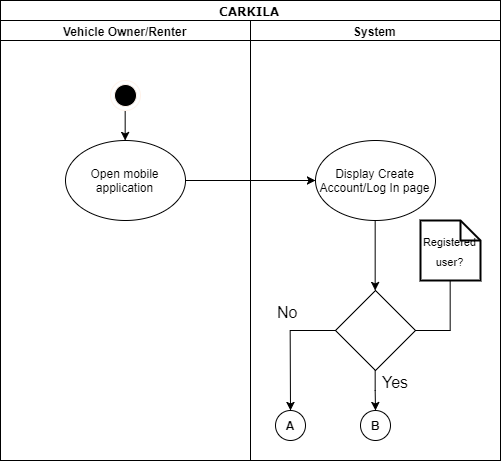
|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | View Feedback | |
| **Scenario:** | The vehicle renter wants to view the feedbacks on the owner. | |
| **Triggering Event:** | View user feedback | |
| **Brief Description:** | This will allow the vehicle renter to view the feedbacks on the owner. | |
| **Actors:** | Vehicle renter | |
| **Related Use Case:** | View Profile | |
| **Stakeholders:** | Vehicle renter | |
| **Preconditions:** | The vehicle renter must select a vehicle listing. | |
| **Postconditions:** | The vehicle renter must be able to see the feedbacks. | |
| **Assumptions:** | The vehicle renter is already on the profile of the vehicle owner. | |
| **Flow of Activities:** | **Actor** | **System** |
| 1. The vehicle renter taps **Feedbacks.** | 1.1 The system displays the feedback catalog. |
| **Exception Conditions:** | - | |

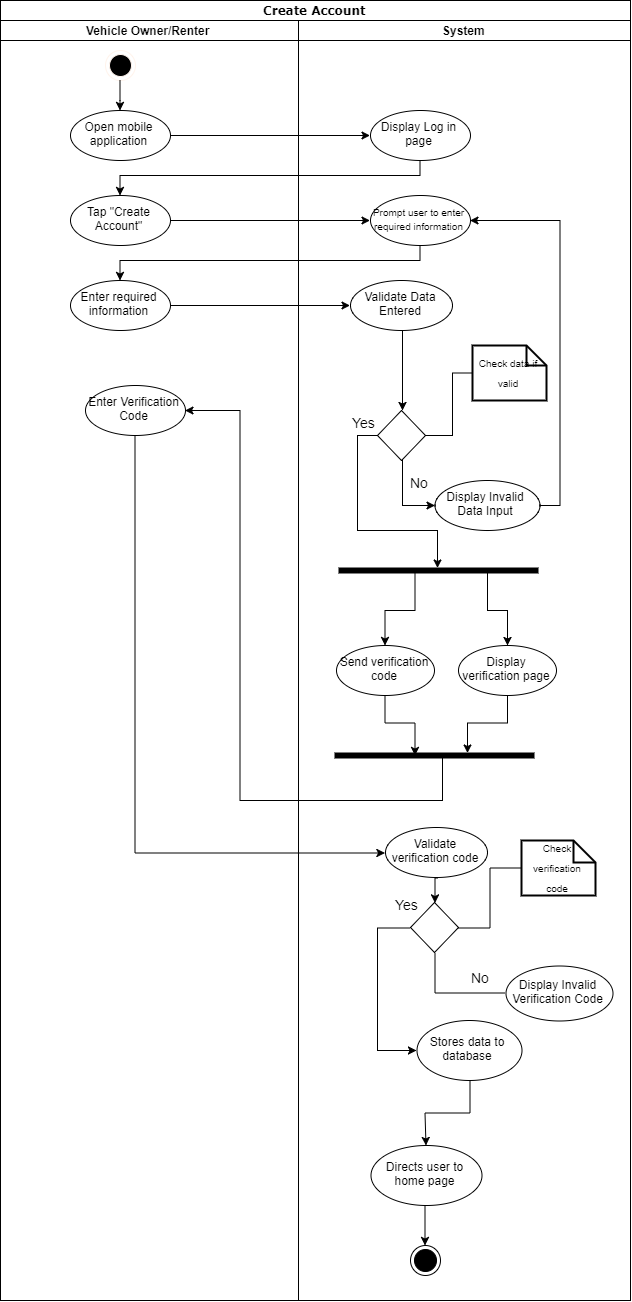
|  |  |  |
| --- | --- | --- |
| **Use Case Name:** | Generate Reports | |
| **Scenario:** | The admin requests the system to generate reports | |
| **Triggering Event:** | Report request | |
| **Brief Description:** | This will allow the administrator to generate a report. | |
| **Actors:** | Administrator | |
| **Related Use Case:** |  | |
| **Stakeholders:** | Administrator | |
| **Preconditions:** | There must be existing bookings. | |
| **Postconditions:** | 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 sends the generated report.   4. The system stores the report in the database. |
| **Exception Conditions:** | If there is no existing booking. | |

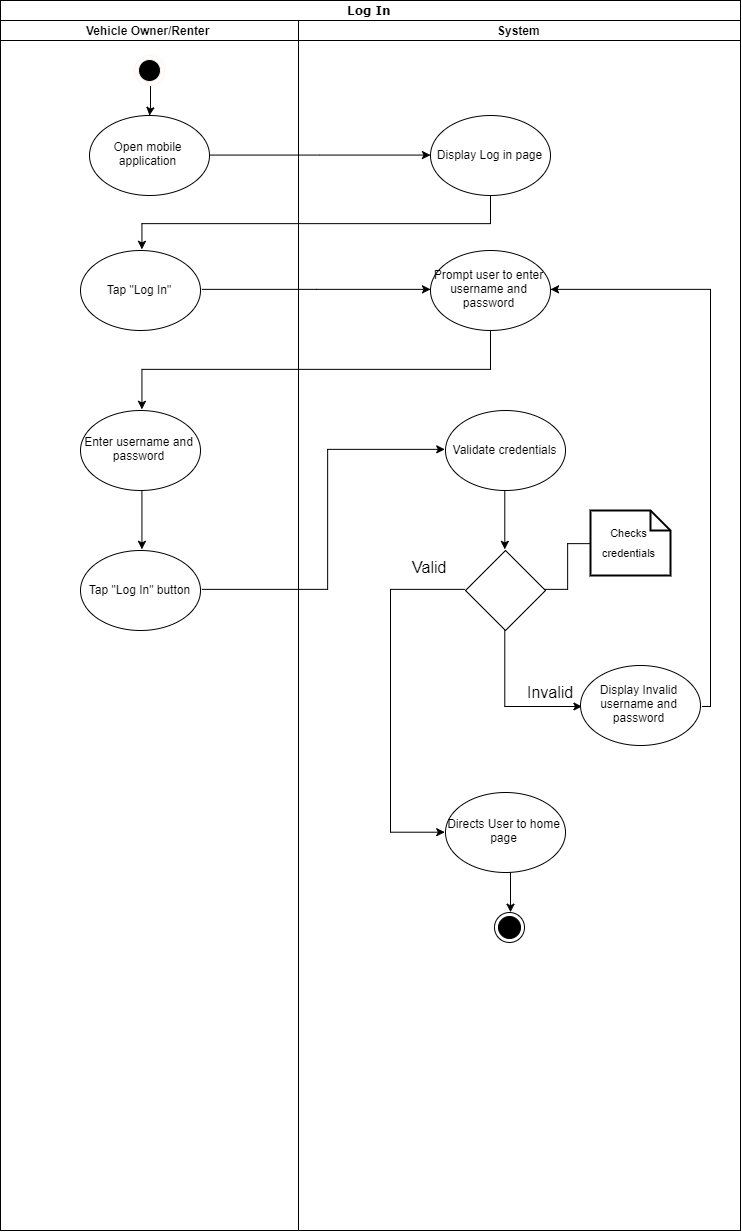
## **Event Table**

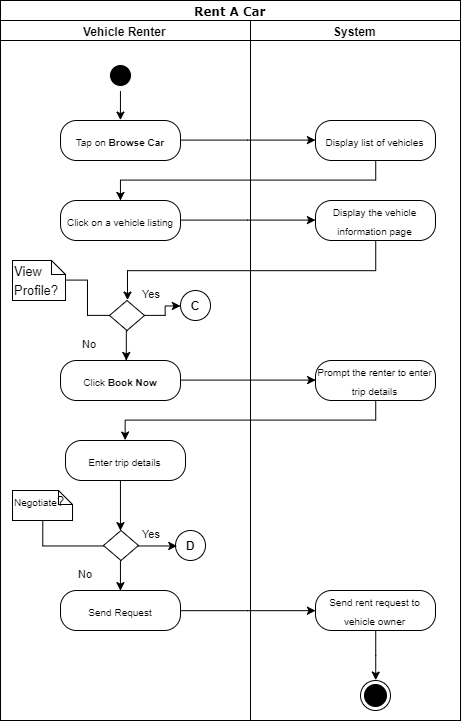


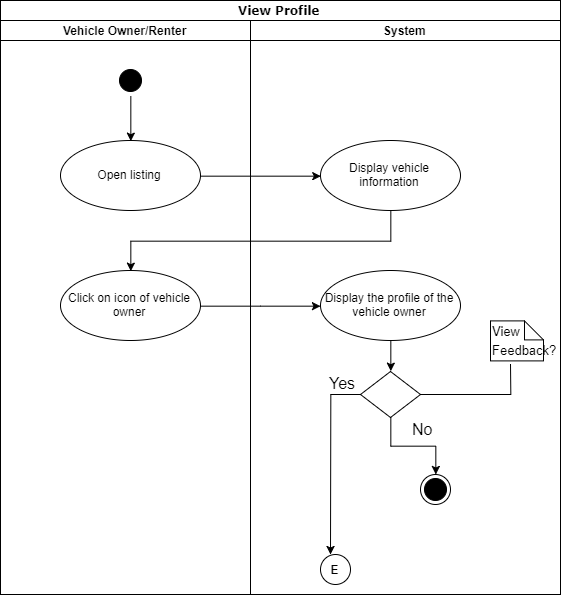
## **Activity Diagram**

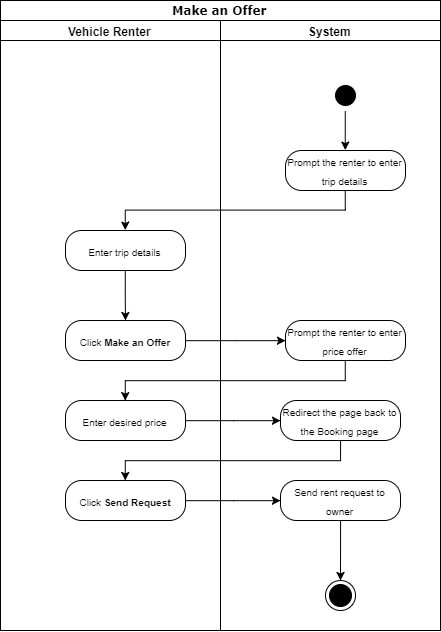


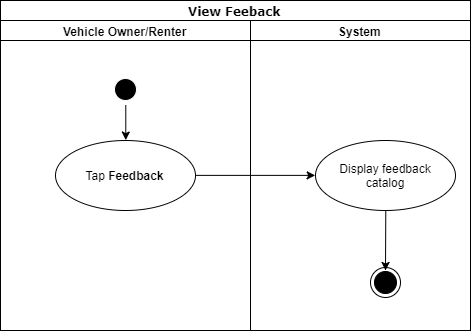


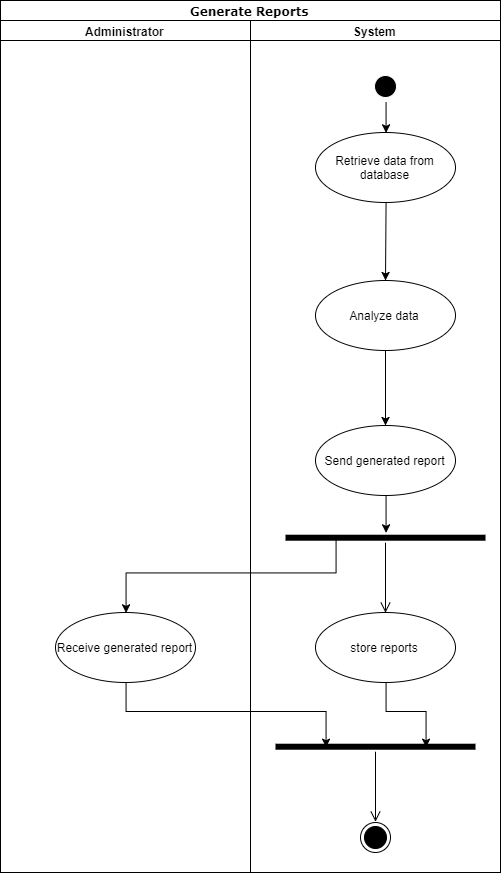


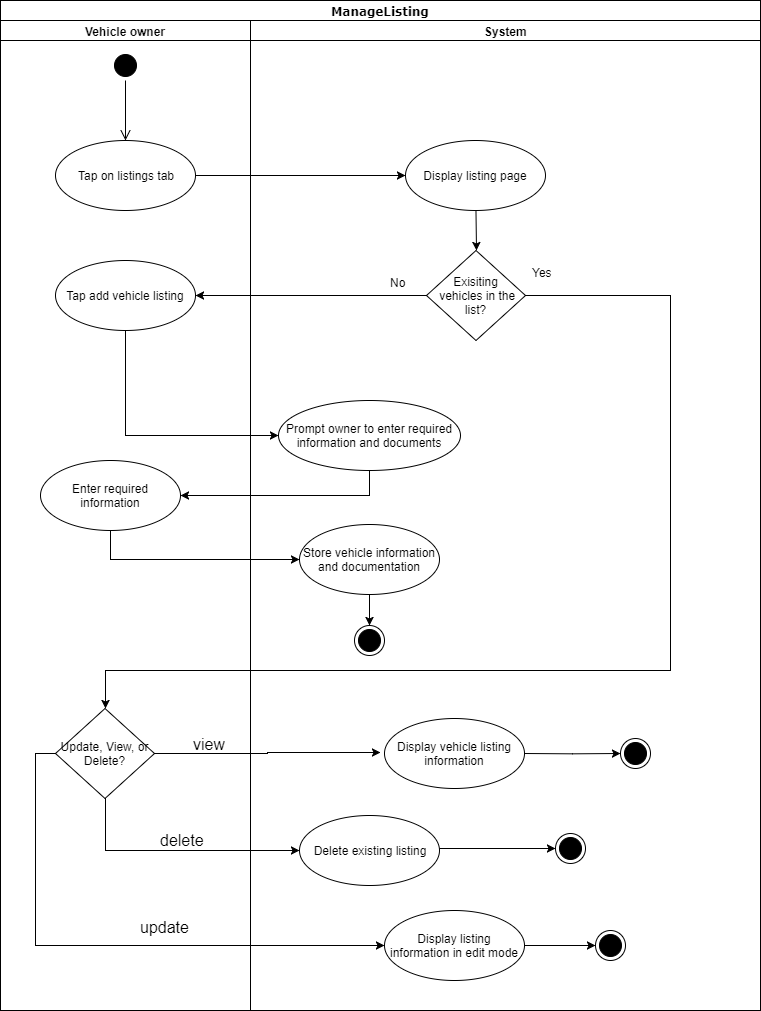












## **Sequence Diagram**

# Appendix A

# Appendix B

# Bibliography

Alba, D. (2016, February 4). *AI Software That Could Score You The Perfect Job*. Retrieved from https://www.wired.com/2015/02/beansprock/

*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/

Finley, K. (2016, May 16). *Amazon's Giving Away the AI Behind Its Product Recommendations*. Retrieved from https://www.wired.com/2016/05/amazons-giving-away-ai-behind-product-recommendations/

Fraiberger, S., & Sundararajan, A. (2016, October 6). *Peer-to-Peer Rental Markets in the Sharing Economy*. Retrieved from https://papers.ssrn.com/sol3/papers.cfm?abstract\_id=2574337

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/

Gutierrez, D. (2016, July 2). *How Machine Learning Will Transform the Way Employers and Candidates Connect*. Retrieved from https://insidebigdata.com/2016/07/02/how-machine-learning-will-transform-the-way-employers-and-candidates-connect/

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

LeCun, Y., Bengio, Y., & Hinton, G. (2015, May 28). Deep Learning.

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

Peltier, D. (2015, November 2). *5 New Travel Startups Redefining the Model for Car Rentals*. Retrieved from https://skift.com/2015/11/02/5-new-travel-startups-redefining-the-model-for-car-rentals/#1

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

Thomas, T. (2016, September 15). *How Artificial Intelligence Recommendation engines will change Online Publishing*. Retrieved from https://boomtrain.com/artificial-intelligence-recommendation-engines-publishing/

Vdovychenko, H. (2017, July 31). *How To Build a Peer to Peer Car Rental App Like Turo.* Retrieved from JSSolutions: https://jssolutionsdev.com/blog/how-to-build-a-car-rental-app-like-turo/

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

Wallbank, P. (2017, August 16). *Facebook launches M suggestions in Messenger powered by AI*. Retrieved from https://mumbrella.com.au/facebook-launches-m-suggestions-messenger-powered-ai-464957

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