

ACKNOWLEDGEMENT

I would like to express our special thanks of gratitude to our supervisor **Milan Baral** who gave us the golden opportunity to do this wonderful project on the topic of **Online Vehicle Rental System**, which also helped us in doing a lot of research and we came to know about so many new tools and technologies.

I would like to express my special gratitude and thanks to our BCA Program Coordinator **Milan Baral** for his support and help for our personnel development and mainly for the completion of this project.

I am highly indebted to **TRITON INTERNATIONAL COLLEGE** for their guidance and constant supervision as well as for providing necessary information regarding the Project and support in the completion.

I would also like to express my gratitude towards library and member of Triton International College for their kind co-operation and encouragement which help me in completion of this project.

I would also like to thank our parents and friends who helped us a lot in finalizing this project within the limited time frame

In the end, we would also like to thank Tribhuvan University for giving us this opportunity via the course of Computer Application to help us understand the project ethics at this early stage and helped us to evaluate my knowledge and expand it a little more.

ABSTRACT

In the partial fulfillment of the requirement for the degree of BCA project titled “Online Vehicle Rental” is an online portal to search for self-driving rental vehicle. This system is built to let the vehicle owners able to rent their vehicle without any third party inbetween. And, people willing to get a vehicle on rent to drive throughout the journey by them. This website has search options where users can easily search for Vehicles in your desired area. Users can easily review the vehicles rented. This standalone web application is built with object-oriented approach. This project uses haversine algorithm to calculate the distance between two locations and content-based filtering to recommend vehicle according to users’ preference. The project includes IntelliJ IDEA and Visual Studio Code as design and development software, Vue JS as frontend, Java Spring Boot as programming language and MySQL database as backend for the development.

Keyword: Car rental, Vehicle rental, Spring boot, Java, DBeaver

TABLE OF CONTENTS

ACKNOWLEDGEMENT.....	i
ABSTRACT.....	ii
TABLE OF CONTENTS.....	iii
LIST OF ABBREVIATIONS	v
LIST OF FIGURES	vi
LIST OF TABLES.....	viii
CHAPTER 1: INTRODUCTION.....	1
1.1 Introduction	1
1.2 Problem Statement	1
1.3 Objectives.....	1
1.4 Scope and limitation.....	1
1.4.1 Scope.....	1
1.4.2 Limitation.....	2
1.5 Development Methodology	2
1.6 Report Organization	3
CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW.....	5
2.1 Background Study.....	5
2.2 Literature Review.....	5
CHAPTER 3: SYSTEM ANALYSIS AND DESIGN.....	7
3.1 System Analysis	7
3.1.1 Requirement Analysis	7
3.1.2 Feasibility Study	9
3.1.3 Object Modeling using Class and Object diagram	11
3.1.4 Dynamic Modelling using State and Sequence Diagrams	13
3.1.5 Process Modelling using Activity Diagrams.....	14

3.2 System Design	17
3.2.1 Refinement of Class, Object, State, Sequence and Activity diagrams.....	17
3.2.2 Component Diagram.....	22
3.2.3 Deployment Diagram.....	22
3.3 Algorithm	23
3.3.1 Haversine Algorithm:.....	23
3.3.2 Content-based filtering Algorithm:	24
CHAPTER 4: IMPLEMENTATION AND TESTING.....	25
4.1 Implementation	25
4.1.1 Tools Used.....	25
4.1.2 Implementation Details of Modules.....	26
4.2 Testing.....	27
4.2.1 Test Cases for Unit Testing	28
4.2.2 Test Cases for System Testing.....	30
4.3 Result Analysis.....	32
CHAPTER 5: CONCLUSION.....	34
5.1 Conclusion	34
5.2 Future Recommendations	34
References.....	35
APPENDIX	

LIST OF ABBREVIATIONS

CSS	Cascading Style Sheet
GUI	Graphical User Interface
HTML	Hypertext Markup Language
MySQL	Structured Query Language

LIST OF FIGURES

Figure 1.1: Agile Methodology of Online Vehicle Rental System	3
Figure 3.1: Use case diagram of Online Vehicle Rental System.....	8
Figure 3.2: Gannt Chart of Online Vehicle Rental System	10
Figure 3.3: Class Diagram of Online Vehicle Rental System	11
Figure 3.4: Object Diagram of Online Vehicle Rental System	12
Figure 3.5: Sequence Diagram of User of OVRS	13
Figure 3.6: Sequence Diagram of Vehicle Owner of OVRS.....	13
Figure 3.7: Sequence Diagram of Admin of OVRS.....	13
Figure 3.8: Activity Diagram of Admin of OVRS	14
Figure 3.9: Activity Diagram of Vehicle Owner of OVRS.....	15
Figure 3.10: Activity Diagram of Admin of OVRS	16
Figure 3.11: Refined Class Diagram of Online Vehicle Rental System	17
Figure 3.12: Refined Object Diagram of Online Vehicle Rental System.....	18
Figure 3.13: Refined Sequence Diagram of User of OVRS	19
Figure 3.14: Refined Sequence Diagram of Vehicle Owner of OVRS	19
Figure 3.15: Refined Sequence Diagram of Admin of OVRS.....	19
Figure 3.16: Refined Activity Diagram of Admin of OVRS.....	20
Figure 3.17: Refined Activity Diagram of Vehicle Owner of OVRS	20
Figure 3.18: Refined Activity Diagram of Admin of OVRS.....	21
Figure 3.19: Component Diagram of OVRS	22

Figure 3.20: Deployment Diagram of OVRS	22
Figure 3.21: Haversine Algorithm	23

LIST OF TABLES

Table 4.1: Test Case of admin login of OVRS.....	28
Table 4.2: Test Case of vehicle owner registration of OVRS.....	28
Table 4.3: Test Case of vehicle owner login of OVRS.....	29
Table 4.4: Test Case of User registration of OVRS.....	29
Table 4.5: Test Case of user login of OVRS.....	30
Table 4.6: Test Case of Add-Vehicle of OVRS.....	31
Table 4.7: Test Case of Vehicle Deletion of OVRS.....	31
Table 4.8: Test Case of Search-Vehicle of OVRS.....	32
Table 4.9: Test Case of Book-Vehicle of OVRS.....	32

CHAPTER 1: INTRODUCTION

1.1 Introduction

The project "Online Vehicle Rental System" is an online portal which has specially been designed & developed to resolve the current market problem of renting a vehicle directly through a web portal. The vehicle owners will be provided the full privileges of managing their vehicle information from data entry to booking management. And, the vehicle renters to upload & manage required legal documents and also view their booking history. The system has been given a smart search engine for searching the vehicle according to journey locations & date required.

1.2 Problem Statement

The problem statement of this project is to develop an online process of vehicle rental service. The current system is time consuming as customers have to call manually to rent or reserve a vehicle.

Many vehicle owners buy vehicles but do not use regularly. There are middle class people who want to buy vehicles but cannot afford to do so. Rather than hiring a driver people prefer to self-drive as most as possible. Tourists that like to drive while visiting cannot rent it in affordable price.

The goal of this project is to automate vehicle rental and reservation so that customers do not need to call and spend unnecessary time in order to reserve a vehicle and the passive owners can utilize their unused vehicle.

1.3 Objectives

- To target for vehicle owners who intend to rent their vehicle for some extra income,
- To manage details of vehicle, payment, user and owner,
- To reduce manual work for managing vehicle, payment, user and vehicle owner,
- To allow users to book/rent vehicles.

1.4 Scope and limitation

1.4.1 Scope

Vehicle Owners, in this online portal a vehicle owner can rent their vehicle online in this platform and can also get cash even in the leisure time or holidays when their vehicles are

not being used or can also earn enough money by continuously let rent their vehicle in this platform. A vehicle owner can also rent their vehicle in this portal according to their feasible Location and time.

Users, a user can book a self-drive vehicle which was listed in the portal by the vehicle owner and can take that vehicle to the place he/she desired to be with a freedom to use the vehicle efficiently. A user will get different facilities and discounts on the booking as well and can get varieties of option in vehicle he wants to take for a few days. There will be no involvement of third party, generally known as broker as this online portal will be a platform for both the vehicle owner and user to get in touch once booking is done and mutually discuss whether the vehicle will be delivered or the user will come to pick up the vehicle.

1.4.2 Limitation

- Its an online portal, so wouldn't be efficient on area where internet connectivity is a major problem.
- A mobile app supporting will be a great add on.

1.5 Development Methodology

Online Vehicle Rental System is developed under the process of agile methodology. Agile methodology focuses on obtaining functional requirements through iterations. Since this project cannot be completed in one phase, agile methodology is ideal for the development of this project.

The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins, teams' cycle through a process of planning, executing, and evaluating. Continuous collaboration is vital, both with team members and project stakeholder's deals with the introduction of the system with its objectives, scope and limitations along with the reason why the system is made.

Here are some concepts for using the agile method in this project:

- It suits small size project, with rapidly changes in the requirements as customer is involved during each phase and empowers the team.
- Very limited planning is required to get started with the project.

- Risks are minimized because of its versatile nature.
- Project are divided by short and transparent iterations. [1]



Figure 1.1: Agile Methodology of Online Vehicle Rental System

1.6 Report Organization

Introduction

This chapter deals with the introduction of the system with its objectives, scope and limitations along with the reason why the system is made.

Background Study and Literature Review

This chapter summarizes the work that has been carried out in the field of mask detection and also describes the features about some existing system related to mask detection.

System Analysis and Design

This chapter focuses on the different requirement of the system, which describes about the functional, non-functional, feasibility analysis, Data Modelling, Process modelling, System design like: architectural diagram, database schema diagram and many more.

Implementation and Testing

This chapter emphasizes tools used in system development, implementing details and result of test performed.

Conclusion and Future Recommendation

This chapter highlights brief summary of lesson learnt, outcome and conclusion of the whole project and explain what have been done and what further improvements could be done.

CHAPTER 2: BACKGROUND STUDY AND LITERATURE REVIEW

2.1 Background Study

For Online Vehicle Rental System, it has been assumed that vehicle owners would be able to use the online portal along with updating their vehicle details with pictures to upload. The vehicle information entity has been designed assuming common specification titles provided in the market. The availability of the vehicle has been managed using the date & location of the vehicle recorded into the database. The search engine is accordingly designed too. Three different users are managed according to the system requirement. The entities in database are strongly related so that the data is consistent over the system. Once the vehicle is booked, it is reserved for the booked timeframe to the specific user only.

2.2 Literature Review

For this project, I researched and reviewed some of the related websites and applications. Throughout the research, we get to find out that there are very few websites or applications related to Vehicle rental websites. So, the users aren't able to get the quality services and suitable offer. If some websites are available for users, it was found the websites are looking for many features regarding the vehicles. Especially, the users are not convinced with the quality of services given to them.

After our research, it came to know that there are various kinds of websites for Vehicle Rental System like 'rentalcars', 'autoeurope', 'enterprise' etc. But however, there are certain limitations which we are trying to overcome through our project.

Rentalcars:

In Rentalcars a user must be 21 years to be able to rent a vehicle and the prices are also different for different age groups users. Rentalcars provide insurance for collision damage of vehicles. A user must pre-book the car before 2 or 3 days and can prebook for future for fair price if a user is in urgency the Rentalcars charges high change to the customer and also need to choose the location which cannot be edited without cancelling the booking [3].

Autoeurope:

In Autoeurope a user does not get refund or cannot cancel the reservation after the 48 hours of the reservation and the pricing is done with the 24-hour basis for example if a user takes

a vehicle at 4am then the user has to return the vehicle before 4 otherwise the user will be charged of another 24 hours. Here a user can only rent the car that the company owns and don't gave an opportunity to utilizes their car as it does not let other people to rent their car because it's more of a profit-oriented system rather than the service-oriented system [4].

Enterprise:

In Enterprise a user can't rent a car on an hourly basis. A user must be able to rent a car in an hourly basis as it would be a service for a user which will cost reasonable cost for user and the repeating booking of a customer will be a way for a greater success of a company in the future and the user will only be charged of an exact time, he/she uses the vehicle. Only user with 25 years age is able to rent vehicle in enterprises and targets the small space of the people which will affect the company for the wellbeing of it.

A reliable service oriented rather than maximum profit-oriented Vehicle Reservation System is proposed which will keep track of the user and the vehicle. Vehicle reservation system is a service provided by a company to the customers who are willing to rent different vehicle for their small period of use in case to pick up somebody , or for a vacation with family in a luxurious car or a small ride to the nearest hill station and many more in a affordable price and we are willing to have two platform one for an user who will rent the car or vehicle from our website and another one for a vehicle owner who will be willing to give his/her car or vehicle and earn some money as well as will utilize his/her resources when unneeded for him/her better standing in a corner rusting inside. A fair priced and better website is needed with maximum facility for a user and also the vehicle owner to get a well serviced conditioned vehicle for an user to rent and to get some extra money in a leisure time for a vehicle owner in a website [5].

CHAPTER 3: SYSTEM ANALYSIS AND DESIGN

3.1 System Analysis

3.1.1 Requirement Analysis

The requirements are to be collected before starting projects' development life cycle. To design and develop system, functional as well as non-functional requirement of the system has been studied.

Functional Requirements

These are the requirements that the website fulfills. It shows outlines of workflows performed by the website. The functional requirements of the website are to allow user to rent a vehicle at the convenience of their homes and allow owners to put up vehicles for rent.

The list of functional requirements for admin of the website include:

- a. The system should allow the system administrator to login and logout from the system.
- b. The system should allow the system administrator to block the vehicle owner.

The list of functional requirements for Vehicle Owner of the website include:

- a. The system should let Vehicle Owner to register in to the system.
- b. The system should let Vehicle Owner to login to the system.
- c. The system should let Vehicle Owner to add/register new vehicles.
- d. The system should let Vehicle Owner to delete the vehicle.
- e. The system should let Vehicle Owner to edit the vehicle details.

The list of functional requirements for Vehicle Owner of the website include:

- a. The system should let User to register in to the system.
- b. The system should let User to login to the system.
- c. The system should let User to browse or search for vehicles.
- d. The system should let User to rent or book the vehicle.
- e. The system should let User to cancel the booking.

USE CASE DIAGRAM

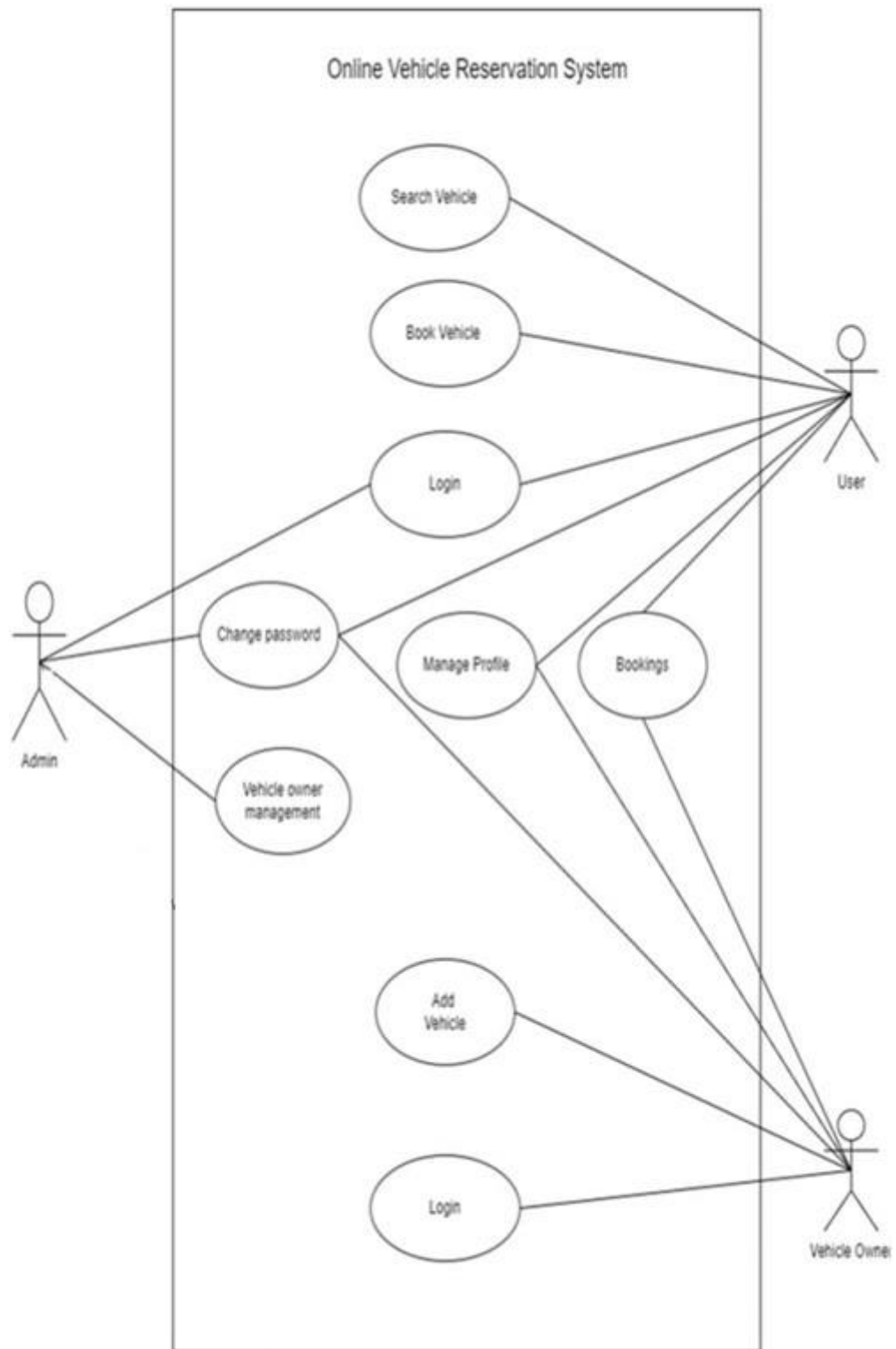


Figure 3.1: Use case diagram of Online Vehicle Rental System

Non-Functional Requirements

These requirements are the constraints or the requirement imposed on the system. Some of the lists of non-functional requirements are listed below:

a. Maintainability:

An interface should be well maintainable and should be easy to maintain when the errors arise and should be maintained by the support before the system crashes and should be able to upgradable.

b. Efficiency Requirement:

The efficiency of a software system refers to how well it handles capacity, throughput, and response time. With the deployment of the Online Vehicle Rental system, Vehicle Owners, Users and admins can be in the same portal and vehicle owner can post vehicles for renting and users can book the vehicles and the admins can manage the vehicle owners.

c. Reliability Requirement:

The degree to which the software system regularly executes the stated functions without failure is referred to as reliability. User registration, user login, owner registration, owner login, adding vehicle and booking vehicle were all performed accurately by the system.

d. Usability Requirement:

The system's usability criteria states how simple it must be to use. The system was created in a user-friendly environment so that users and administrators could easily and successfully complete various activities in the system.

e. Implementation Requirement:

The process of turning strategies and plans into actions in order to achieve strategic objectives and goals is known as implementation. The frontend is created using HTML, CSS and Vue JS, with Spring Boot serving as the server-side programming language for database connectivity at the backend, i.e., MYSQL was utilized to develop the database.

3.1.2 Feasibility Study

The feasibility study concludes that the project is able to be implemented successfully as it was carefully planned.

i. Technical feasibility study

The system is technically feasible as the requirement for the development of the system is easily accessible. The necessary hardware and software required for the development and implementation of the system is available. The basic programming language which is suitable for project is available and the libraries required for project is capable of achieving the result that we are aiming for. All the existing resources can be used for the development and maintenance system.

ii. Operational feasibility study

The system is easy to operate with the basic knowledge of computer and internet and well-trained manpower is not necessary. User can also easily access the system as it is user friendly in many aspects with good User Interface (UI).

iii. Economic feasibility study

The system is economically feasible and cost effective. Availability of a working device (desktop or laptop) and Wi-Fi would be enough to work on this as the software used is freely available.

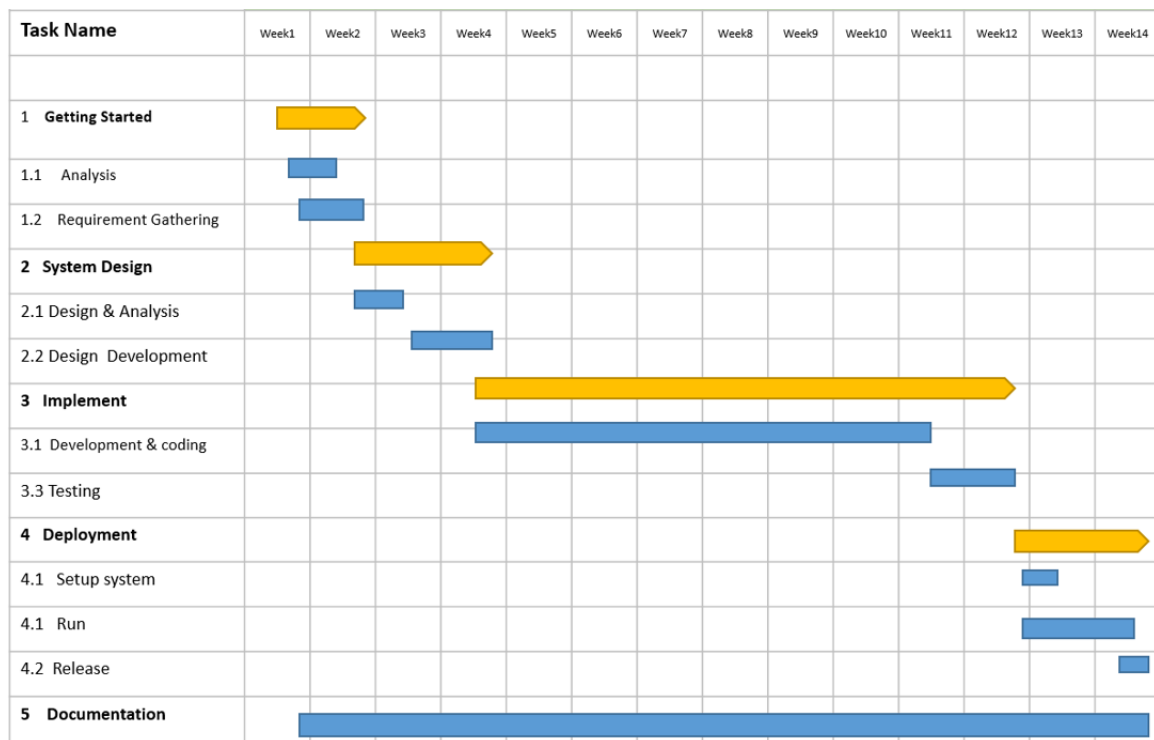


Figure 3.2: Gantt Chart of Online Vehicle Rental System

3.1.3 Object Modeling using Class and Object diagram

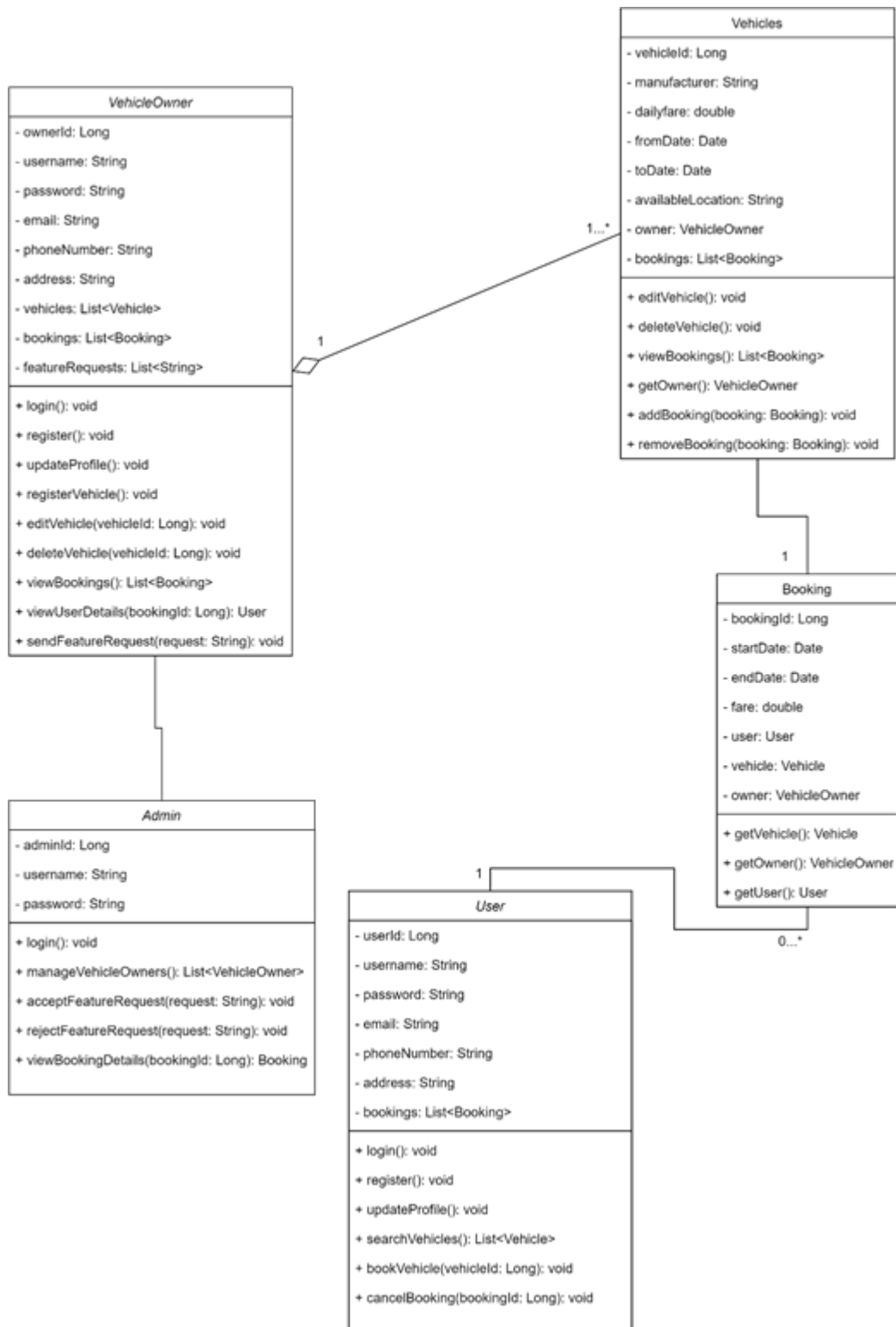


Figure 3.3: Class Diagram of Online Vehicle Rental System

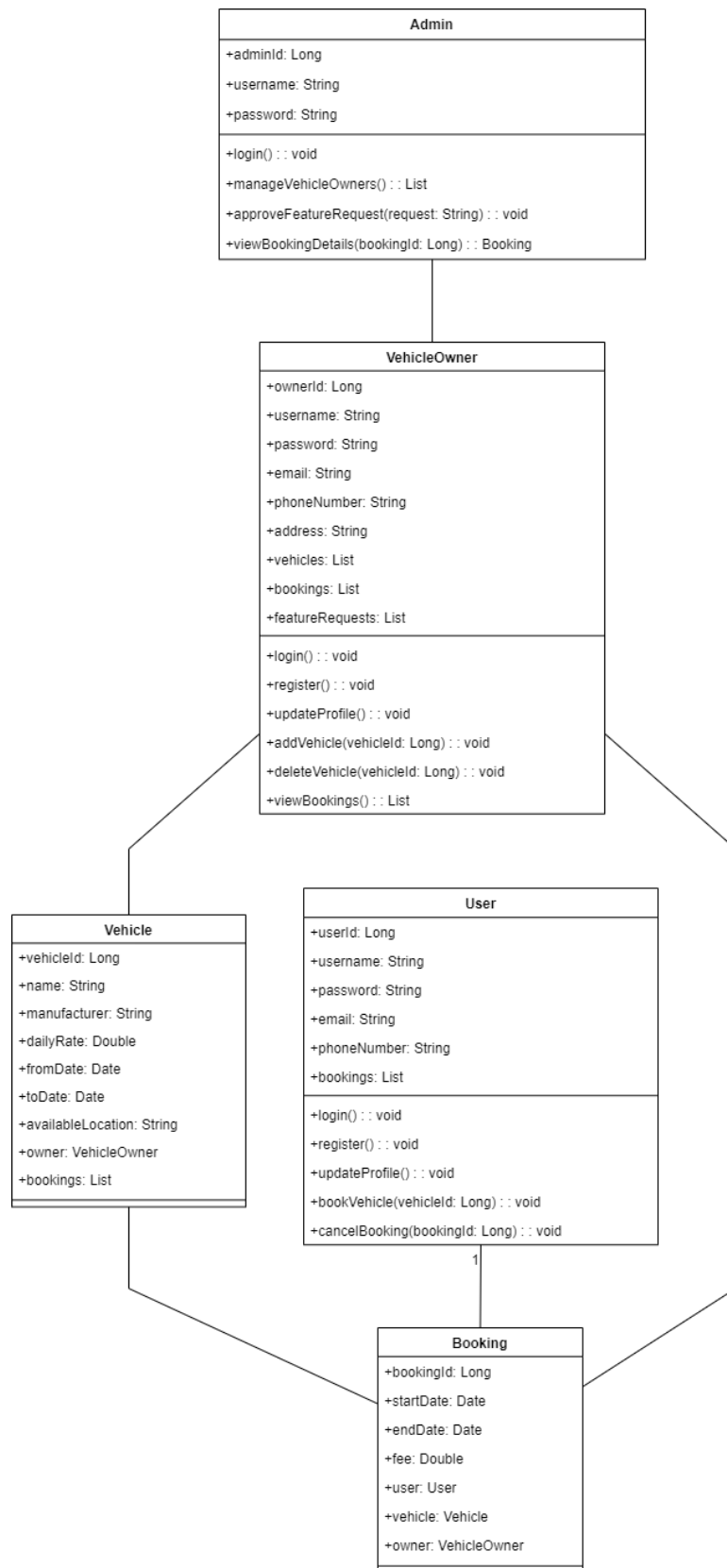


Figure 3.4: Object Diagram of Online Vehicle Rental System

3.1.4 Dynamic Modelling using State and Sequence Diagrams

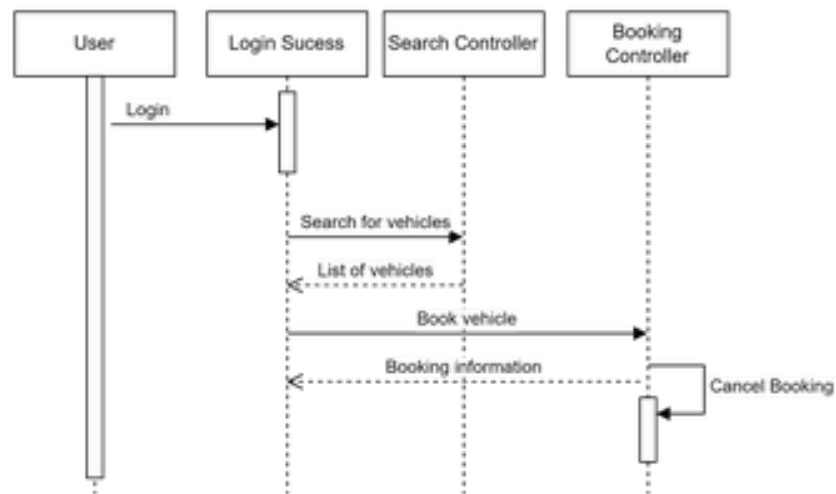


Figure 3.5: Sequence diagram of User of Online Vehicle Rental System

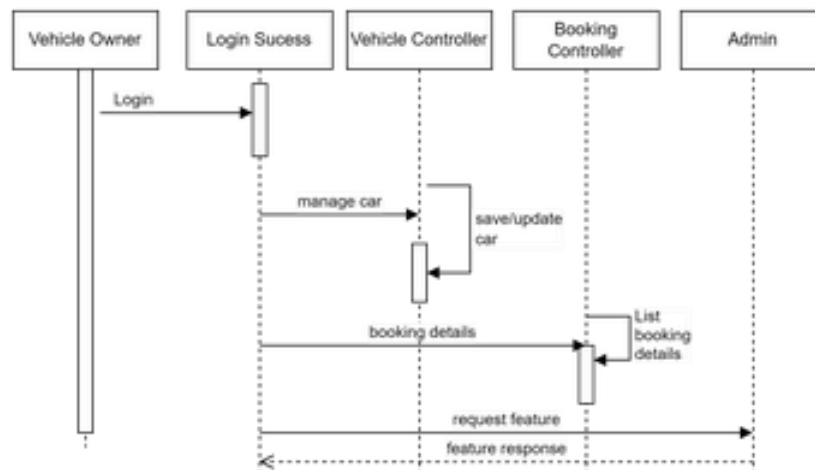


Figure 3.6: Sequence diagram of Vehicle Owner of Online Vehicle Rental System

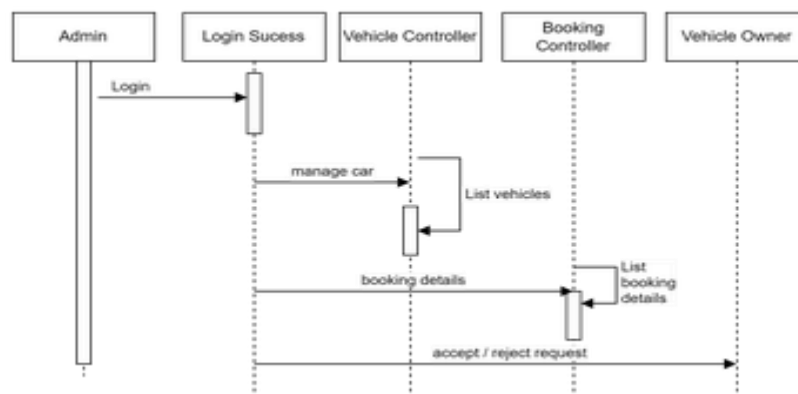


Figure 3.7: Sequence Diagram of Admin of Online Vehicle Rental System

3.1.5 Process Modelling using Activity Diagrams

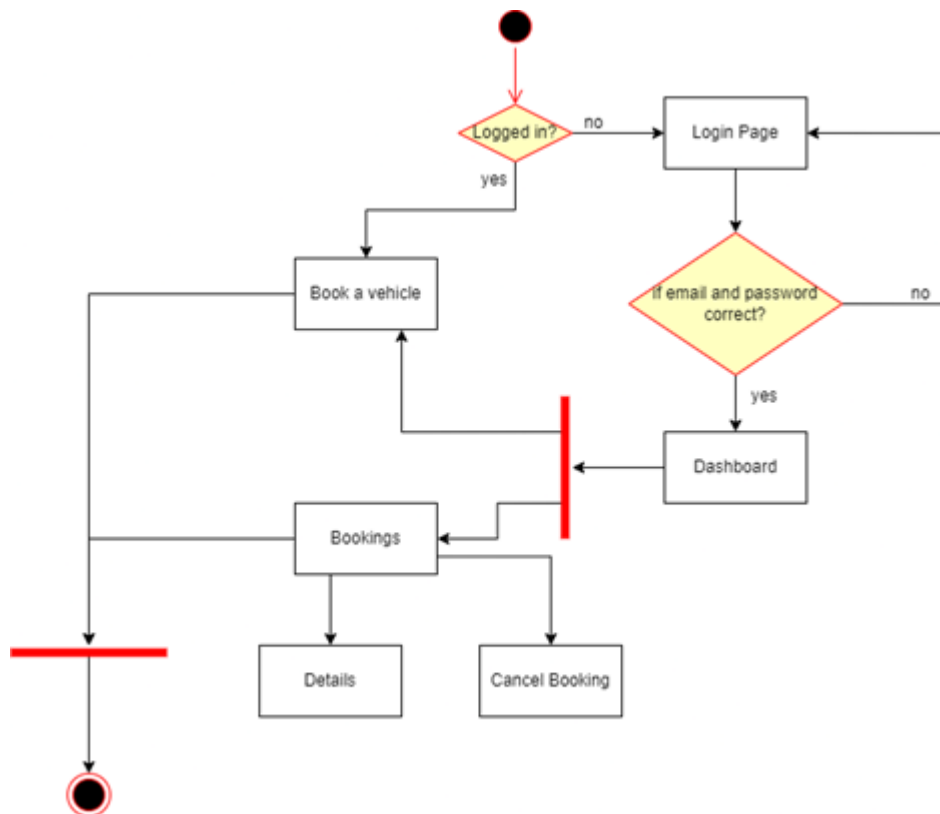


Figure 3.8: Activity diagram of User of Online Vehicle Rental System

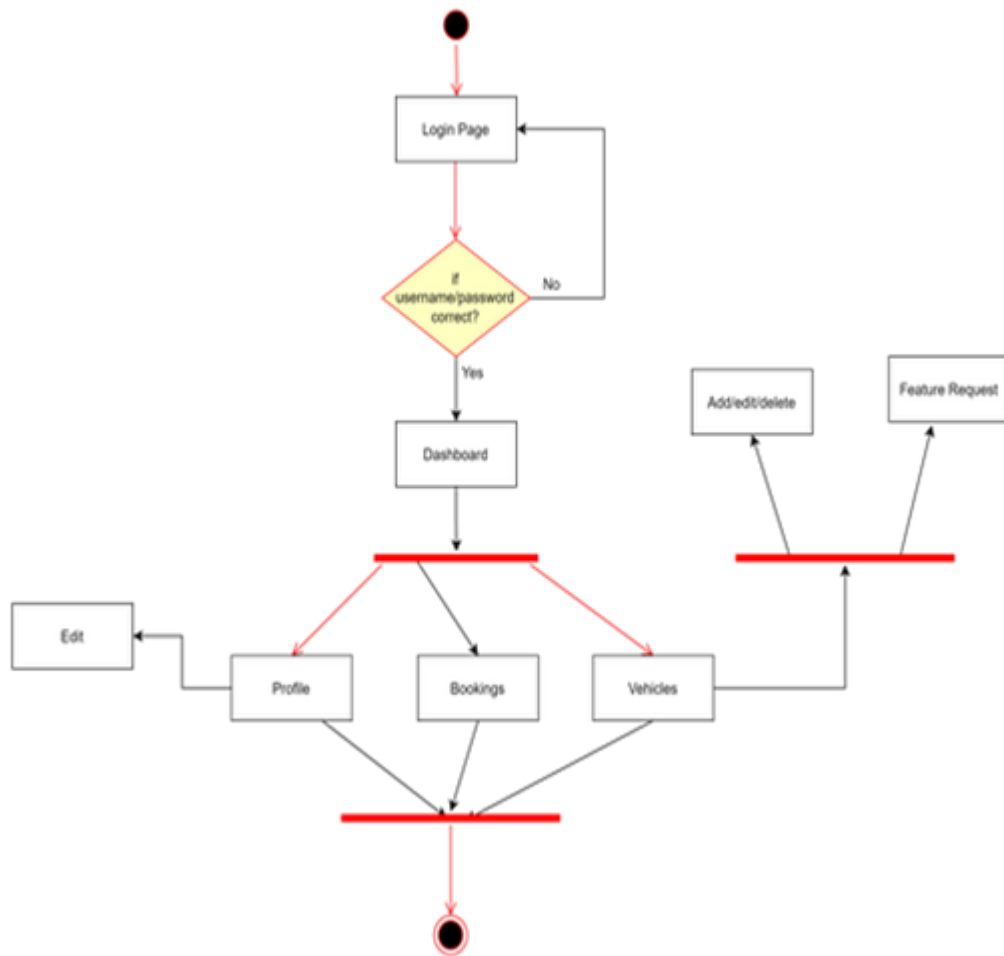


Figure 3.9: Activity diagram of Vehicle Owner of Online Vehicle Rental System

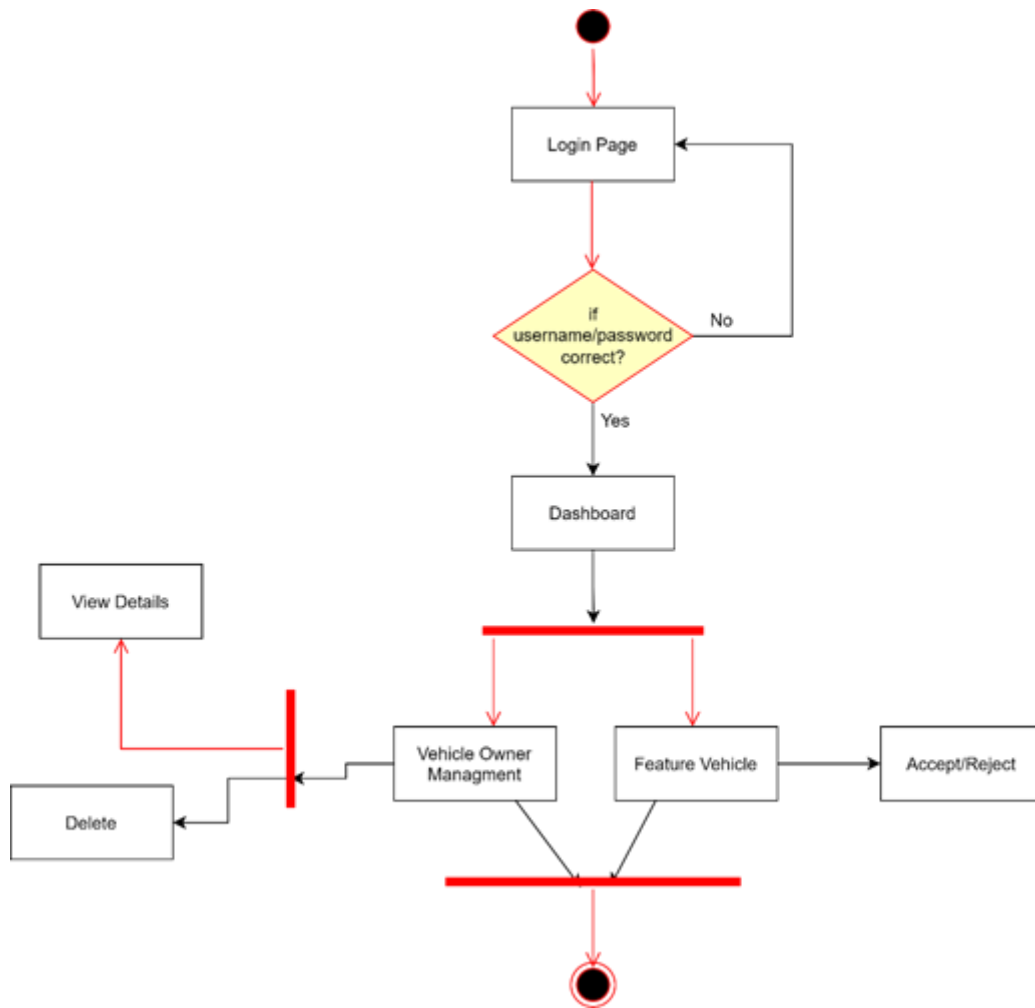


Figure 3.10: Activity Diagram of Admin of Online Vehicle Rental System

3.2 System Design

3.2.1 Refinement of Class, Object, State, Sequence and Activity diagrams

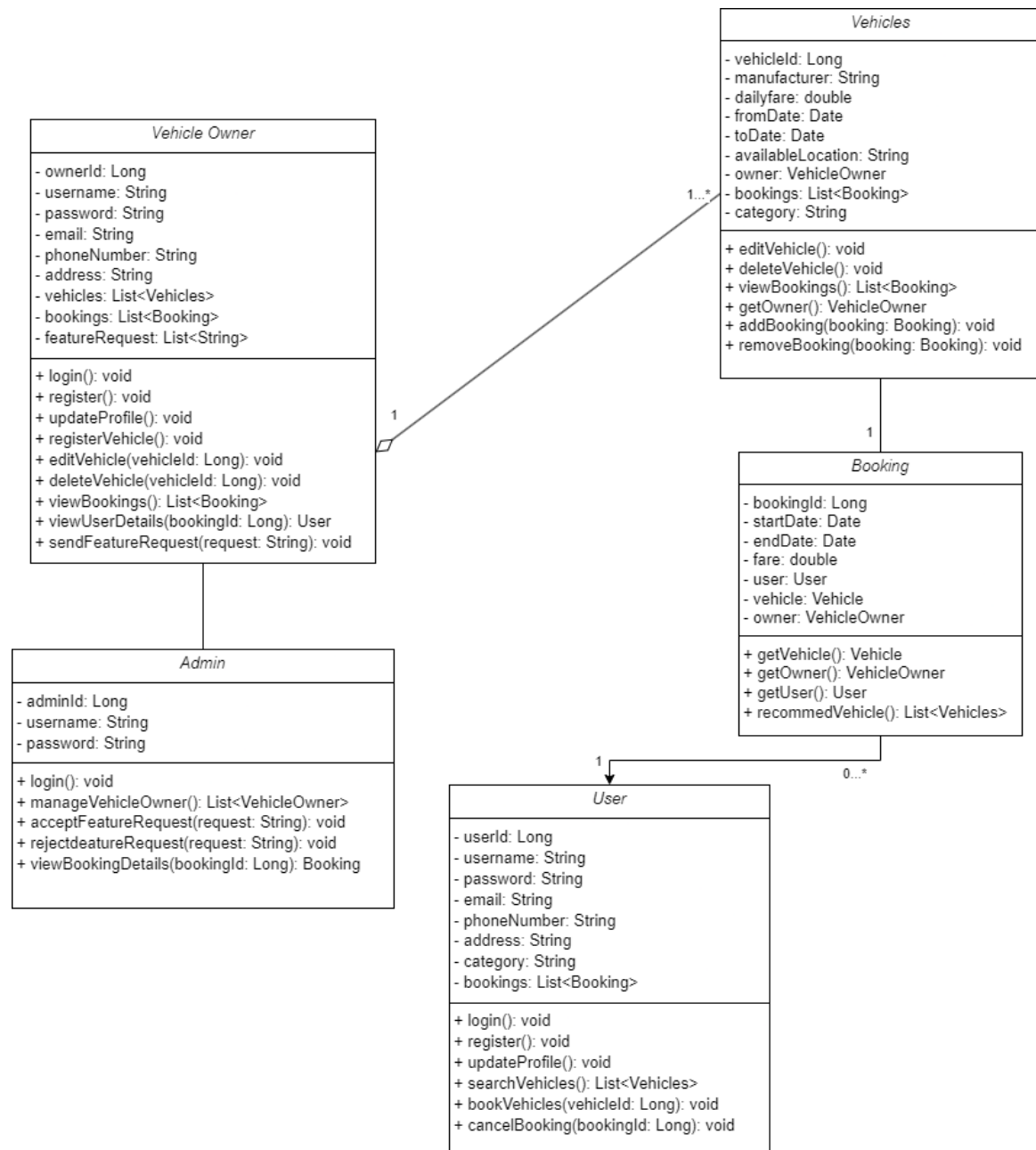


Figure 3.11: Refined Class Diagram of Online Vehicle Rental System

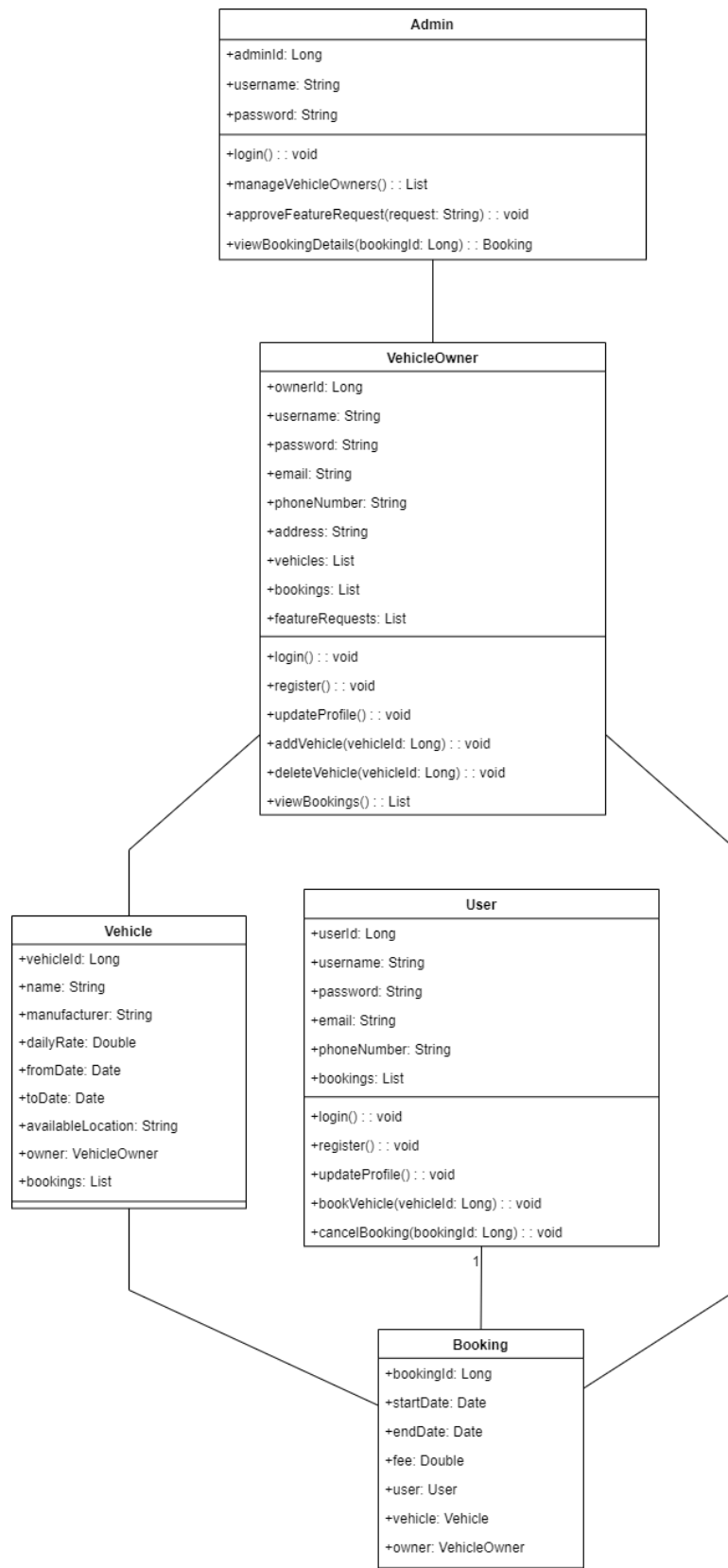


Figure 3.12: Refined Object Diagram of Online Vehicle Rental System

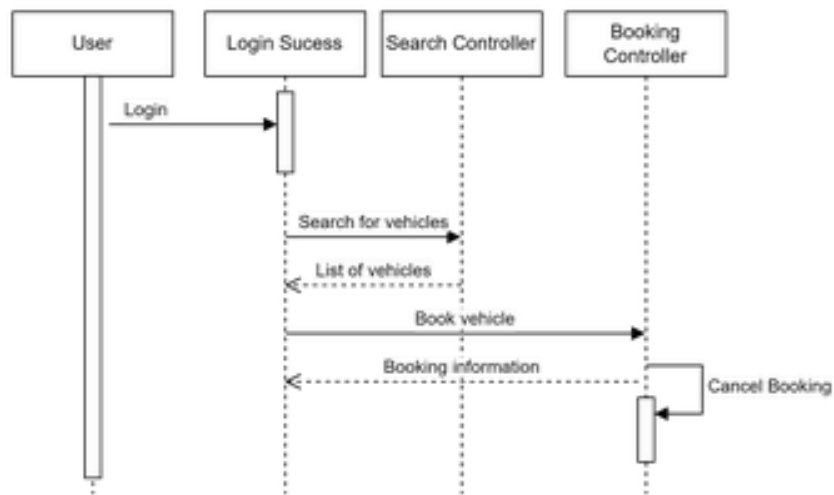


Figure 3.13: Refined Sequence diagram of User of Online Vehicle Rental System

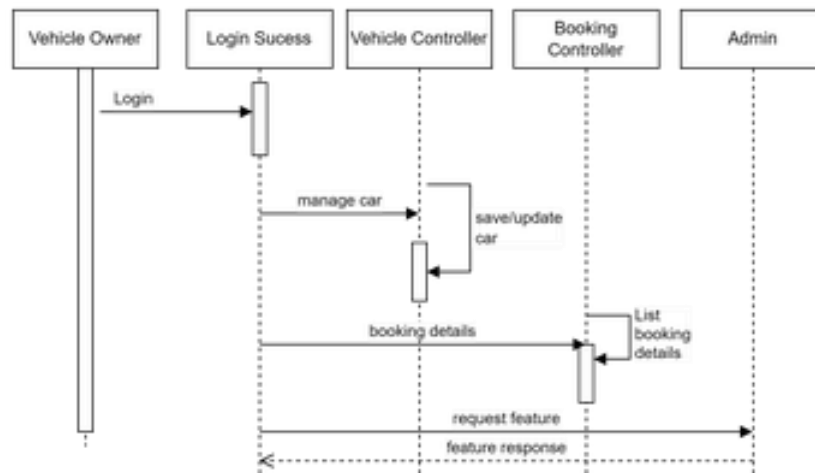


Figure 3.14: Refined Sequence diagram of Owner of Online Vehicle Rental System

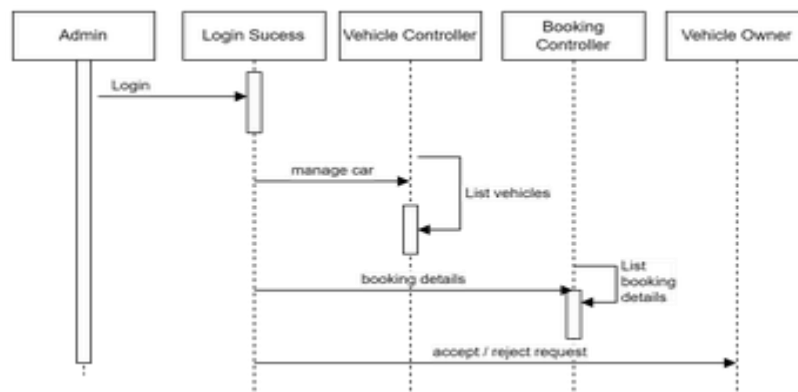


Figure 3.15: Refined Sequence diagram of Admin of Online Vehicle Rental System

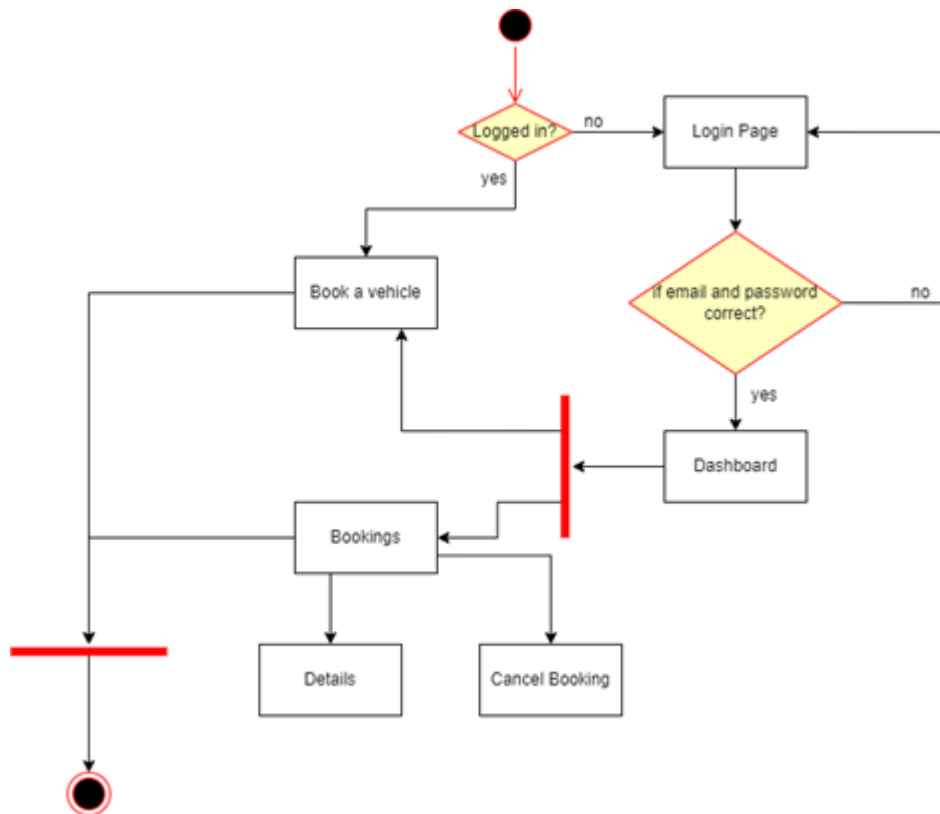


Figure 3.16: Refined Activity diagram of User of Online Vehicle Rental System

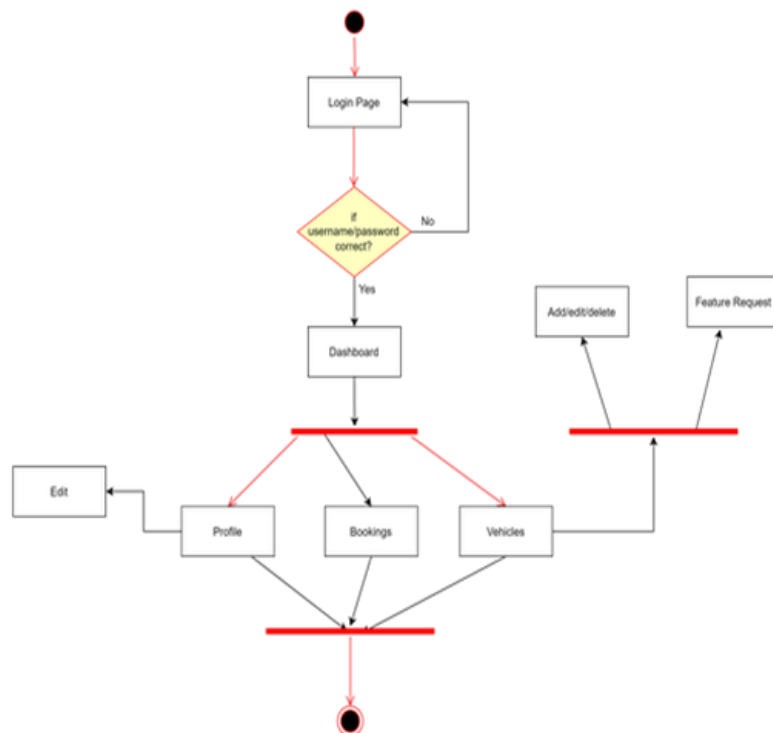


Figure 3.17: Refined Activity diagram of Vehicle Owner of Online Vehicle Rental System

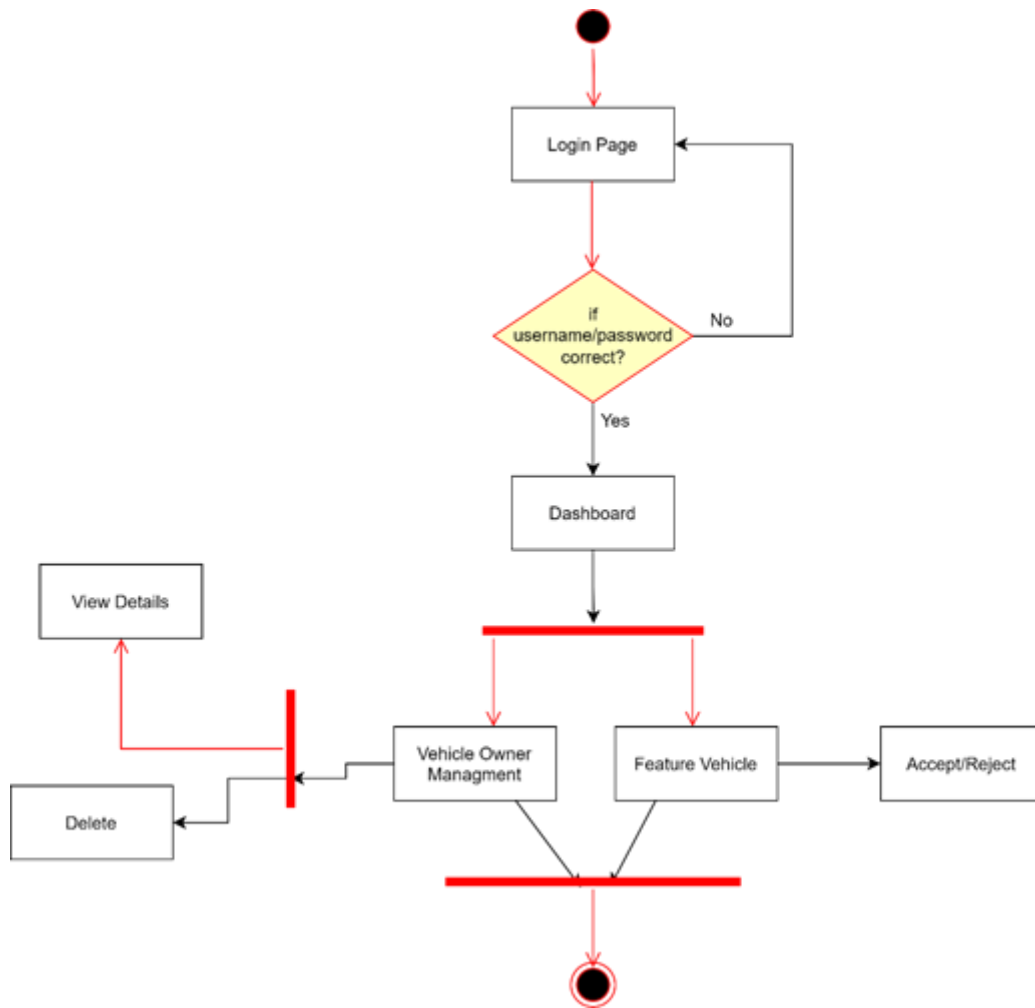


Figure 3.18: Refined Activity Diagram of Admin of Online Vehicle Rental System

3.2.2 Component Diagram

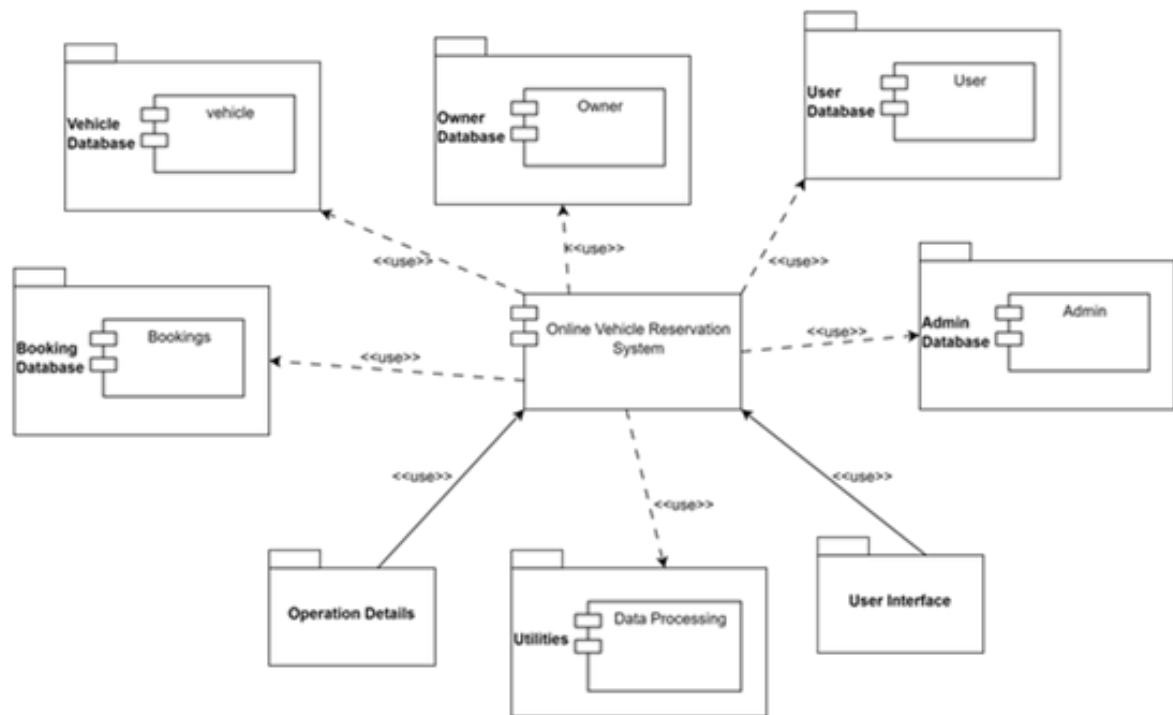


Figure 3.19: Component Diagram of Online Vehicle Rental System

3.2.3 Deployment Diagram

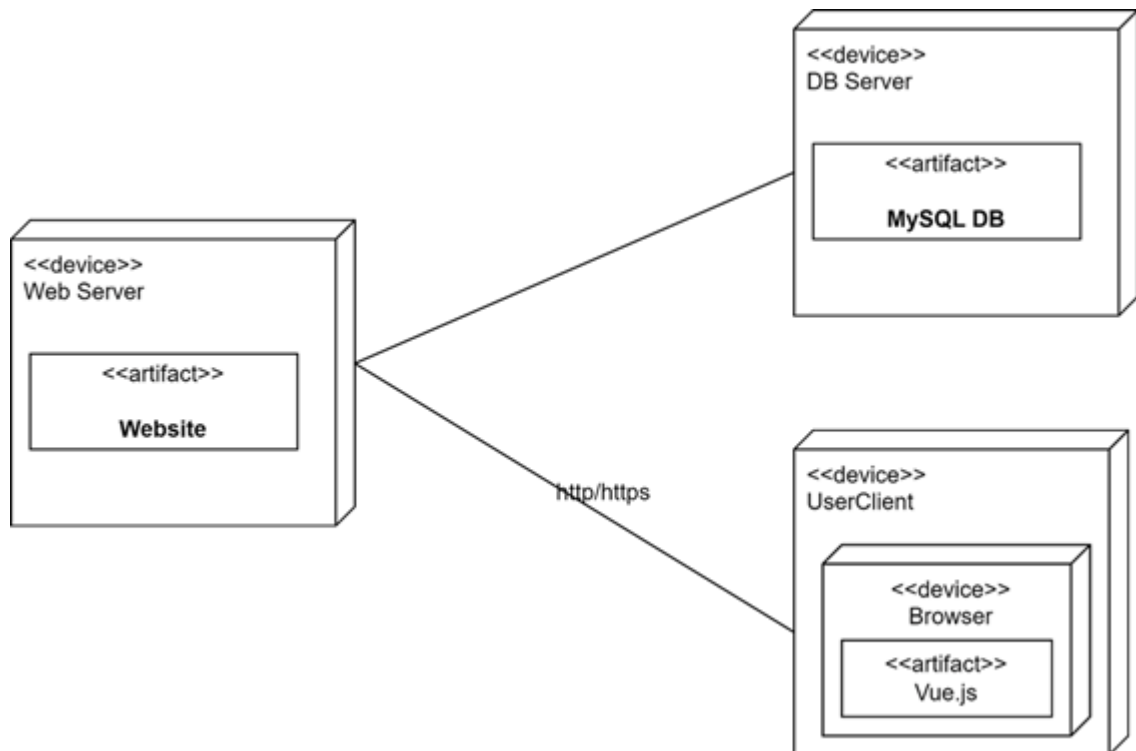


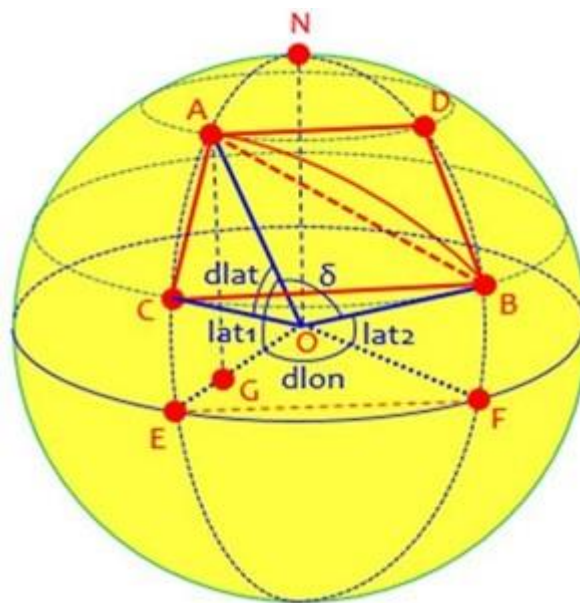
Figure 3.20: Deployment Diagram of Online Vehicle Rental System

3.3 Algorithm

3.3.1 Haversine Algorithm:

There are more than the 3 trigonometric functions we are all familiar with sine, cosine, And, tangent. These additional trigonometric functions are now obsolete, however, in the past, they were worth naming. The additional trigonometric functions are versine, haversine, coversine, Coversine, hacoversine, exsecant, and excosecant.

All of these can be expressed simply in terms of the more familiar trigonometric functions. For Example, $\text{haversine}(0) = \sin^2(0/2)$. The haversine formula is a very accurate way of computing Distances between two points on the surface of a sphere using the latitude and longitude of the Two points. The haversine formula is a re-formulation of the spherical law of cosines, but the Formulation in terms of haversines is more useful for small angles and distances.



[6]

Figure 3.21: Haversine Algorithm

Formula: In the following equation location1 is collected whenever user logs in the system and location2 is set by the vehicle owner while registering vehicle along with their latitude and longitude. Note that angles need to be in radians to pass to trig functions:

```
const a = Math.sin(dLat / 2) * Math.sin(dLat / 2) + Math.cos(lat1) * Math.cos(lat2) *  
Math.sin(dLon / 2) * Math.sin(dLon / 2);
```

```
const c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));
```

```
const distance = earthRadius * c;
```

Where,

```
const {latitude: lat1, longitude: lon1} = location1;
```

```
const {latitude: lat2, longitude: lon2} = location2;
```

```
const earthRadius = 6371;
```

```
const dLat = degToRad(lat2 - lat1)
```

```
const dLon = degToRad(lon2 - lon1)
```

After calculating the distance, the vehicles are recommended by the shortest distance.

3.3.2 Content-based filtering Algorithm:

Content-based filtering is a recommendation system algorithm used in online vehicle rental to suggest vehicles based on the content of their profiles (users) or vehicles. This algorithm focuses on finding similarities between the user's preferences and the category of registered vehicles.

How it works

In content-based filtering, each user is associated with a profile that encapsulates their preferences, and other relevant information. Similarly, every vehicle in the system is described by a set of attributes, such as category, type (e.g., sedan, SUV, truck). The recommendation process involves the following steps:

1. Input Parameters:

- a. `List<String> userCategories`: An array representing the user's profile, containing attributes such as user information and preferences.
- b. In this model, the Conv2D layers apply filters to detect features like edges, textures, and patterns at different levels of complexity.

2. Similarity Calculation:

- a. `calculateSimilarity(userCategories, vehicle.getCategory())`: Calculates similarities between two array.

3. Recommendation Output:

- a. `List<Vehicles> recommendedVehicles`: An array of vehicles recommendations sorted by similarity score in descending order.

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Implementation

Implementation basically means the phase where the system is actually being built. Firstly, all the information that we grabbed is studied and analyzed and implemented a system in operation for users. It is one of the important phases of any project. Implementation usually consists of coding, testing, installation, documentation, training and support. Different tools and technologies that have used to develop the system which are already discuss in the previous chapter. It is basically covering system design specification in to working software.

4.1.1 Tools Used

The various tools that have been used in developing both the front-end and back-end of the project are being discussed in this chapter.

Front-End Tools:

The front-end is developed using Vue JS, with a focus on creating a responsive and user-friendly interface. State management is handled efficiently with Context API to simplify the flow of data across components.

Vue JS:

Vue JS is a JavaScript framework used for building user interfaces and web applications. It is often compared to other frontend frameworks like React and Angular. Vue is known for its simplicity, flexibility, and ease of integration into existing projects. It allows developers to create reactive and dynamic user interfaces with ease.

IntelliJ IDEA:

IntelliJ IDEA is a powerful, feature-rich integrated development environment (IDE) designed for developers by JetBrains. Known for its intelligent coding assistance and developer-friendly features, IntelliJ IDEA is a go-to tool for a wide range of programming languages and frameworks, particularly Java. It provides a seamless development experience, emphasizing productivity and efficiency.

Visual Studio Code (VS Code):

VS Code is a popular and powerful source code editor developed by Microsoft. It is designed for developers and supports a wide range of programming languages, making it a versatile tool for various software development tasks. VS Code is known for its lightweight and fast performance while offering a rich set of features and extensions to enhance productivity.

Back-End:

Spring Boot:

In Online Vehicle Rental System, java programming language is used for back- end development of the project with Spring boot framework. Spring Boot is a framework built on top of the Spring Framework that simplifies and accelerates the development of production-ready Spring-based applications.

MySQL:

MYSQL is use for storing all the information required to the database in women safety alert system. It is used for performing CRUD operation such as create, delete and update data from the database as requested by the user.

Documentation Tool: Documentation and reports are prepared using Microsoft Office.

4.1.2 Implementation Details of Modules

Different modules are used for various purpose in this website.

I. Vehicle Owner Module

i. Add Vehicle category:

In this module Vehicle Owner who is also a user of the system add vehicles. The vehicle owner starts the action by clicking on Add-vehicle button where the VehicleOwner will fill up the form about the vehicle, he/she is thinking of renting in this system and once the form is filled and clicks the add button the vehicle is availablefor user to book.

II. User Module.

i. Booking Category:

User once clicks the get a quote button and then fills the information about the dateand location he/she will search the vehicle and the vehicles are

shown in user interface by the system and among the vehicles the user selects the vehicle and canbook the vehicle by simply pressing the book button.

III. Register Module:

In register module, we have implemented the Vehicle Owner register and User register as we have two types of users in this system. They can register in to the system by entering all the details such as name, email, contact and password. And then can login to the system using valid email and password.

IV. Login Module:

In login module, we have implemented Vehicle Owner login and User login as this system is the two-user based system. Both Vehicle Owner and User can enter in the system using valid username and password.

4.2 Testing

Testing is done to check the behavior of a complete and fully integrated software product based on the software requirement specification document. For the application or website to be deployed, it has to be tested. Hence, test cases will be written to test this application. There are many types of tests to be carried out in a web application, including performance, functionality, server response time, handling user actions, and more. We will not carry out all types of tests for the application considering the time scale to present this project. Hence, performance checks related to upload time, memory usage, and similar tests will be part of future evaluations. We will focus on test cases related to functionality, security, and response time to ensure the system operates as intended.

We ran tests on various browsers to ensure that the application produces consistent results and is stable on the major popular browsers. Finally, the last test will involve checking all input sources such as query strings, web services, and textboxes to help prevent cross-site scripting attacks and SQL injection.

System testing is done by giving different training and testing datasets. This test is done to evaluate whether the system is providing accurate summary or not. During the phase of the development of the system, our system is tested time and again. The series of testing conducted are as follow:

4.2.1 Test Cases for Unit Testing

System testing is done by giving different training and testing datasets. This test is done to evaluate whether the system is providing accurate summary or not. During the phase of the development of the system, our system is tested time and again. The series of testing conducted are as follow:

Admin Login Table,

Table 4.1: Test Cases for Admin Login of OVRs

S.N	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Admin enters a wrong username/password	Username: admin Password: admin	Invalid username or password	As expected	Pass
2	Admin enters correct username and password	Username: admin Password: admin	Logged in to Admin Page	As expected	Pass

Vehicle Owner Registration Table,

Table 4.2: Test Cases of Owner Registration of OVRs

S.N	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Enters invalid Name, Email, Password, Confirm-password, contact no. and click register button	Full name: Owner owner Email: owner@gmail.com Password: owner1 Confirm password: owner1 Contact no.:	Please enter your phone number	As expected	Pass

2	Enters valid Name, Email, Password, Confirm-password, contact no. and click register button	Full name: Owner owner Email: owner@gmail.com Password: owner1 Confirm password: owner1 Contact no.:1234567890	Redirect to Owner login page	As expected	Pass
---	---	---	------------------------------	-------------	------

Vehicle Owner Login Table,

Table 4.3: Test Cases for Owner Login of OVRS

S.N	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Vehicle Owner enters a wrong Email and Password	Email:own@gmail.com Password: owner5	Invalid Email or Password	As expected	Pass
2	Vehicle Owner enters correct Email and Password	Email: owner@gmail.com Password: owner1	Logged in to owner page	As expected	Pass

User Registration Table,

Table 4.4: Test Cases for User Registration of OVRS

S.N	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Enters invalid Name, Email, Password, Confirm-password, contact no. and click register button	Full name: User user Email: User@gmail.com Password: user12 Confirm password: user12 Contact no.:	Please enter your phone number	As expected	Pass

2	Enters valid Name, Email, Password, Confirm-password, contact no. and click register button	Full name: User user Email: User@gmail.com Password: user12 Confirm password: user12 Contact no.:1234567890	Redirect to Owner login page	As expected	Pass
---	---	---	------------------------------	-------------	------

User Login Table,

Table 4.5: Test Cases User Login of OVRS

S.N	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	User enters a wrong Email and Password	Email: use@gmail.com Password: user15	Invalid Email or Password	As expected	Pass
2	User enters correct Email and password	Email: user@gmail.com Password: user12	Logged in to user page	As expected	Pass

4.2.2 Test Cases for System Testing

Check system behavior,

- If the site launches properly with all the relevant pages, features and logo.
- If the user can register/login to the site.
- If the site works properly in the newest versions of all major browsers.
- If the content of pages is properly aligned, well managed and without spelling mistakes.
- If session is working as expected.
- If a user is satisfied with the site after utilizing it, or if the user does not find it difficult to utilize it.

Test Case for Add-Vehicle,

Table 4.6: Test Cases of Add vehicle of OVRS

S.N	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Vehicle Owner enters a invalid Dailyfare	Manuafacturer:TATA Model:Nexon Daily fare:aaa From Date:05/10/2022 To Date:05/17/2022 Vehicle Current Location:BKT Description:	Please enter number	As expected	Pass
2	Vehicle Owner enters a valid Daily fare	Manuafacturer:TATA Model:Nexon Daily fare:100 From Date:05/10/2022 To Date:05/17/2022 Vehicle Current Location:BKT Description:	Redirent to Vehicle Management page and the added vehicle is displays	As expected	Pass

Test Case for Vehicle Deletion,

Table 4.7: Test Cases of vehicle deletion of OVRS

S.N	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	Vehicle Owner tries to delete an unbooked vehicle	View the details of vehicle and click on delete button	Vehicle get deleted and redirect to vehicle management page	As expected	Pass
2	Vehicle Owner tries to delete an booked vehicle	View the details of vehicle and click on delete button	Displays message “Vehicle already booked. Sorry, cannot delete vehicle.”	As expected	Pass

Test Case for Search vehicle,

Table 4.8: Test Cases Search Vehicle of OVRS

S.N	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	User searches for vehicle without logging in	View search results and click on book this car button.	Redirects to user login page	As expected	Pass
2	Logged in User searches for vehicle	View search results and click on book this car button.	Proceed to Booking Details page	As expected	Pass

Test Case for Book vehicle,

Table 4.9: Test Cases of Book vehicle of OVRS

S.N	Test Case Description	Test Data	Expected Result	Actual Result	Pass/Fail
1	User selects a vehicle from the search results	From Booking Details page user selects the confirm booking button	Booking confirmed message.	As expected	Pass

4.3 Result Analysis

The test results show that the individual modules operate properly each of them efficiently carrying out their allocated operation. These modules independently operate without any chance of error. The implementation of algorithms, such as the Haversine algorithm for location-based distance calculation and the content-based filtering algorithm for personalized recommendations, enhances the overall functionality of the system.

The Haversine algorithm accurately computes distances between the user's location and available rental vehicles, ensuring that location-based searches return precise results. Similarly, the content-based filtering algorithm delivers personalized vehicle recommendations by evaluating user preferences and comparing them with vehicle attributes. This approach provides users with highly relevant suggestions, improving the rental experience.

Overall, the combination of the Haversine algorithm for distance-based searches and the content-based filtering algorithm for recommendations ensures that the system meets user expectations. While there is room for further enhancements to make the recommendation engine more robust, the system as a whole is functional, reliable, and provides a satisfying user experience.

CHAPTER 5: CONCLUSION

5.1 Conclusion

Hence, “Online Vehicle Rental System” will be a very useful online portal for reserving vehicles for self-driving. It will also help the vehicle owners easily rent their vehicle without caring of the third party in between and directly co-ordinate with the person who will be driving the vehicle.

Since the application is online & totally web based, it will cost less for configuring a system to make the project up and running.

5.2 Future Recommendations

In the future this app will be helpful for all humans. The future plans for this app are as follows:

- Remove bug of users facing website crashes while using it.
- OTP (One Time Password) feature can be added to verify the email address.
- Online Payment functionality will be a great add-on.

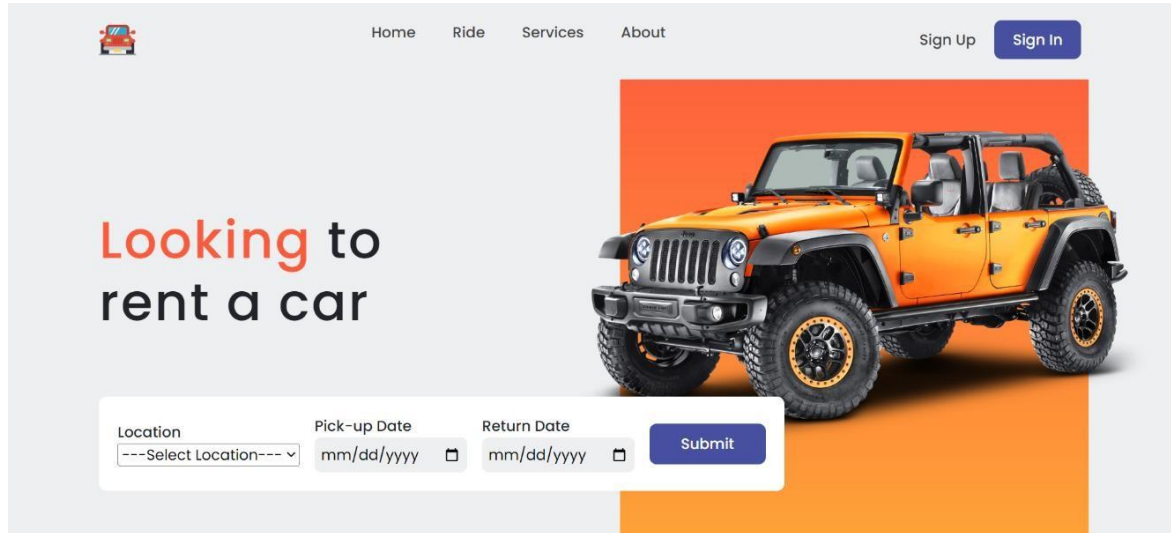
I will tie up with the communication and it will be easy for the users. It will be easy to track vehicle location easily. I will try to implement these things in future. I will focus on security issues and helps to solve security issue. This website will be user friendly to all platforms.

References

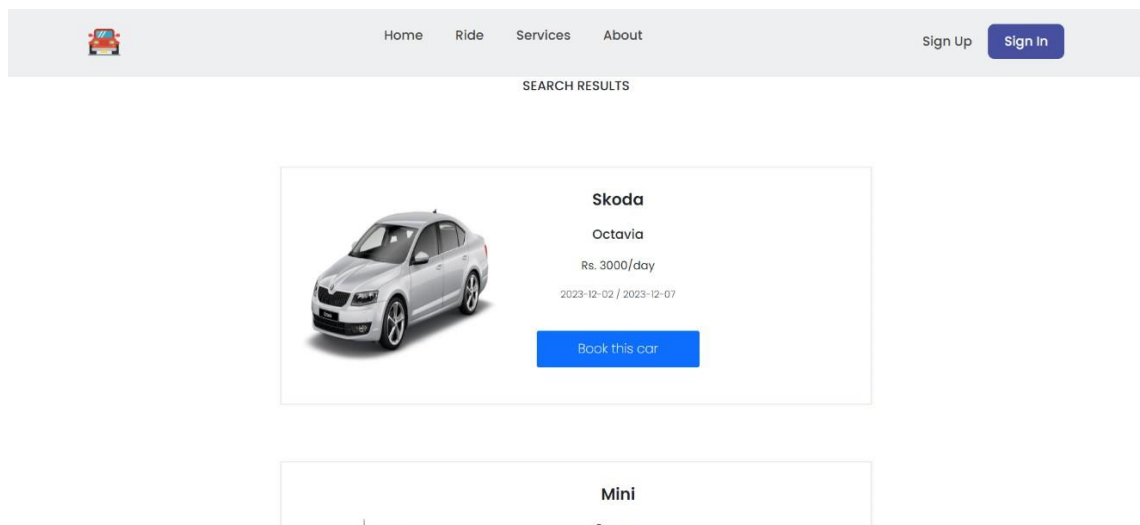
- [1] rajkumarupadhyay515, "GeeksforGeeks," 22 May 2021. [Online]. Available: <https://www.geeksforgeeks.org/agile-methodology-advantages-and-disadvantages>. [Accessed 12 March 2023].
- [2] Team, "Skylark," 04 January 2021. [Online]. Available: https://www.researchgate.net/figure/Agile-Methodology-in-System-Development-source-Okeke2021-retrieved-from_fig1_354310848. [Accessed 12 March 2023].
- [3] rentalcars, "rentalcars," 16 February 2015. [Online]. Available: <https://www.rentalcars.com/>. [Accessed 14 March 2022].
- [4] autoeurope, "autoeurope," 4 February 2013. [Online]. Available: <https://www.autoeurope.com/about-us/>. [Accessed 9 March 2022].
- [5] enterprise, "enterprise," 20 December 2012. [Online]. Available: <https://www.enterprise.com/en/about.html>. [Accessed 24 March 2022].
- [6] D. Peterson, "themathteachers," 23 March 2023. [Online]. Available: <https://www.themathteachers.org/distances-on-earth-2-the-haversine-formula/>. [Accessed 16 October 2023].

APPENDIX

- Home Page(index)



- Vehicle search result



- User Dashboard

Online Vehicle Reservation

Dashboard

Profile

Bookings

User Profile

Name:

Janam Sunuwar

Email:

janam@gmail.com

Address:

Sinamangal

Contact:

9845033112

Lisence:

DRIVER LICENSE

ID 00000

Surname
LOREM

- Admin Dashboard

ONLINE VEHICLE RESERVATION



Dashboard

Owner Management

Feature Requests

Logout

Feature Request

SN	Image	Manufacturer	Model	Rate	From	To	Location	Feature	Actions
1		Skoda	Octavia	Rs. 3000/day	2023-12-02	2023-12-07	Kathmandu	<div>Accept</div> <div>Reject</div>	<div></div> <div></div>
2		Toyota	Yaris	Rs. 1200/day	2023-12-02	2023-12-13	Bhaktapur	<div>Accept</div> <div>Reject</div>	<div></div> <div></div>

Owner Dashboard

Owners Dashboard

Dashboard

Profile



Vehicles

Bookings


Miraj Shrestha

Vehicles

Add vehicle

SN	Image	Manufacturer	Model	Rate	From	To	Location	Feature	Actions
1		Skoda	Octavia	Rs. 3000/day	2023-12-02	2023-12-07	Kathmandu	Request	Edit Delete
2		Mini	Cooper	Rs. 1500/day	2023-12-02	2023-12-19	Makawanpur	Request	Edit Delete


- Featured Vehicle



[Home](#)[Ride](#)[Services](#)[About](#)

[Sign Up](#)[Sign In](#)

FEATURED VEHICLES




Octavia

Kathmandu

Rs. 3000 /day

Rent Now




Cooper

Makawanpur

Rs. 1500 /day

Rent Now



Yaris


Bhaktapur

Rs. 1200 /day

Rent Now

- Vehicle Recommended by Distance

NEARBY VEHICLES




Skoda

Kathmandu

Rs. 3000 /day

Rent Now




Toyota

Bhaktapur

Rs. 1200 /day

Rent Now



Mini

Makawanpur

Rs. 1500 /day

Rent Now