Software Requirements Specification

Version 1.0

for

SOEN-6461-Team 11

Prepared by

Anusha Keralapura Thandavamurthy	40102962	kt.anusha21@gmail.com
Arvind Korchibettu Adiga	40105178	adiga1993@gmail.com
Basant Gera	40082433	basantgera29@gmail.com
Koteswara Rao Panchumarthy	40084998	kotichowdary18@gmail.com
Sai Charan Duduka	40103928	charan140494@gmail.com
Sourabh Rajeev Badagandi	40098471	sourabh.rajeev@gmail.com

Instructor: Dr. C. Constantinides

Course: SOEN-6461

Date: 03-12-2019

Document history

Date	Version	Description	Author
26- September-2019	1.0	Writing the software requirement based on Clerk Functionality.	Basant Gera
21- October -2019	1.1	Changes done in iteration 2 requirements & added requirement for iteration 3	Basant Gera
27-October-2019	1.2	Changed the document to follow Requirement Analysis Standard.	Sourabh Rajeev Badagandi
05-November-2019	1.3	Added domain model	Sourabh Rajeev Badagandi
15-November-2019	1.4	Added final changes according to persistence and concurrency,Made System sequence diagram with team,Updated Operation contracts.	Basant Gera
17-November-2019	1.5	Updated Non functional requirements model with Persistence and Concurrency identifiers, added operational contracts for client record management. Added non functional requirements coverage matrix. Reformatted and prepared final draft.	Sourabh Rajeev Badagandi

Table of contents

List of figures

1. Introduction

The following document tells us about the nature of the project or software application.The document consist of some artifacts which will help our team to kick start on the web application (Vehicle Management System) which are trying to build. We are following a set of guidelines which consists of Functional and No functional requirements towards which we are trying to build a sustainable system which is capable of booking vehicle for clients with their specific requirements.

The Vehicle Management System which we are trying to make will be solving the capabilities which most often website fails to address or forgot to think.We have divided to do this project into 4 iterations which will be updated sequentially in SRS document while the works go on.The detailed requirements of **Vehicle Management system** will be listed in this document which concentrates on features and capabilities.

Purpose

The purpose of the document is to understand and present artifacts which will help developers what Vehicle Management system requirements are and what is it capable of doing.The essential things about this document is to collect and analyze all sorts of ideas which comes to mind and to define the system w.r.t our developer ,designer and user. In this document we shall predict and sort out how the product (Vehicle Management System : Web Application) made based on scenarios and outline the concepts which can be developed later.The ideas which are described here may be changed or discarded as product develops.

In brief, the SRS document will provide you the details overview of software with respective parameters and goal which will be listed in this document.It also define how our users, clerk,administrator and target audience look and address how application would work and respond.Never the less it will help designer and developer to build the system efficiently and resourcefully as per the requirements stated to us.

Scope

Primarily, the scope come to out to be helping clients who wanted to book vehicles to travel in day today life. It focuses mainly on how sale and distribution of vehicles has been done over time with the ease of renting, reserving and returning the vehicles and their management respectively by users.At the same time a vehicle can be taken for rental and reservation too.

The SRS documents is also entitled to tell you specific requirements of software which tells you the small scenarios which are called functionality we are trying to achieve.

Definitions, acronyms, and abbreviations

SRS	Software Requirements Specification

?param	Parameters
VMS	Vehicle Management System
CRMS	Customer related management software
GPL	General public Licence
ORM	Object Relational Mapping

References

2. Overall description

The following SRS documents contains the detailed Web application (Vehicle Management system) perspective to the users.It will give you the detailed description how a functionality should be made from top level and what are the expectations of the stakeholder to developer and designer.We would be following agile methodology in which 4 iterations would be carried so that we can confirm whether every build is according to stakeholder requirements or not.

The following are scenarios which are listed according to which (Vehicle Management system) should work :

Iteration :1

Setting up Environment based team unanimous decision.

Iteration :2 Clerk Functionality

- **Register** a clerk/administrator based on checkbox.[Checking up on checkbox gives user to make a username and password which will work as administrator.]
- **Login** for registered clerk/administrator. Since clerk/administrator both has different landing pages after login.
- **Vehicle Catalog page** is made for clerk which will filter the the vehicle information based on 4 ?parms which are Vehicle Make, Vehicle Type, Vehicle year and Vehicle Model.The functionality consists of the following things which are as follows :
    - Filtering out or searching based on by Vehicle Make, Vehicle Type, Vehicle year and Vehicle Model.
    - Sorting Ascending/Descending based on Vehicle Make, Vehicle Type, Vehicle year and Vehicle Model [a-z] [z-a] [1-highest] [highest-1] (In case of year).
- **Detailed View Page** is made for clerk to check all the details which are booked by clerk buy asking client are ok.Clerk can book on rental book now for rental or he/she can click on reservation book now for booking a reservation. .It also has the following functionality which are as follows :
    - It allows clerk to click on the next button to view new car with their respective details.
    - Clerk can click on 2 buttons book now buttons which generally has the following functionality which are as follows :
        - Rental Book Now : To book the vehicle ASAP and start date would be current date.
        - Reservation Book Now : To book a vehicle for future start date and end date.
- **Booking form Page** is made for the functionality which can ask the clerk to fill the first name , last name , Phone number and Dates such as Start date ,end date and license expired date with all validation to book the vehicle for current client via clerks.

  - ❖ Validation such as license no should not be expired or greater than vehicle booking end date.
  - ❖ Start date of vehicle registration should not be greater than vehicle end date.
- ❖ **Client Manager** to help clerk manage all clients, supported functions are:
  - ➢ Modify client details
  - ➢ Delete client
  - ➢ Cancel a reservation
  - ➢ Handle return of vehicle

Iteration :3 Administrator Functionality

- ❖ **Vehicle Master : The Screen consist of following things :**
  - ❖ Filtering out or searching based on by Vehicle Make, Vehicle Type, Vehicle year and Vehicle Model.
  - ❖ Add the vehicle record.
  - ❖ Delete the vehicle record.
  - ❖ Edit the vehicle record.
- ❖ **Transaction recording Screen :** The screen captures various things which are as follows **:**

  - ❖ Any modify/ Cancel/return history of vehicle is recorded and updated on the screen.
  - ❖ Filtering based on particular criteria based on search button.
    - ❑ **Client First Name.**
    - ❑ **Client Last Name.**
    - ❑ **Vehicle Model.**
    - ❑ **Vehicle Type.**
    - ❑ **Due Date of vehicle.**
    - ❑ **Start Date of vehicle.**
- ❖ **Security rights given based on user group :**
  - ➢ **Administrator** duty is to only add /edit/ Delete the vehicles apart from that administrator can check transactions and see the vehicle catalog but can't not book vehicle for any clerk.
  - ➢ **Clerk** can book vehicle for the client as per dates clients want and can not view the Vehicle master page and don't have sufficient rights to view the page.

Iteration :4 Persistence and Concurrency

- ❖ **Persistence is about** Storing of working memory into the database is followed through object relational mapping .

Now the question comes how the VMS (Our renting web application when connected with Database retain data throughout the application).For the following things we are using ORM to achieve persistence which is explained below.

- **Object Relational Mapping (ORM):** We have created datamapper class where the working memory is stored and data gateway class for database programming to make connections with database.All DML commands (Select,Update,Delete,Insert) are generally stored in a batch file and executed when the operation or process is completed to get the result set and shows wherever required.Below is the diagram which shows how ORM generally works.

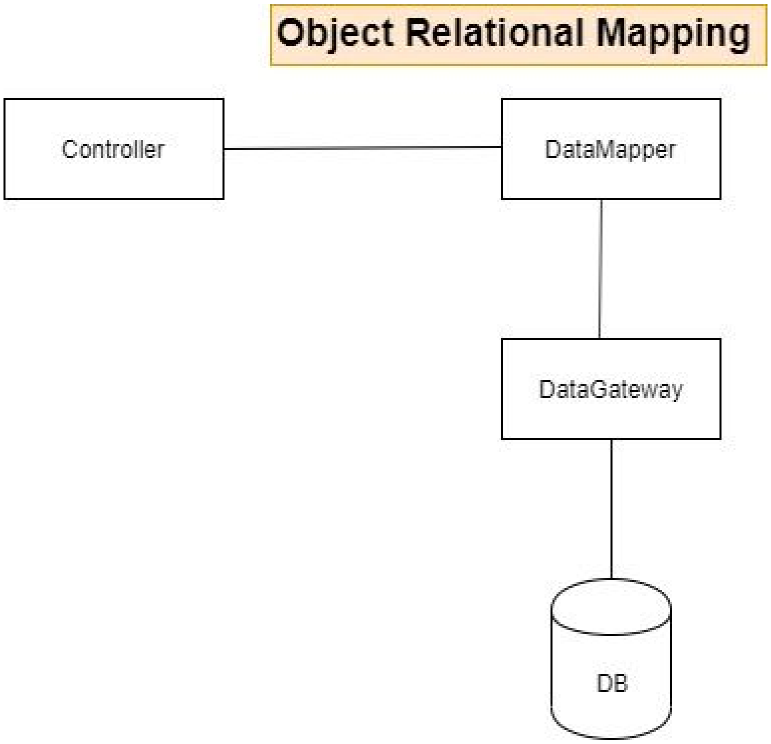


Figure : 1

Every Controller will call Respective datamapper class in which will call Data Gateway in which queries are written.

Following are 4 tables which will be associated with database to attain persistence which are as follows:

1. **Clerks Table:**Clerk working data is first stored in the ClerksDataMapper Class and then passed it to ClerksDataway Class where data connection is opened and stored in clerks table .Clerk Table needs to be mapped with the respective model class to attain persistence.
2. **Vehicles Table :** Vehicles table is used to store the vehicle data when admin tried to add the vehicle record and consists of booking id which is updated when the vehicle is

booked/returned/cancelled.Vehicles Table needs to be mapped with the respective model class to attain persistence.

3. **Bookings Table** : Booking data is stored with booking time stamp and return date and cancel date is updated when the vehicle is returned or cancelled.Booking table needs to be mapped with the respective model class to attain persistence.
4. **Clients Table** : When the vehicle is booked client data is inserted into the table and vehicle ID is updated from the vehicle data.Clients Table needs to be mapped with the respective model class to attain persistence.

   Remember : Make sure very table follows identity pattern to have a uniques auto increment ID which would be the same in the model class also.

Concurrency

What happens when two users trying to access the same resources which is not taken by anyone.The above situation is called race condition.Which will be faced when someone is trying to book the same vehicle at the same time. The process needs to be handled by the developers based on following situations which are written below.

Roles/ operation in concurrency which are defined below :

Writer / Write operation : A write operation is when someone modifies the shared resource content.

Reader / Read operation : A read operation is something which doesn't modify but try to access the content of shared resource.

Requirements need to be achieved:

- **Restrict the application for administrator** : Only one administrator should be able to login into the system at a time .Administrator can do read/write in the application.

- **Restrict the application for clerk :** The system performance should be handled when multiple clients logging into system through different browsers and performing different tasks to make the system on sync for multiple clients.

  Remember : This can be achieved through data storing in session variables(which is different for each client) when the client login into the system.

  The below situation can be handled by various methods which are as follows :

- By applying optismitics locks which has various states like Inuse/ Release. (***Good Method to handle concurrency***).Mostly preferred in Industry.
- By maintaining the versions of each record through database.
- By applying Pestismitics locks via making synchronized keywords.

Situation Needs to be handled developers which are as follows :

S.No	Situations	Mitigating factors
1	Two admins should not be able to login into the system at a time.	Can be done by making one static variable for admin login. So only one admin can log in the VMS application
2	Multiple clerks should be login into the system and perform tasks	Multiple login of clerks can be done via storing in logging into session.Multiple Clerks can't rent same vehicle for clients at the same time.
3	When admin is trying to edit vehicle record clerk should be able to read the data	When modifying a vehicle record via admin multiple clerks can able to view the updated records (Able to read it)
4	When two clients are trying to book vehicle they should be in waiting list and fairness should be handled	When two clients when accessing and trying to book the same vehicle record.One is able to rent a vehicle and other will go into the waiting state.
5	Priority of shared resources should be given to the client who is trying to edit	Priority would be given to writer/ over reader.At the same time if two clients open the records in update mode the ones who click book now will able to update the record and others will be assigned in Waiting state.(Try again)
6	Shared resource access should not be simultaneously by more than one client	The Client who is able to access the resources first will able to get access to it and other will be given 2nd

		priority.

The general factors that affect the product (Web Application: Vehicle management system) based on given scenarios are as follows :

Product perspective

The overall picture of the product is self-owned and self-contained by the company. The vehicle that the company owns has the sole purpose to give vehicles on the rent to the user based on certain requirements that may be asked by the user. The product (VMS) has a set of teams and set of departments that have different functionality on which they are trained such as user group for e.g Clerk, Administrator.

Following is the overall picture been shown to the Designer and developer for VMS.

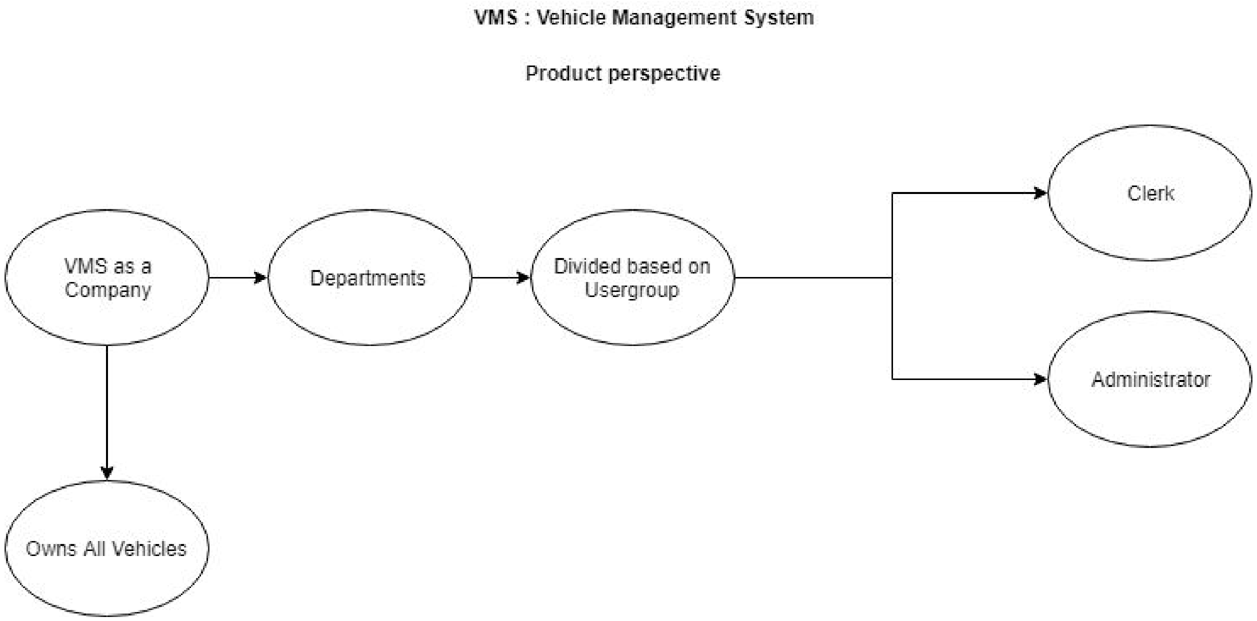


Figure : 2.1

Product functions

The product VMS should functions are as follows :

Since every functionality varies from the clerk to Administrator so we have decided to make 2 user groups which are 1. clerk 2.Administrator.

1. It should be able to register and login.
2. If successfully logged in then the clerk should be able to give ?param such as type,make,year,model which he wants to rent a vehicle and sort according what he wants like (ascending to descending in terms of alphabets and numbers).And can open the details and can book the booking based on whether client want it as Rental or reservation and click on respective book now button and fill clients name and Clicking on Booking.
3. Clerk can able to search how many available vehicles are available and unavailable based on that a customer can modify his information if he wants to modify the reservation date and end date.
4. Administrator at the same time can filter information and Can add,modify and delete the information of the vehicles.
5. Administrator can view transactions screens and filter based on Clients,vehicle , due and start date.
6. Administrator can add vehicle but cannot book the vehicle for any client whereas clerk doesn't have the right to view Vehicle master screen where vehicles are added and transactions are shown.

Figure : 2.2 below shows how VMS should perform.

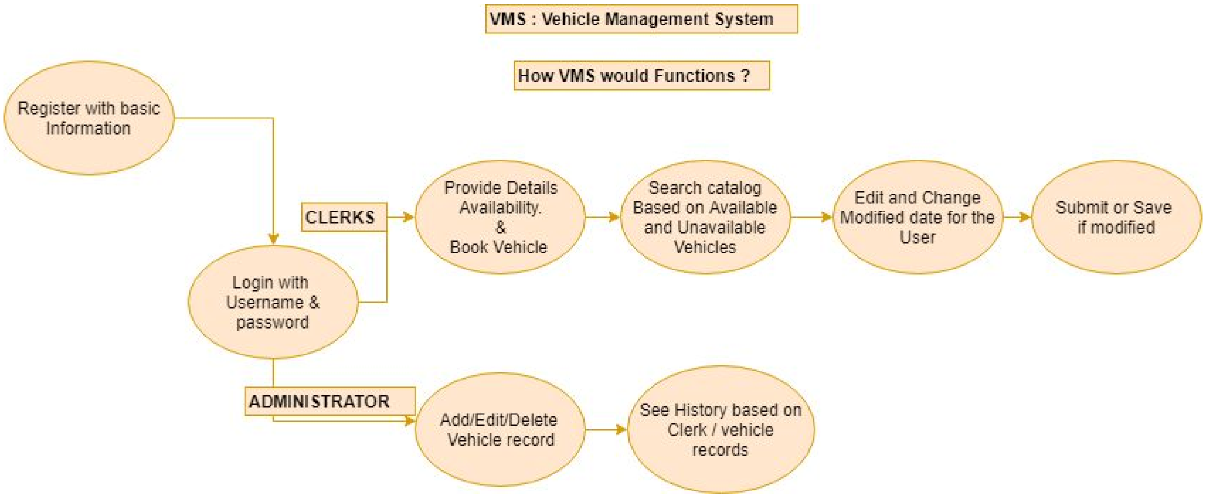

VMS : Vehicle Management System
How VMS would Functions ?
Register with basic Information
CLERKS
Provide Details Availability. & Book Vehicle
Search catalog Based on Available and Unavailable Vehicles
Edit and Change Modified date for the User
Submit or Save if modified
Login with Username & password
ADMINISTRATOR
Add/Edit/Delete Vehicle record
See History based on Clerk / vehicle records

***Figure :2**.2*

User characteristics

The table listed below shows the level of expectancy required to operate VMS.

User/Actors	Expected Education level	Experience	technical Expertise ***(if any)***	Basic Description***(if any)***
Clerk	Educated person required or High School passed.	High School passed.	No. Basic Training is required to operate VMS.	None.
Administrator	Educated person required or High School passed.	High School passed.	No. Basic Training is required to operate VMS.	None.

Constraints

Safety Regulation : Stakeholder or Administrator can View or monitor the request submitted by the clerk.

Hardware : Since the application will run on browser if hosted on Server than connection with internet is required to open the VMS site.or if planned to host on server to work on intranet also then TCP/IP protocol suite is required.

Limitation : VMS will work on HTTP on browser on Port Number 80.So if you are planning to run something on port number 80 better be careful about the same.

Safety & Security : It can store the username and password in Session variable and cookies.So better delete it if you don't want to share with anyone else.

User Interface : The user can open it in Chrome,IE and Safari and the user interface shall be implemented in java,EJB etc.

Software Interface : It has various screens such as FrmRegister,FrmLogin,FrmChoose,FrmSearch,FrmClient.

Communication Interface : The VMS should use HTTP protocol for communication on internet if hosted.

Licensing requirement: Not Applicable.

Applicable Standards : It should be as per industry standard.

Assumptions and dependencies

1. Our Assumption would be that this product should only be used on browser since its a web application.If may work on mobile but since the design is according to PC browser height width width.
2. Other assumption which need to be cleared since it is a vehicle management website it doesn't use any GPS.
3. Make sure java plugin is installed in your browser. many a times if user may not allow to run java in back end the application seems to be not working properly.
4. It should work on various OS like Windows 10,Windows 7 and Mac.

3. Specific requirements

1. Functional Requirements

Identification	REQ-001
Version	1.0
Priority	High
Risk	Medium
Description	● There should be an interface to take all the information of the clerk: First Name, Last Name, Username, Password, Email and an option to register as admin if needed. ● There should be a register button which validates user input fields and creates new clerk record.
Type	Functional

Identification	REQ-002
Version	1.0

Priority	High
Risk	High
Description	● There should be an interface to take Username and Password from the user. ● There should be a login button which validates the credentials with existing records and allows only valid users to access the system.
Type	Functional

Identification	REQ-003
Version	1.0
Priority	High
Risk	High
Description	● There should be a vehicle catalog page to display vehicles, it should have the following information: Make, Model, Year and Type ● There should be an option to view the details of the vehicle. This option should only be visible for the clerk and not the admin.
Type	Functional

Identification	REQ-004
Version	1.0
Priority	High
Risk	Medium
Description	● There should be an option to filter the vehicles tables according to Make, Model, Year and Type.

Type	Functional

Identification	REQ-005
Version	1.0
Priority	High
Risk	Low
Description	● There should be an option to sort the vehicles tables according to Make, Model, Year and Type.
Type	Functional

Identification	REQ-006
Version	1.0
Priority	High
Risk	High
Description	● There should be a detailed view page for the vehicle which provides its Make, Model, Year, Type, Color, Licence Plate Number and availability details. ● There should be an option to book the vehicle. ● Detailed View page should be only accessible to the clerk.
Type	Functional

Identification	REQ-007

Version	1.0
Priority	High
Risk	High
Description	● There should be an option to view next vehicle record from vehicle details option page. ● There should be an option to go back to the vehicle catalog page from vehicle details page.
Type	Functional

Identification	REQ-008
Version	1.0
Priority	High
Risk	High
Description	● There should be a client manager page, which gives the details of all clients that have made a reservation. Details include: client ID, First Name, Last Name, Licence Number, Licence Validity, Phone Number, Start Date , Due Date, Booking Id, Booken On and Status. ● CLient Manager should only be accessible to the clerk.
Type	Functional

Identification	REQ-009
Version	1.0
Priority	High
Risk	High
Description	● The clerk should be able to update return of vehicle in client manager page.

Type	Functional

Identification	REQ-010
Version	1.0
Priority	High
Risk	High
Description	● The clerk should be able to update cancellation of vehicle in client manager page.
Type	Functional

Identification	REQ-011
Version	1.0
Priority	High
Risk	High
Description	● The clerk should be able to modify client details in client manager page.
Type	Functional

Identification	REQ-012
Version	1.0
Priority	High
Risk	High
Description	● The booking page should accept client details:First Name, Last Name, Licence Number, Licence Validity, Booking Start and End Date. ● The booking page should be only accessible to

	the clerk.
Type	Functional

Identification	REQ-013
Version	1.0
Priority	High
Risk	High
Description	● The booking page should have book now button that validates all client information and confirms booking if only all the client data is valid.
Type	Functional

Identification	REQ-014
Version	1.0
Priority	High
Risk	High
Description	● There should be a Vehicle Master data page which shows vehicle details and should have option to modify, add or delete vehicle records. ● The Vehicle Master data page should be only accessible by the admin.
Type	Functional

Identification	REQ-015

Version	1.0
Priority	High
Risk	High
Description	● There should be a Transactions page which gives all the details about transaction including client, vehicle and booking details. ● The Transactions page should be only accessible by the admin.
Type	Functional

Identification	REQ-016
Version	1.0
Priority	High
Risk	Medium
Description	● There should be an option to filter the transaction table according to booking status, due/return dates, client name or vehicle name..
Type	Functional

2. Non Functional Requirements

Identification	REQ-017

Version	1.0
Priority	High
Risk	High
Description	● The response for any request needs to be processed fast , maximum latency for concurrent operation must be less than 5 seconds.
Type	Non-Functional

Identification	REQ-018
Version	1.0
Priority	High
Risk	Medium
Description	● Storage utilization must be optimal, it should not include redundant data.
Type	Non-Functional

Identification	REQ-019
Version	1.0
Priority	High
Risk	High
Description	● The application should be easily scalable and must support at least 1000 records.
Type	Non-Functional

Identification	REQ-020

Version	1.0
Priority	High
Risk	Medium
Description	● The application should provide an easy to use user interface with proper error messages and error handling.
Type	Non-Functional

Identification	REQ-021
Version	1.0
Priority	High
Risk	Medium
Description	● The application should be modular and easy to maintain and modify.
Type	Non-Functional

Identification	REQ-022
Version	1.0
Priority	High
Risk	Medium
Description	● The application should be secure to use, all passwords must be encrypted and stored. ● Username and Password length validations should be present. ● Admin must be limited to only one per system.
Type	Non-Functional

Identification	REQ-023
Version	1.0
Priority	High
Risk	High
Description	● Create MySQL database table to hold client information with following fields: ○ id ○ firstName ○ lastName ○ licenceNumber ○ licenceValidity ○ phone ○ clerkID ○ bookingId ○ vehicleId ○ version
Type	Non-Functional

Identification	REQ-024
Version	1.0
Priority	High
Risk	High
Description	● There should be an Object Relational Mapper that maps clients model to the client database table. ● Identity pattern should be followed that uses unique id , and auto increments when new records are added.
Type	Non-Functional

Identification	REQ-025
Version	1.0
Priority	High
Risk	High
Description	● Create MySQL database table to hold booking information with following fields: ○ id ○ bookingTS ○ returnDate ○ startDate ○ dueDate ○ cancelDate ○ clientID ○ vehicleId ○ version
Type	Non-Functional

Identification	REQ-026
Version	1.0
Priority	High
Risk	High
Description	● There should be an Object Relational Mapper that maps bookings model to the booking database table. ● Identity pattern should be followed that uses unique id , and auto increments when new records are added.
Type	Non-Functional

Identification	REQ-027
Version	1.0
Priority	High
Risk	High
Description	● Create MySQL database table to hold vehicle information with following fields: ○ id ○ make ○ model ○ iYear ○ Type ○ LicPlate ○ Color ○ bookingId ○ version
Type	Non-Functional

Identification	REQ-028
Version	1.0
Priority	High
Risk	High
Description	● There should be an Object Relational Mapper that maps vehicles model to the vehicles database table. ● Identity pattern should be followed that uses unique id , and auto increments when new records are added.
Type	Non-Functional

Identification	REQ-029
Version	1.0
Priority	High
Risk	High
Description	● Create MySQL database table to hold clerk/admin information with following fields: ○ id ○ firstName ○ lastName ○ username ○ password ○ email ○ usergroup
Type	Non-Functional

Identification	REQ-030
Version	1.0
Priority	High
Risk	High
Description	● There should be an Object Relational Mapper that maps clerks model to the clerks database table. ● Identity pattern should be followed that uses unique id , and auto increments when new records are added.
Type	Non-Functional

Identification	REQ-031
Version	1.0
Priority	High
Risk	High
Description	● The system should be able to handle concurrent requests. ● Optimistic locking pattern using version of record entry has to be implemented to handle updation of critical resources.
Type	Non-Functional

Performance efficiency

The performance efficiently for VMS focuses on following things which are as follows :

1. The system should at least support 1,000 vehicles..
2. In standard CPU usage shall be less than 50%.
3. Going from one page to another may contain data can take upt 1-2 sec.
4. The system should handle concurrent user requests.

Compatibility

Few compatibility issues that VMS can faced which are as follows :

Design compatibility : Since we are working on HTML5 with bootstrap design which can hamper the design when browser can get changed or won't work when opened in old browser of IE 4.0 or earlier or Chrome earlier version which don't not support some functionality.

Java Earlier Version or Plug allowed on web browser : user should have the latest version of Java installed or plugin active in browser which he need to allow it.Since our application is in java. user should allow the java in popup in browser.

It may not work if java is not enabled or clicked on denied.

Usability

The ability in VMS of using existing components which can be reused in VMS application which can be as follows :

1. Code design and percentage of reused requirements
2. Coupling of components.
3. Degree and use of framework.
4. Functions

Reliability

The reliability constraints for VMS consist of things which are as follows :

1. Resistance to be failure free and fault tolerant.
2. Ability to perform certain tasks under specific amount of time.For e.g Asker for reservation and return date of vehicle based on the cretia given by user to client and if any failure occurs may result to pop up saying error message occurred or switching to login page in case of session value turn out to be null since it can save the value for only 10 minutes.
3. It should work based on the conditions which is specified in the system like sorting and searching and booking without errors.

Security

Security measures for VMS are taken as follows :

1. The ability to resist entering unwanted url will result into login page.
2. Success rate at time of authentication.
3. Lifespan of password for session.
4. Passwords are encrypted and stored to add security and user privacy.

Maintainability

Maintainability are measures which has been considered while making VMS such as :

1. Extension of new functionality if given by stakeholders.
2. Adapting and working in new OS such as MAC and windows various versions.
3. Quality/ quantity managing the requirements which should be documented well for understanding what stakeholders is trying to say and developer and designed are trying to design.
4. Code analysis tool can also be used to see whether proper procedure has been followed or not while designing,developing and documenting.

5. Ensure up-to-date documentation which enhances readability, understandability and maintainability.

Portability

Probability measures are those measured in which VMS can run under different environment which are as follows :

1. Target platform(hardware,OS)
2. Some platforms work specific to components with some functionality should be kept in mind while making VMS.

Design constraints

Following are Design constraints must be kept in mind while making VMS which are as follows :

Languages : Java ,HTML5, CSS 3.0, C#,Vb.net

Framework : Spring/ spring boot or .Net

Database : mysql workbench or SQL Server

Tools : Eclipse IDE or IntelliJ

Server : Tomcat 7.0,8.0,9.0 Versions can be used

Architectural design : MVC pattern with java spring framework.

Revision version control : Github or Gerrit

Software Project management tool : Maven in Eclipse for spring dependencies or Gradle or dynamic web project.

Libraries Used : Maven dependencies such as core,context,WebMVC and JSTL expression language or any language in which developer can write expressions easily.

(On-line) user documentation and help

Since we have been designing and developing this document a help document and user manual document will be created at least once the version/iteration 2.0,3.0,4.0 are up and running and complete as per stakeholder.

Purchased components

The plans for VMS if get successful than it will be called as CRMS software which can be purchased in future.

Licensing requirements

A GPL will be carried out for VMS for the user and can be modified according to stakeholder need.

Legal, copyright and other notices

All rights will be reserved by Team which is named as of (Team of 6) lead by team leader and guided by our professor Dr. Constantinos Constantinides.

3.1 External interfaces

These are expected result in psd file which has been given by stakeholder and a glimpse of how an application should look.Following are the .psd file according to which design should me made and get started by development. [later can changed as per design]

Listed all the User Interface which should be as follows:

Registration page

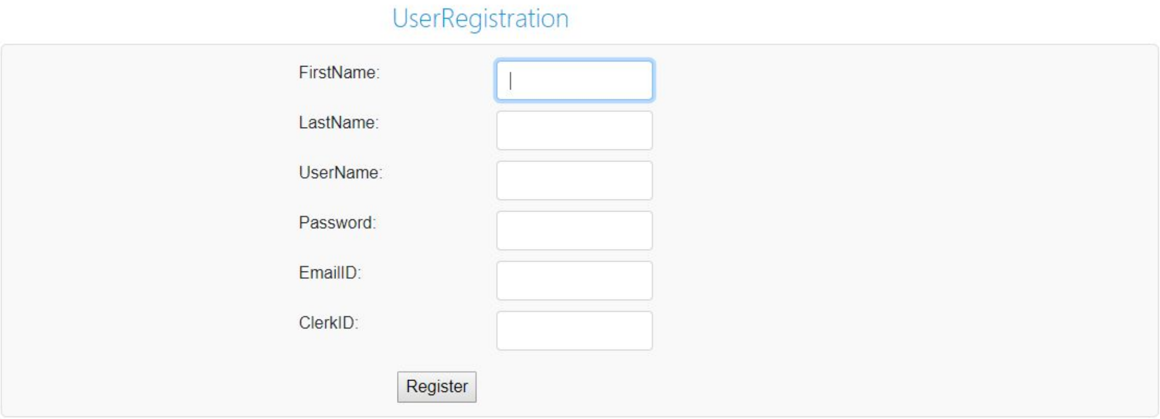


Figure:3.1

Login Page

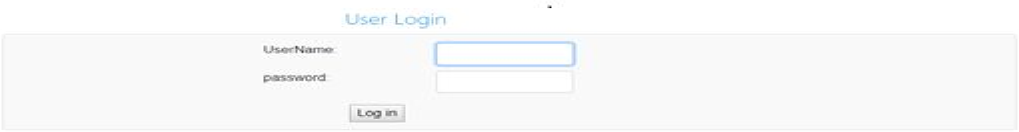
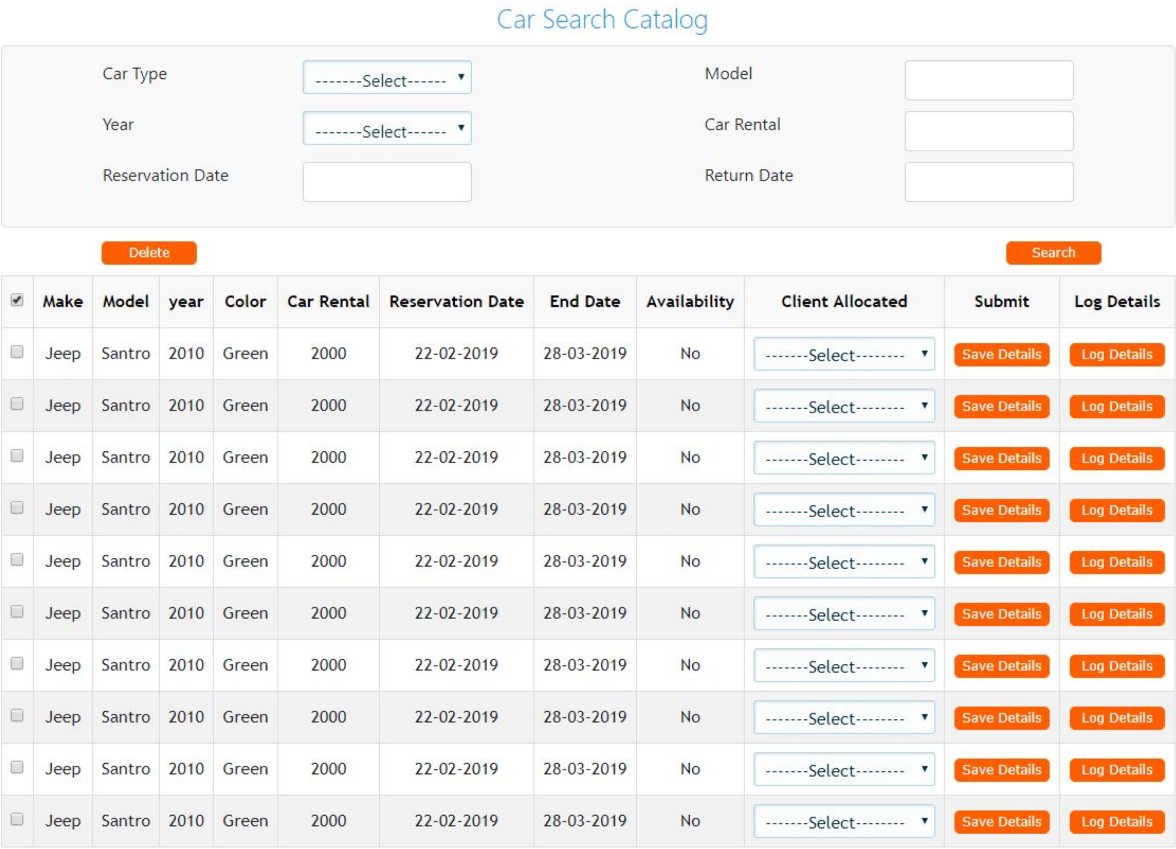


Figure:3.2

Vehicle Information page/ clerk with sorting and booking.



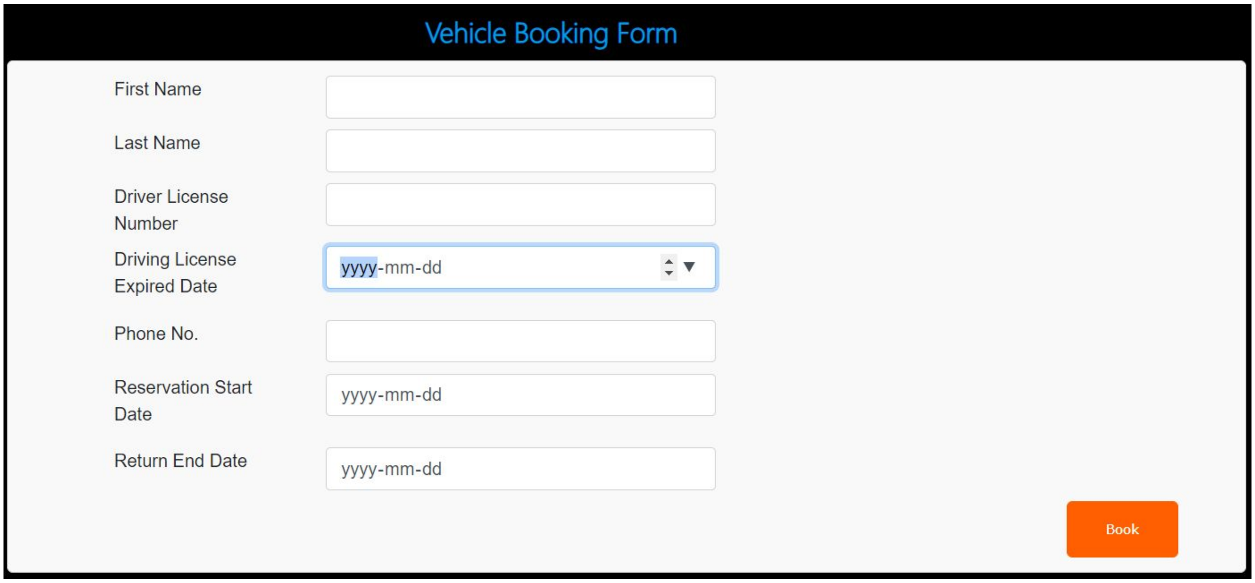
Car Search Catalog

Car Type	-------Select------ ▾	Model	
Year	-------Select------ ▾	Car Rental	
Reservation Date		Return Date	

☑	Make	Model	year	Color	Car Rental	Reservation Date	End Date	Availability	Client Allocated	Submit	Log Details
□	Jeep	Santro	2010	Green	2000	22-02-2019	28-03-2019	No	-------Select-------- ▾	Save Details	Log Details
□	Jeep	Santro	2010	Green	2000	22-02-2019	28-03-2019	No	------Select-------- ▾	Save Details	Log Details
□	Jeep	Santro	2010	Green	2000	22-02-2019	28-03-2019	No	------Select-------- ▾	Save Details	Log Details
□	Jeep	Santro	2010	Green	2000	22-02-2019	28-03-2019	No	------Select-------- ▾	Save Details	Log Details
□	Jeep	Santro	2010	Green	2000	22-02-2019	28-03-2019	No	------Select-------- ▾	Save Details	Log Details
□	Jeep	Santro	2010	Green	2000	22-02-2019	28-03-2019	No	------Select-------- ▾	Save Details	Log Details
□	Jeep	Santro	2010	Green	2000	22-02-2019	28-03-2019	No	------Select-------- ▾	Save Details	Log Details
□	Jeep	Santro	2010	Green	2000	22-02-2019	28-03-2019	No	------Select-------- ▾	Save Details	Log Details
□	Jeep	Santro	2010	Green	2000	22-02-2019	28-03-2019	No	------Select-------- ▾	Save Details	Log Details
□	Jeep	Santro	2010	Green	2000	22-02-2019	28-03-2019	No	------Select-------- ▾	Save Details	Log Details

Figure:3.3

Booking Page



Vehicle Booking Form

First Name

Last Name

Driver License Number

Driving License Expired Date: yyyy-mm-dd

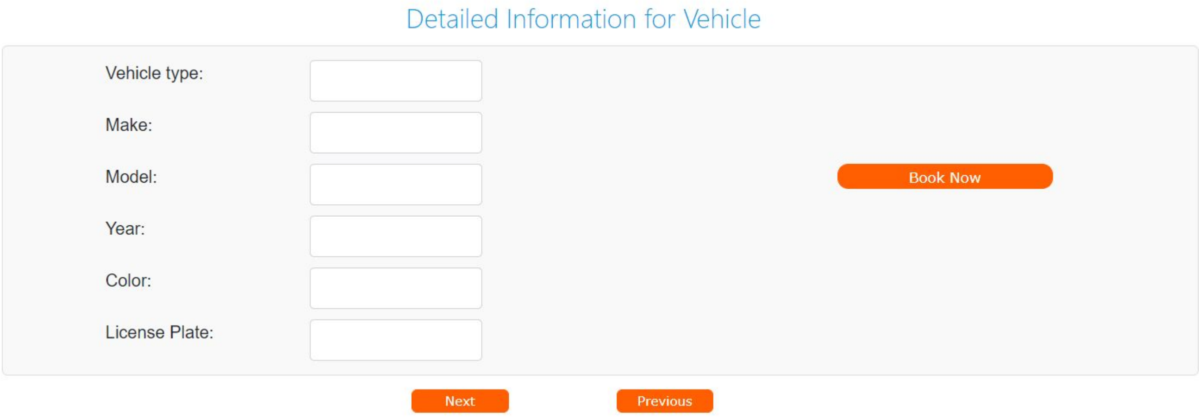
Phone No.

Reservation Start Date: yyyy-mm-dd

Return End Date: yyyy-mm-dd

Figure:3.4

Detailed View page



Detailed Information for Vehicle

Vehicle type:

Make:

Model:

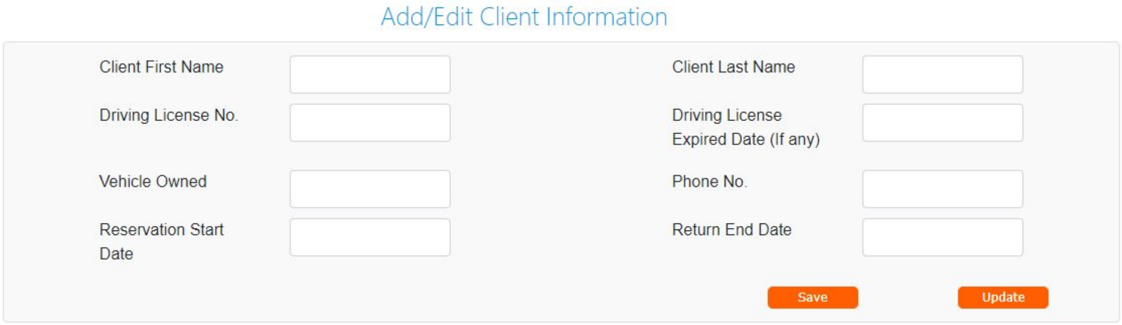
Year:

Color:

License Plate:

Figure:3.5

User Information update page

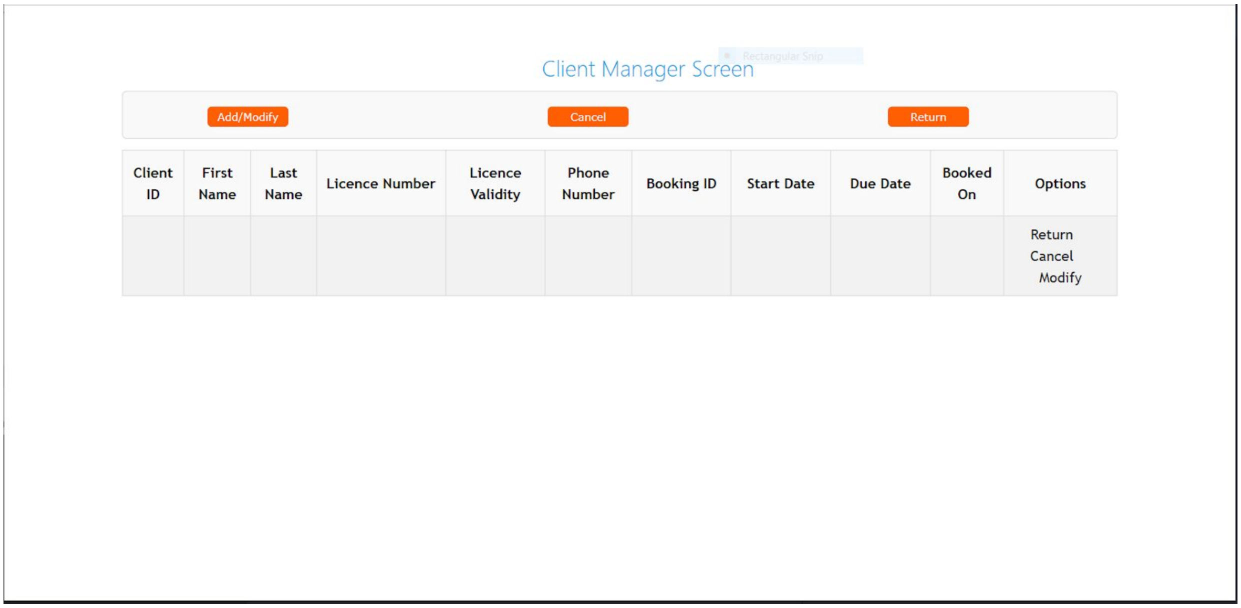


Add/Edit Client Information

Client First Name		Client Last Name	
Driving License No.		Driving License Expired Date (If any)	
Vehicle Owned		Phone No.	
Reservation Start Date		Return End Date	

Figure:3.6

Client Manager page

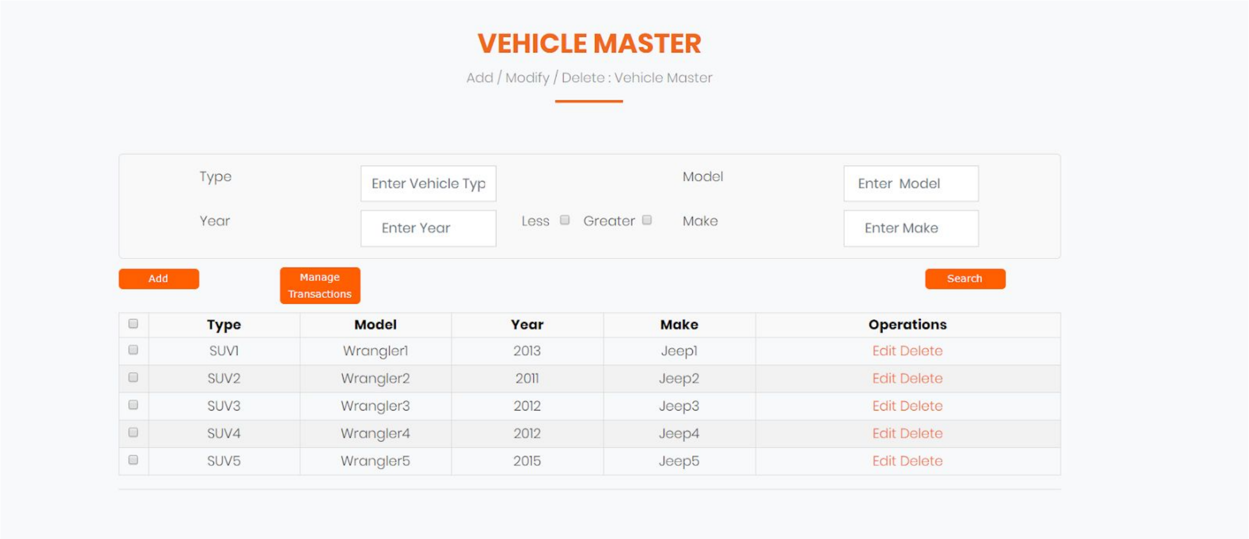


Client Manager Screen

Client ID	First Name	Last Name	Licence Number	Licence Validity	Phone Number	Booking ID	Start Date	Due Date	Booked On	Options
										Return Cancel Modify

Figure:3.7

Vehicle Master -with Filtering criteria



VEHICLE MASTER

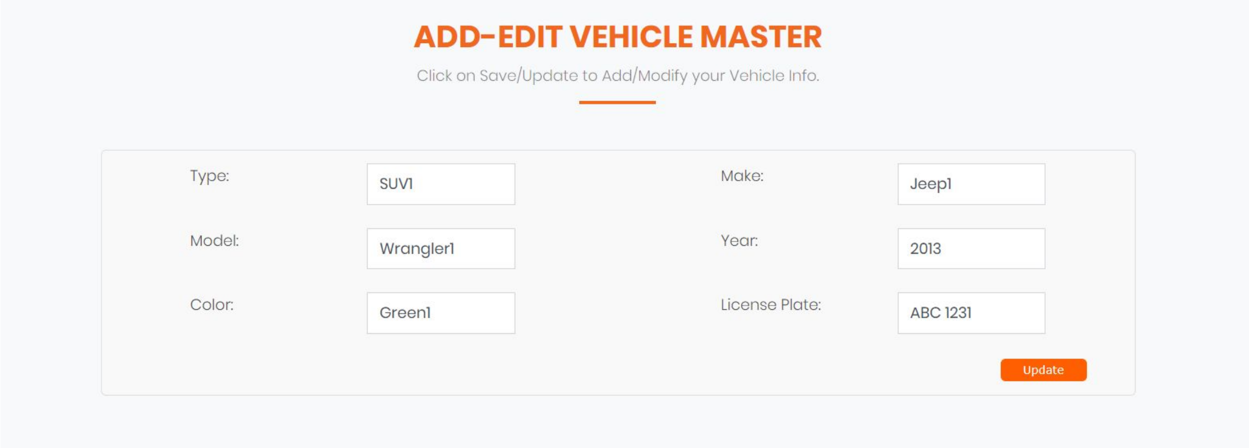
Add / Modify / Delete : Vehicle Master

Type	Enter Vehicle Typ	Model	Enter Model
Year	Enter Year	Less □ Greater □ Make	Enter Make

□	Type	Model	Year	Make	Operations
□	SUV1	Wrangler1	2013	Jeep1	Edit Delete
□	SUV2	Wrangler2	2011	Jeep2	Edit Delete
□	SUV3	Wrangler3	2012	Jeep3	Edit Delete
□	SUV4	Wrangler4	2012	Jeep4	Edit Delete
□	SUV5	Wrangler5	2015	Jeep5	Edit Delete

Figure:3.8

Vehicle Master -with Add/Modify/Delete Screen

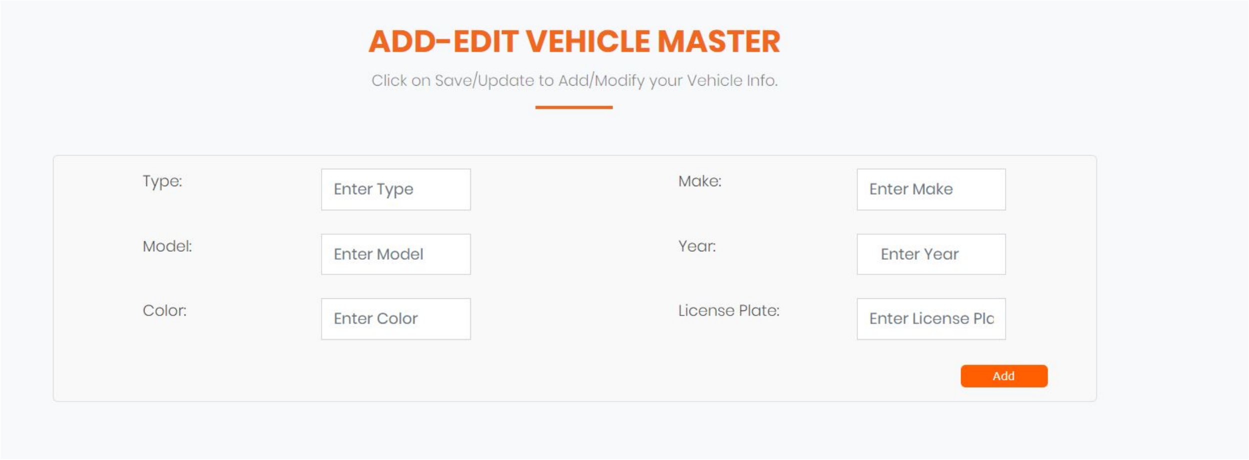


ADD-EDIT VEHICLE MASTER

Click on Save/Update to Add/Modify your Vehicle Info.

Type:	SUV1	Make:	Jeep1
Model:	Wrangler1	Year:	2013
Color:	Green1	License Plate:	ABC 1231

Figure:3.9



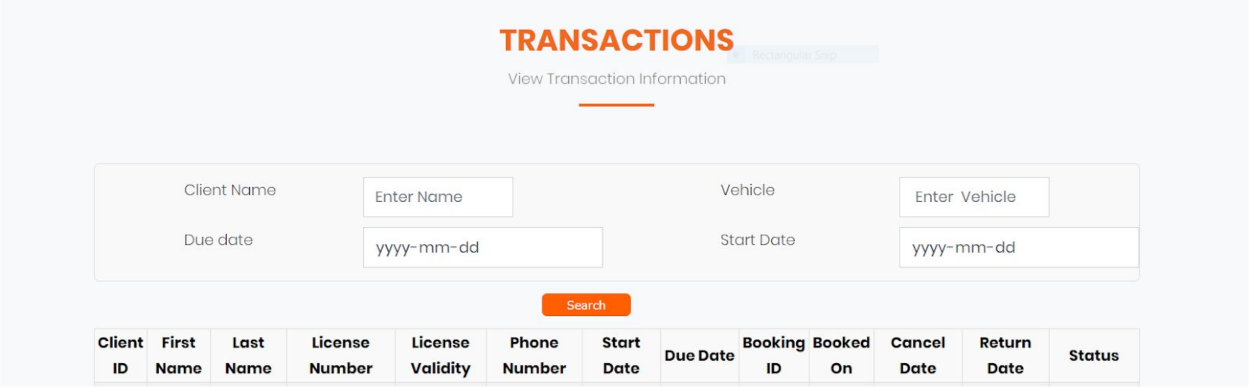
ADD-EDIT VEHICLE MASTER

Click on Save/Update to Add/Modify your Vehicle Info.

Type:	Enter Type	Make:	Enter Make
Model:	Enter Model	Year:	Enter Year
Color:	Enter Color	License Plate:	Enter License Plc

Figure:4.0

Transaction Screen with filtering Screen



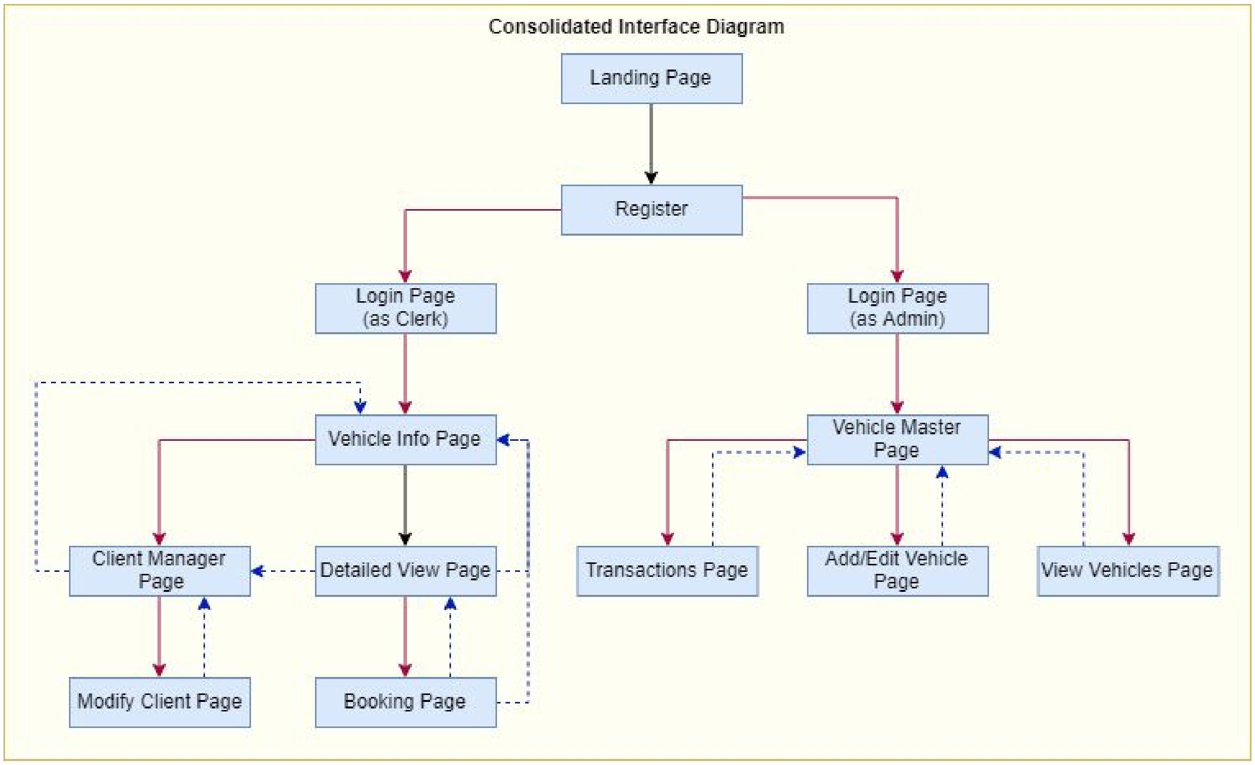
TRANSACTIONS

View Transaction Information

Client Name	Enter Name	Vehicle	Enter Vehicle
Due date	yyyy-mm-dd	Start Date	yyyy-mm-dd

Client ID	First Name	Last Name	License Number	License Validity	Phone Number	Start Date	Due Date	Booking ID	Booked On	Cancel Date	Return Date	Status

Figure:4.1



Functional requirements capture the intended behaviour of the system. This section contains the *Actor Goal List* and the *Use Case view*.

Actor goal list

Actor	Goal
Clerk	The goal of clerk is following which are as follows : ● Register user based on their personal data such as first name , last name, Driving licence number , username , password and expired license date. ● Based on user's Reservation and Return date of vehicle and their availability is checked. ● Clerk can check the list of available and unavailable car. ● Clerk can update the user info which he entered at the time of registration and can modify(update) the user details. ● clerk can check the log details of a particular client so that tracking of Reserve date,return date of vehicle can be stored with time stamp.

	● Clerk can book the vehicle for client based on rental or reservation in detailed view and than go on booking to finalize the booking with client’s personal information.Following are booking which are as follows 1. Vehicle Details(Vehicle Chosen) -->**Rental Book Now**-->Book Now. 2. Vehicle Details(Vehicle Chosen) -->**Reservation Book Now**-->Book Now.
Administrator	● Can load the information of Vehicles which user wanted to rent for specific date he/she wants.Or can see the Updated record/history or enrollment for every user per timestamp.Since money is involved in term of rental everything need to be stored into the history table to view the history of the clerk by administrator.Th update record can be shown to the administrator. ● Administrator can also search and look for the record based on the following criteria which are as follows : ❖ **Client First Name.** ❖ **Client Last Name.** ❖ **Vehicle Model.** ❖ **Vehicle Type.** ❖ **Due Date of vehicle.** ❖ **Start Date of vehicle.**

Use case view

USE CASE MODEL

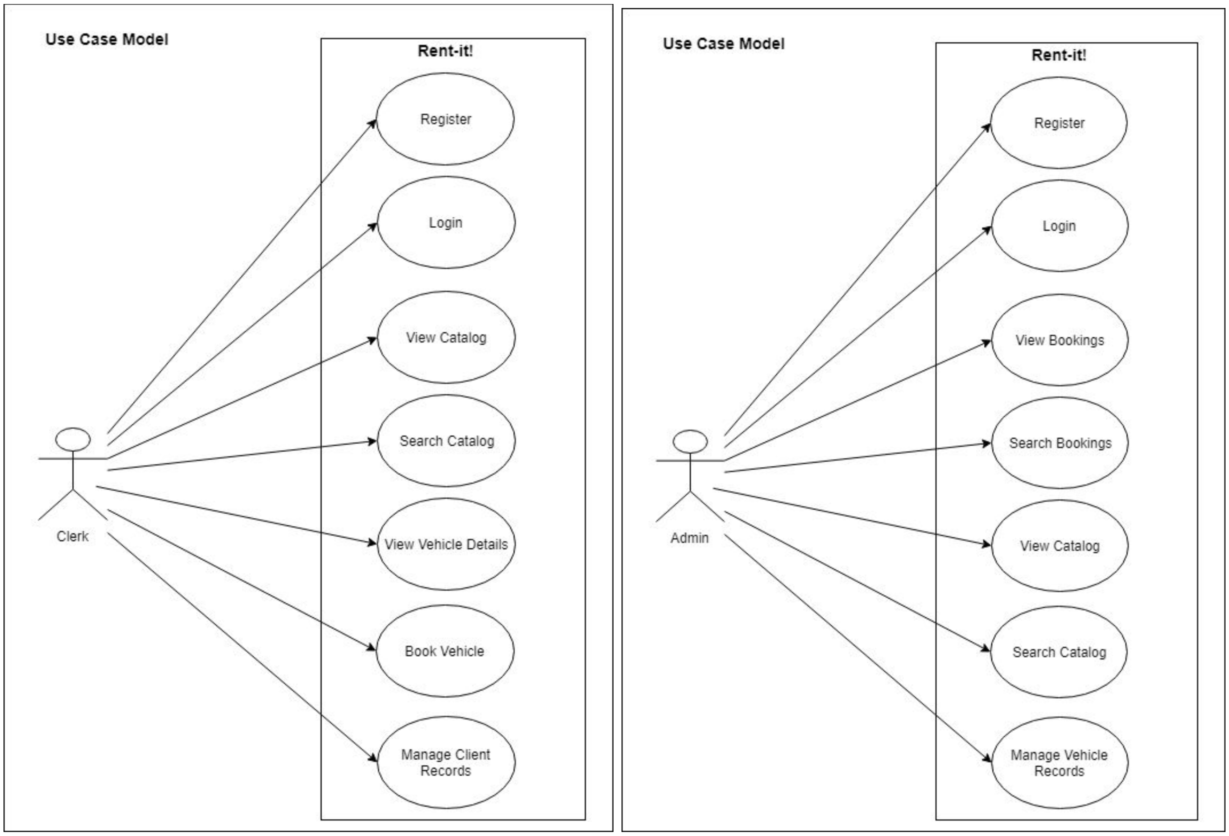


Figure 4.1:UseCase Model

Use Case No.	**Use Case**	**Type**
UC-1	Register	Non-critical
UC-2	Login	Non-critical
UC-3	View Catalog	Non-critical
UC-4	Search Catalog	Critical
UC-5	View Vehicle Details	Non-critical
UC-6	Book Vehicle	Critical
UC-7	Manage Client Records	Critical
UC-8	View Bookings	Non-critical
UC-9	Search Bookings	Critical

UC-10	Manage Vehicle Record	Critical

Use Cases

1. **Register**

   The user enters his details and clicks on the Register button to be registered as a clerk.

2. **Login**

   The user enters his username and password to enter into the system. If invalid credentials are used the system does not grant the user access into the system.

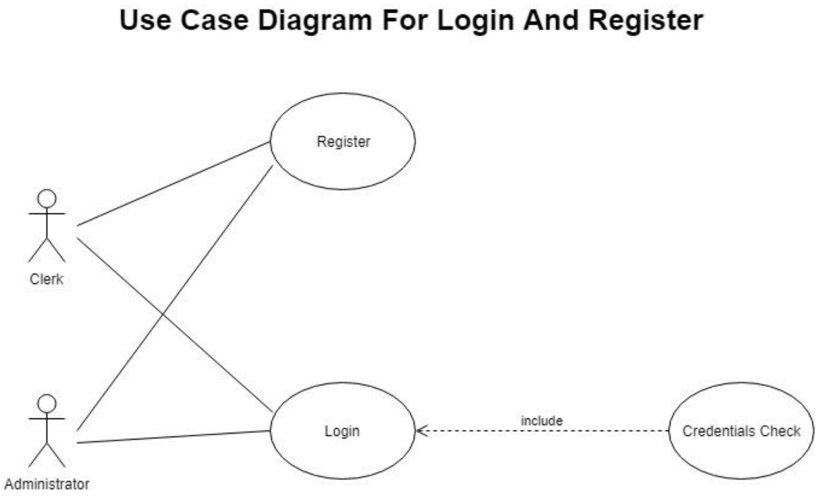


Figure 4.2:UC-1 and UC-2

3. **View Catalog**

   The user can view vehicle table and even sort data.

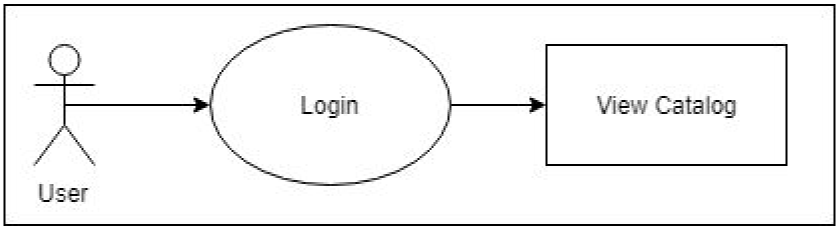


Figure 4.3:UC-3

4. **Search Catalog**

   The user can search vehicle catalog based on different criteria like make, model, type and year.

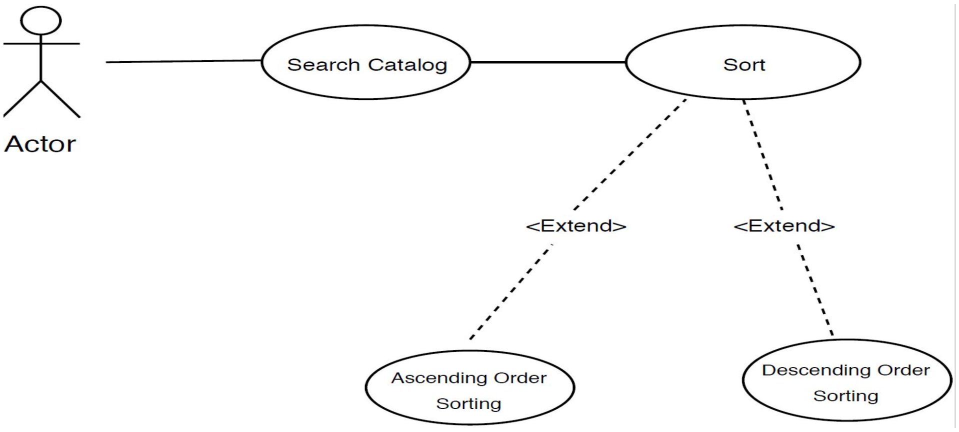


Figure 4.4:UC-4

Description : The clerk can search the vehicle based on filtering criteria which is added by clerk for user at the time of registration. Afterwards clerk can update the personal information of user.

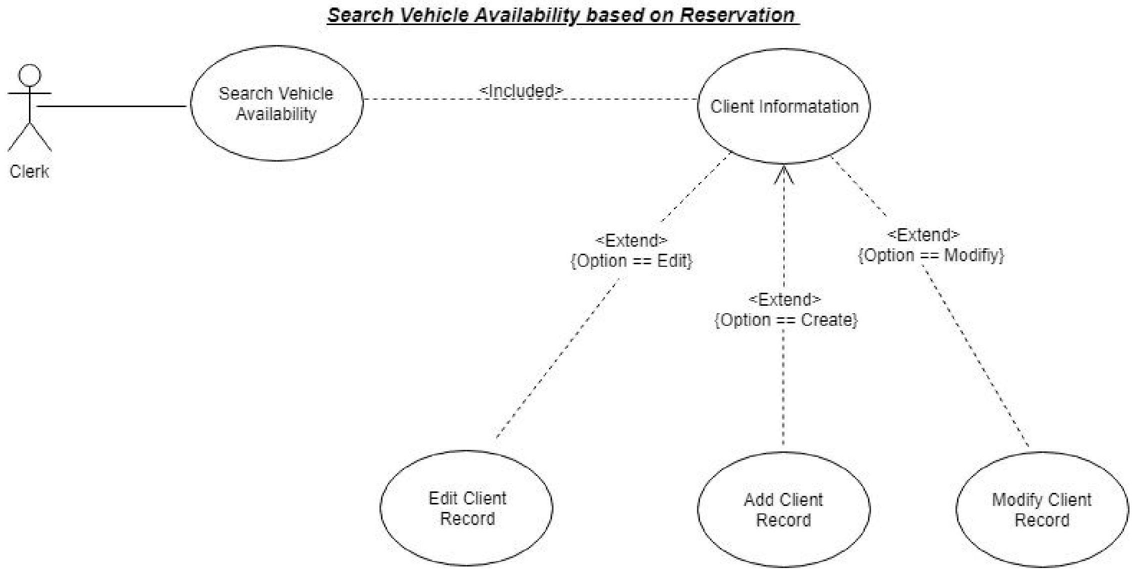
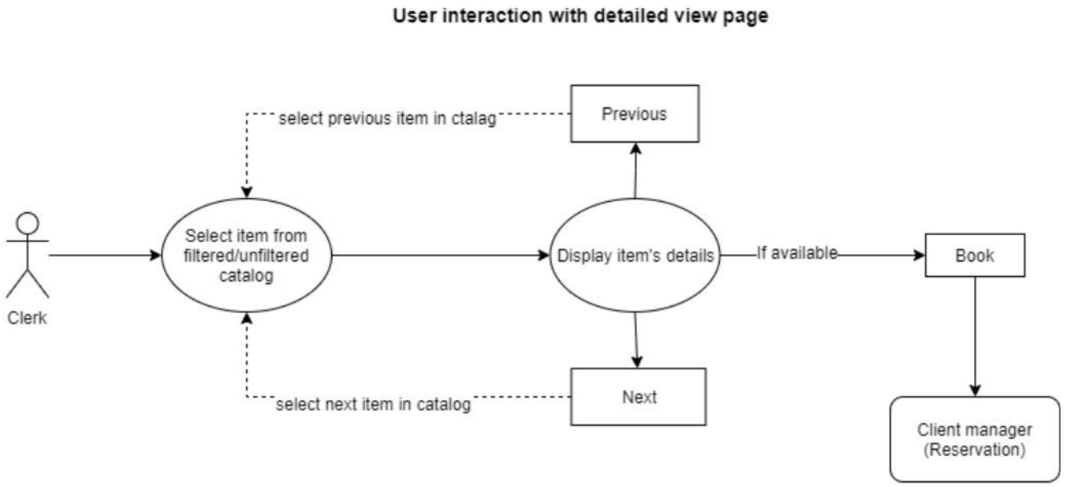


Figure 4.5:UC-4

5. **View Vehicle Details**

The details view gives all the details of the vehicle along with its availability details.



User interaction with detailed view page
select previous item in ctalag
Previous
Clerk
Select item from filtered/unfiltered catalog
Display item's details
If available
Book
select next item in catalog
Next
Client manager (Reservation)

Figure 4.6:UC-5

Details: The clerk can filter the catalog for desired results and view each item in the vehicle catalog in detail. The clerk can then make a booking based on the vehicle's availability. The clerk can also browse through each item of the catalog using the buttons provided in detailed view page.

6. **Book Vehicle**

Provide client details and reserve the vehicle if it is available.

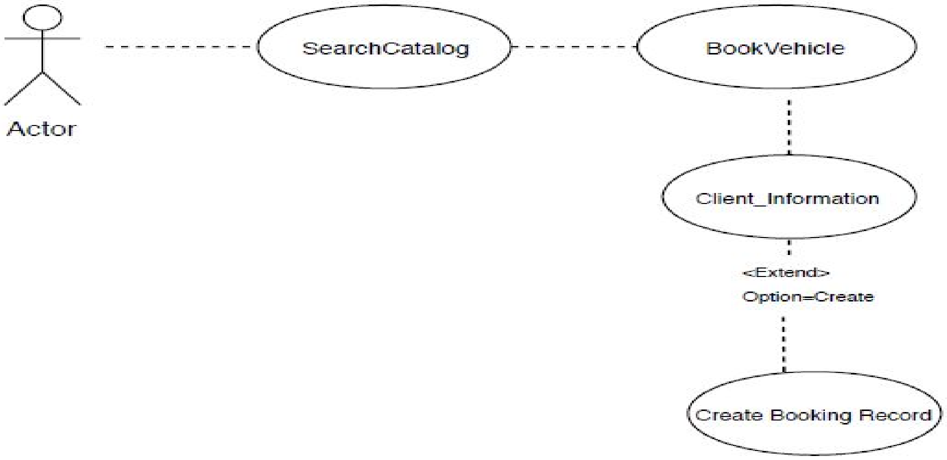
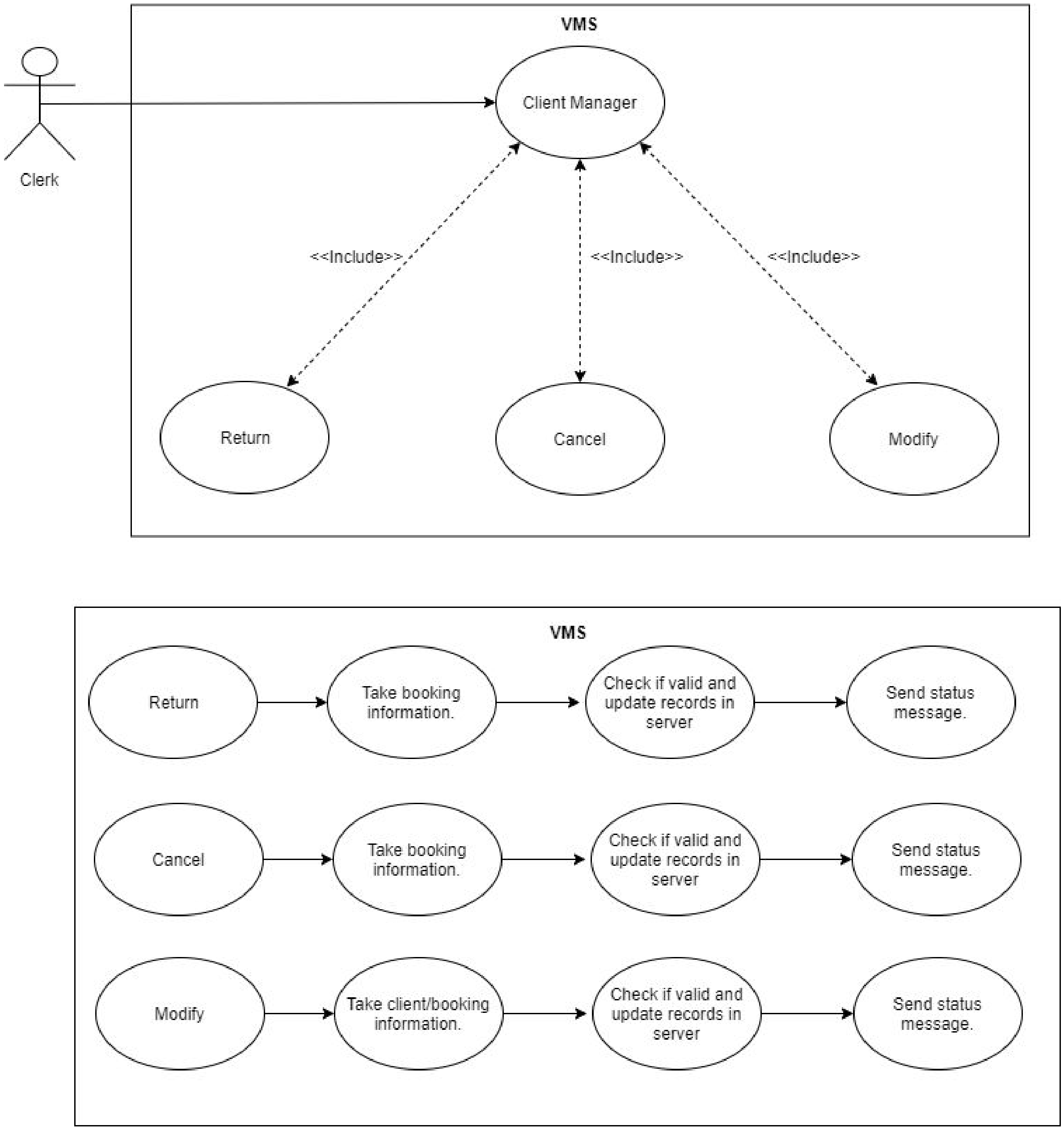


Figure 4.7:UC-6

7. **Manage Client Records**

View client records and also return, cancel a reservation or modify client details.



VMS
Clerk
Client Manager
<<Include>>
<<Include>>
<<Include>>
Return
Cancel
Modify
VMS
Return
Take booking information.
Check if valid and update records in server
Send status message.
Cancel
Take booking information.
Check if valid and update records in server
Send status message.
Modify
Take client/booking information.
Check if valid and update records in server
Send status message.

Figure 4.8:UC-7

8. View Bookings

View Transaction details

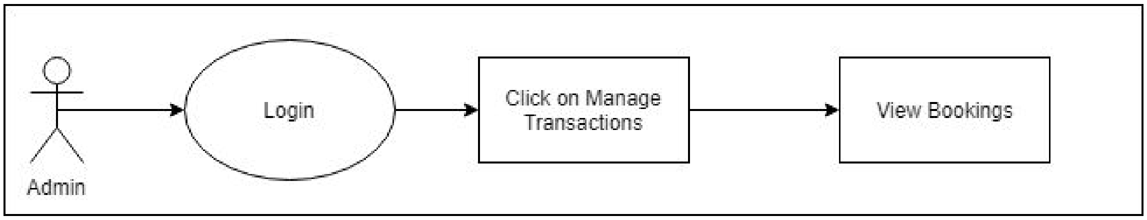


Figure 4.9:UC-8

9. Search Bookings

Filter bookings based on criteria like vehicle, client name and booking details.

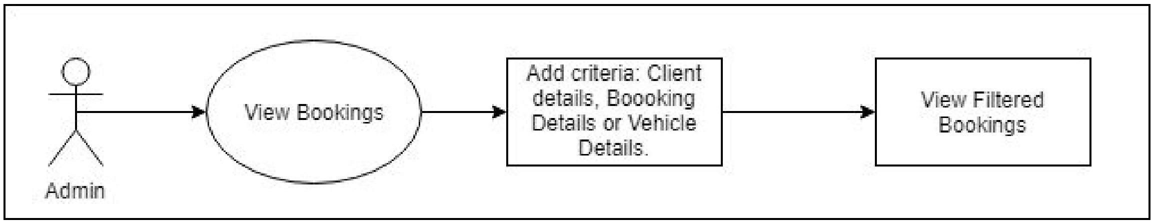

Admin
View Bookings
Add criteria: Client details, Boooking Details or Vehicle Details.
View Filtered Bookings

Figure 5.0:UC-9

10. Manage Vehicle Record

Add/modify or delete vehicle catalog data.

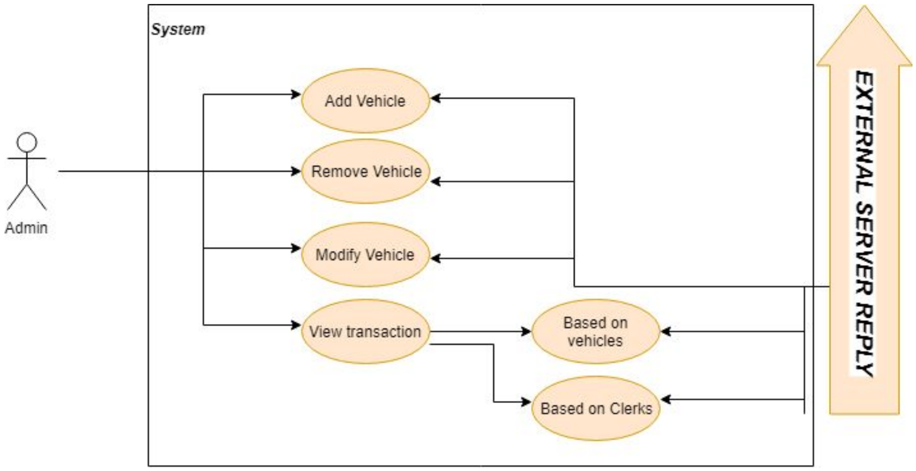


Figure 5.1:UC-10

State Diagrams (Only for Critical use cases):

1. Search Catalog

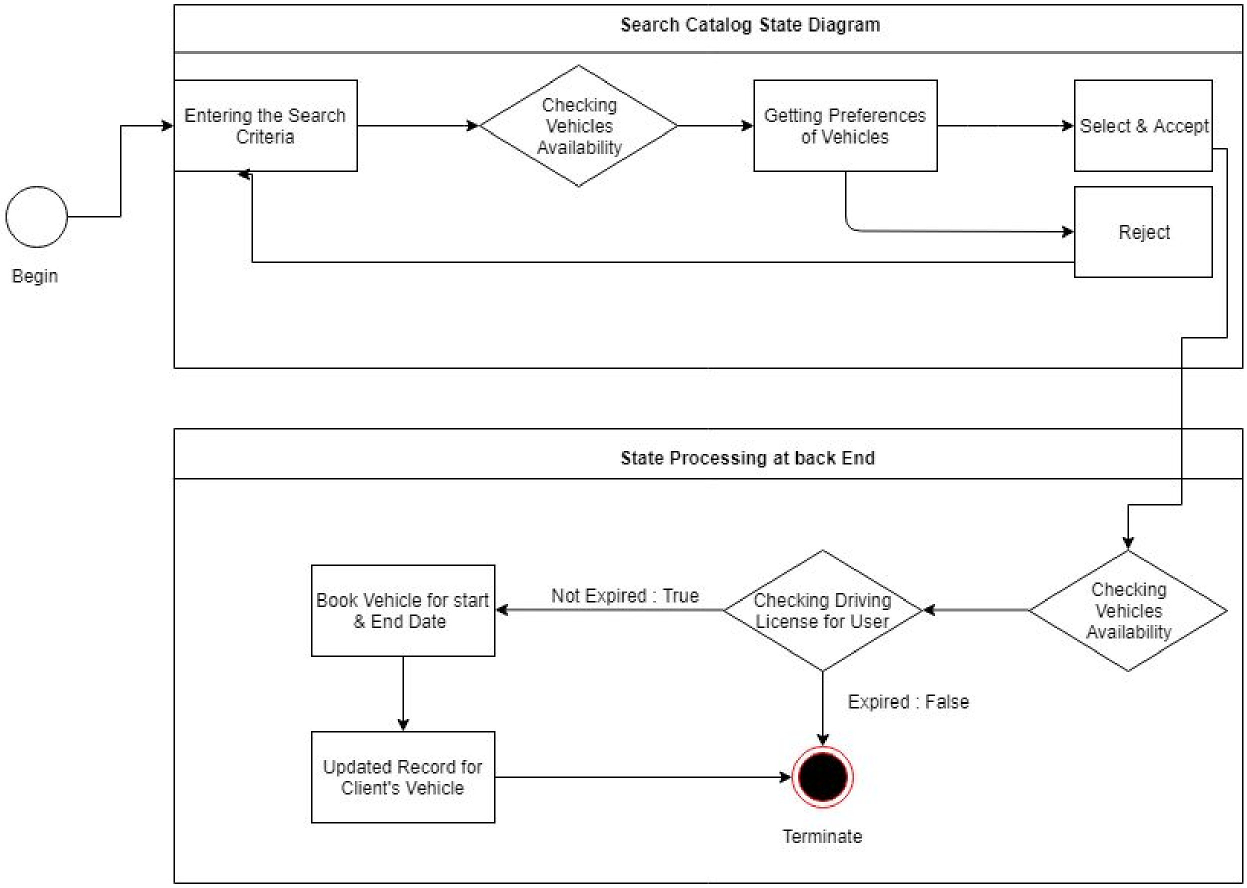

Search Catalog State Diagram
Entering the Search Criteria
Checking Vehicles Availability
Getting Preferences of Vehicles
Select & Accept
Reject
Begin
State Processing at back End
Book Vehicle for start & End Date
Not Expired : True
Checking Driving License for User
Checking Vehicles Availability
Expired : False
Updated Record for Client's Vehicle
Terminate

Figure 5.2:UC-4

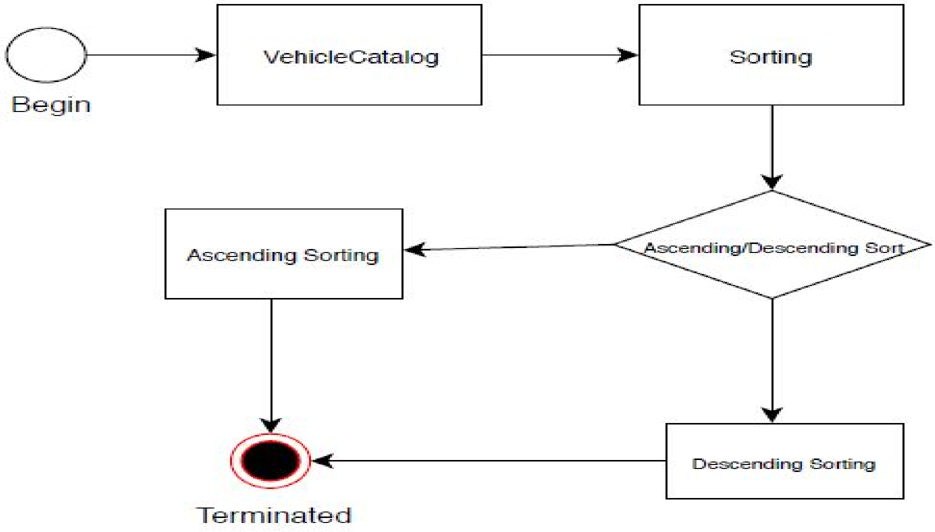


Figure 5.3:UC-4

2. Book Vehicle

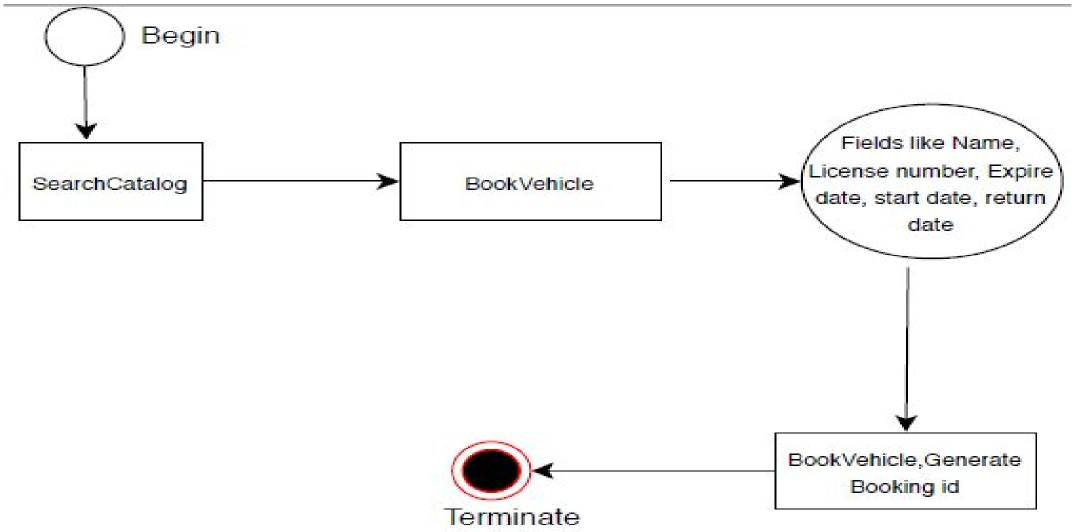
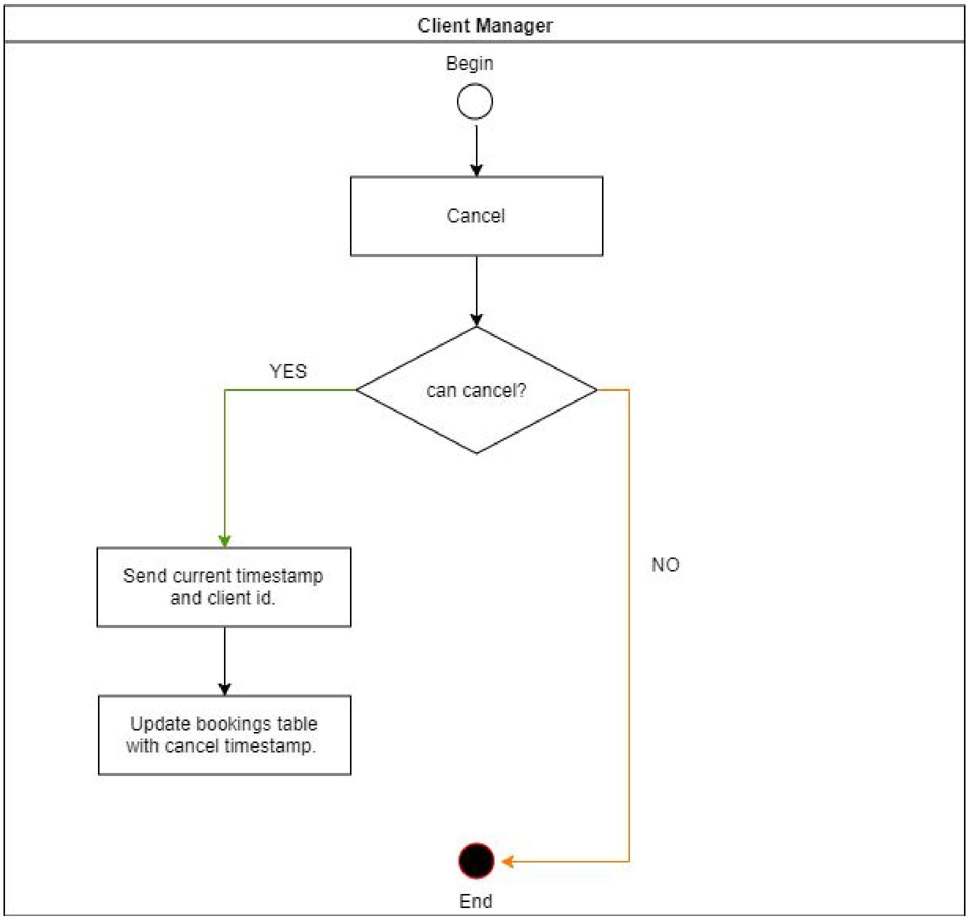

Begin
SearchCatalog
BookVehicle
Fields like Name, License number, Expire date, start date, return date
BookVehicle,Generate Booking id
Terminate

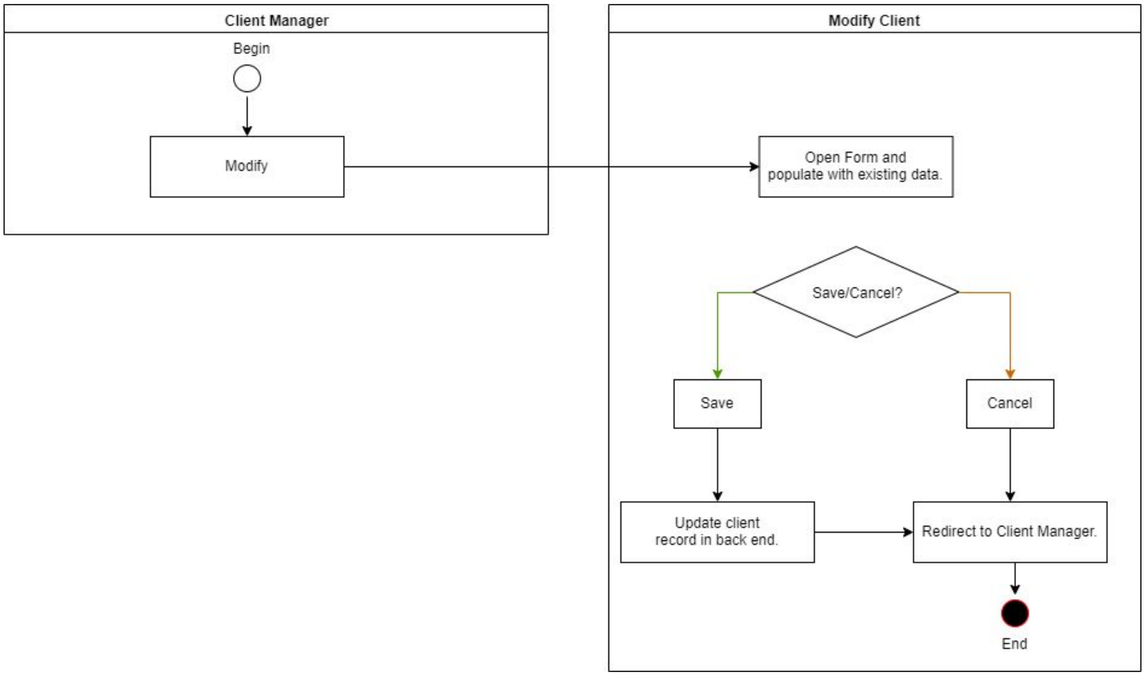
Figure 5.4:UC-6

3. Manage Client Records



Client Manager
Begin
Cancel
can cancel?
YES
NO
Send current timestamp and client id.
Update bookings table with cancel timestamp.
End

Figure 5.5:UC-7



Client Manager
Begin
Modify
Modify Client
Open Form and populate with existing data.
Save/Cancel?
Save
Cancel
Update client record in back end.
Redirect to Client Manager.
End

Figure 5.6:UC-7

4. Search Bookings

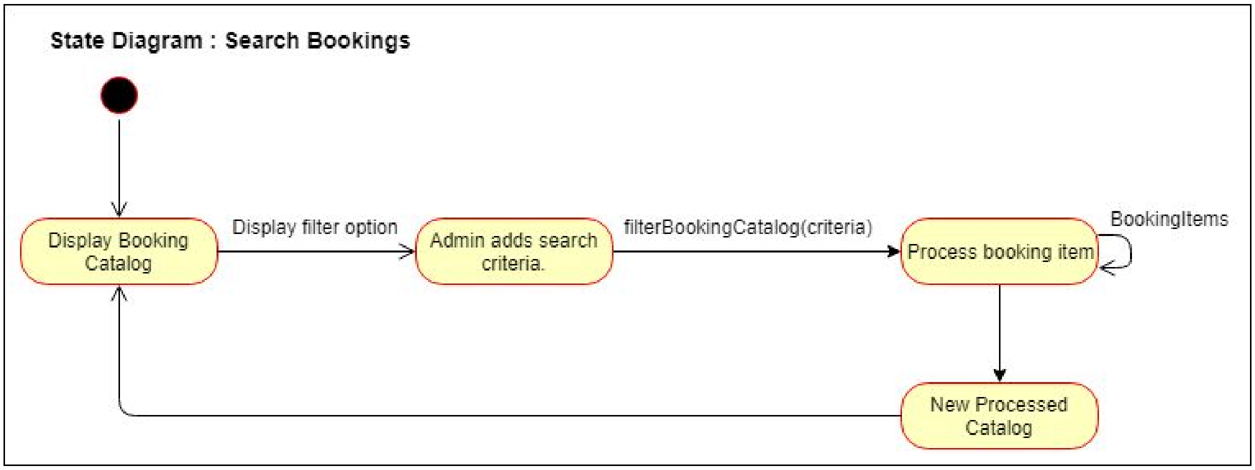
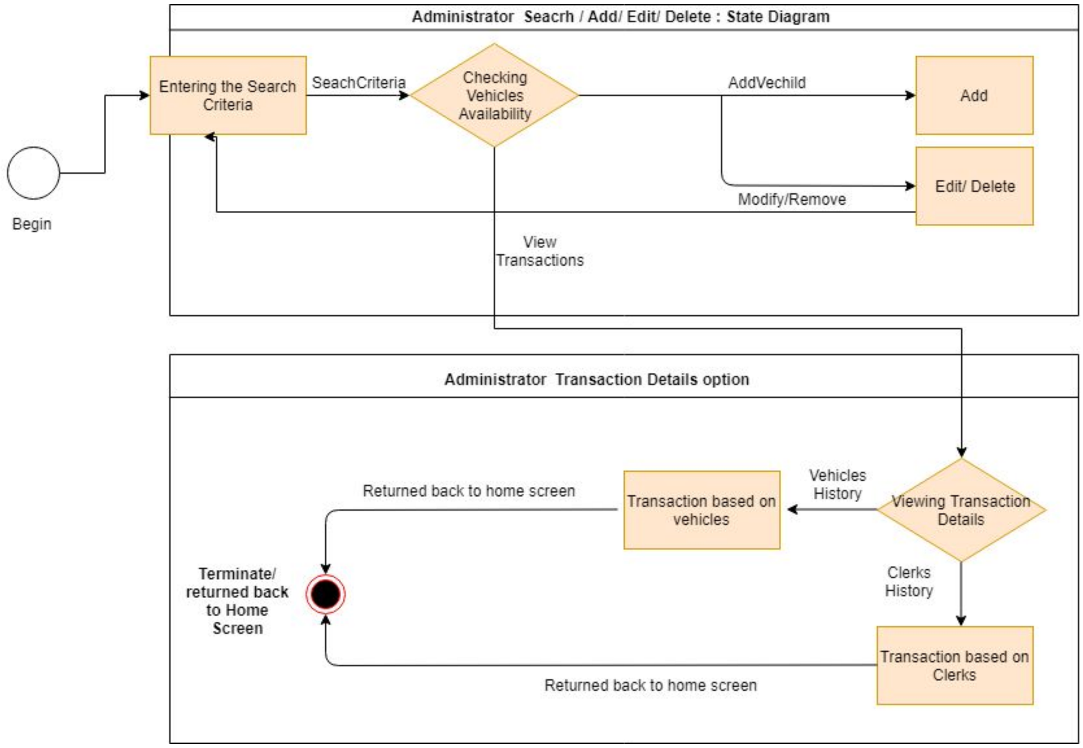


Figure 5.7:UC-9

5. Manage Vehicle Records



Administrator Seacrh / Add/ Edit/ Delete : State Diagram
Begin
Entering the Search Criteria
SeachCriteria
Checking Vehicles Availability
AddVechild
Add
Edit/ Delete
Modify/Remove
View Transactions
Administrator Transaction Details option
Viewing Transaction Details
Vehicles History
Transaction based on vehicles
Returned back to home screen
Clerks History
Transaction based on Clerks
Returned back to home screen
Terminate/ returned back to Home Screen

Figure 5.8:UC-10

System Sequence Diagrams:

1. **UC 1 : User Registration:**

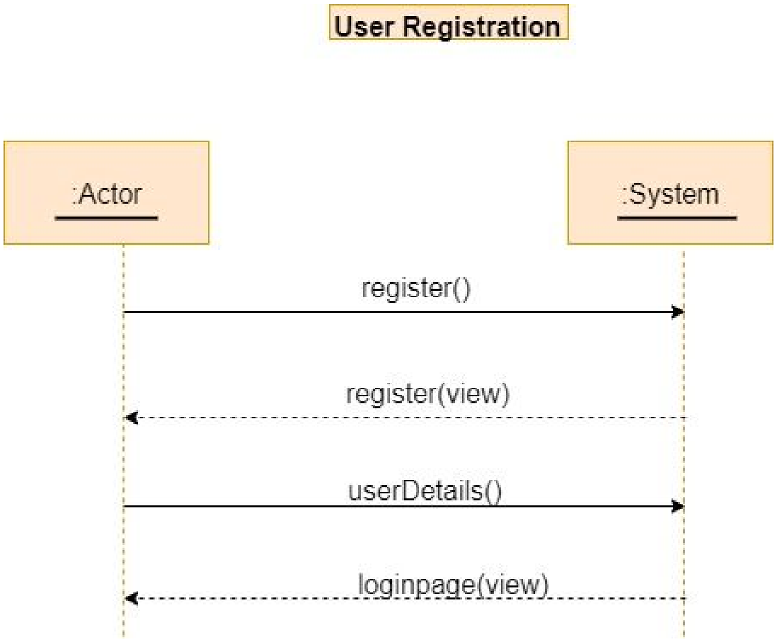


Figure 5.9:UC-1

2. **UC 2 :User Login**

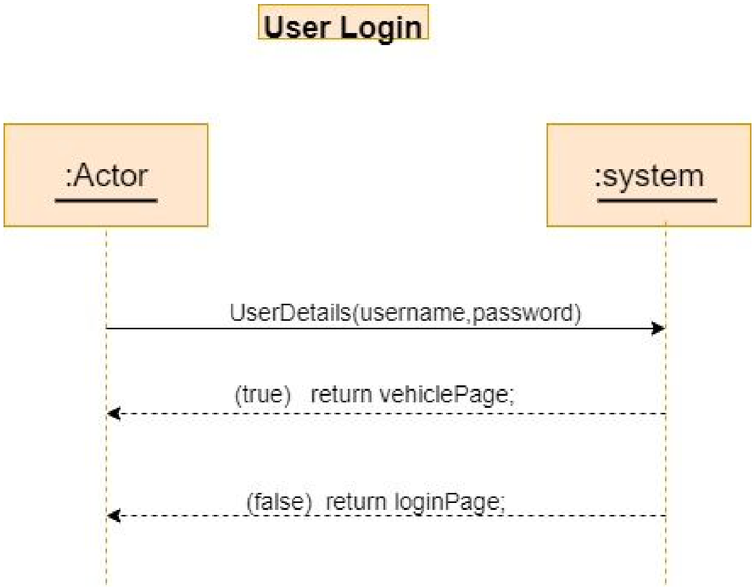

User Login
:Actor
:system
UserDetails(username,password)
(true) return vehiclePage;
(false) return loginPage;

Figure 6.0:UC-2

3. UC 3 : View Catalog

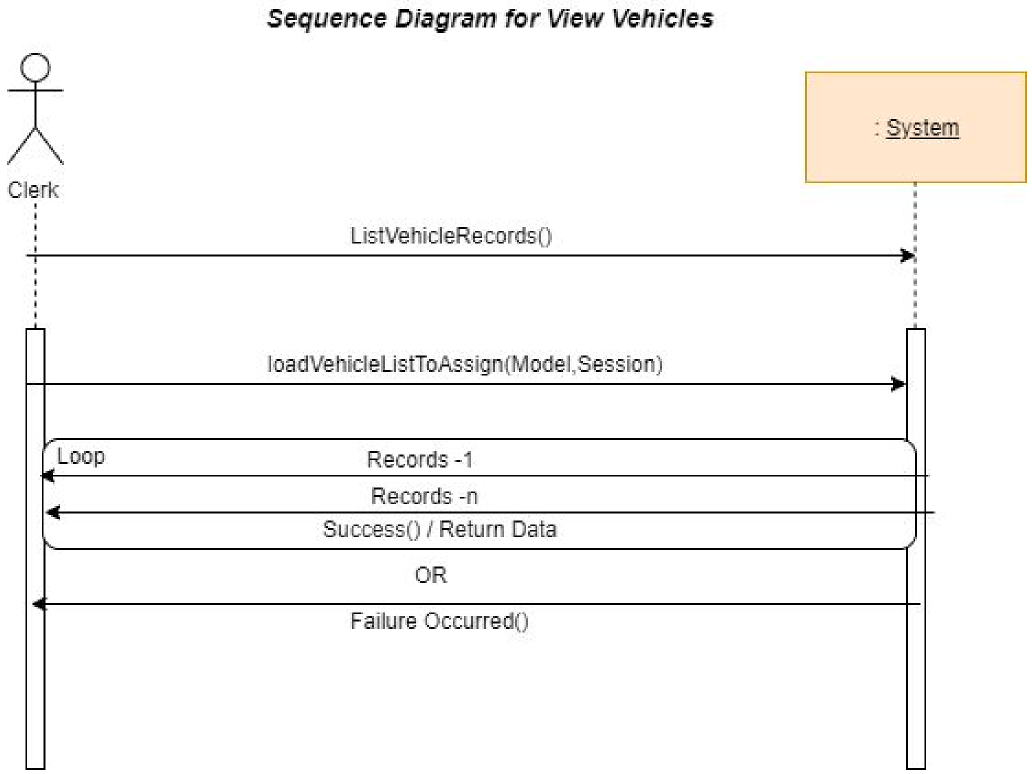


Figure 6.1:UC-3

4. UC 4: Search Catalog

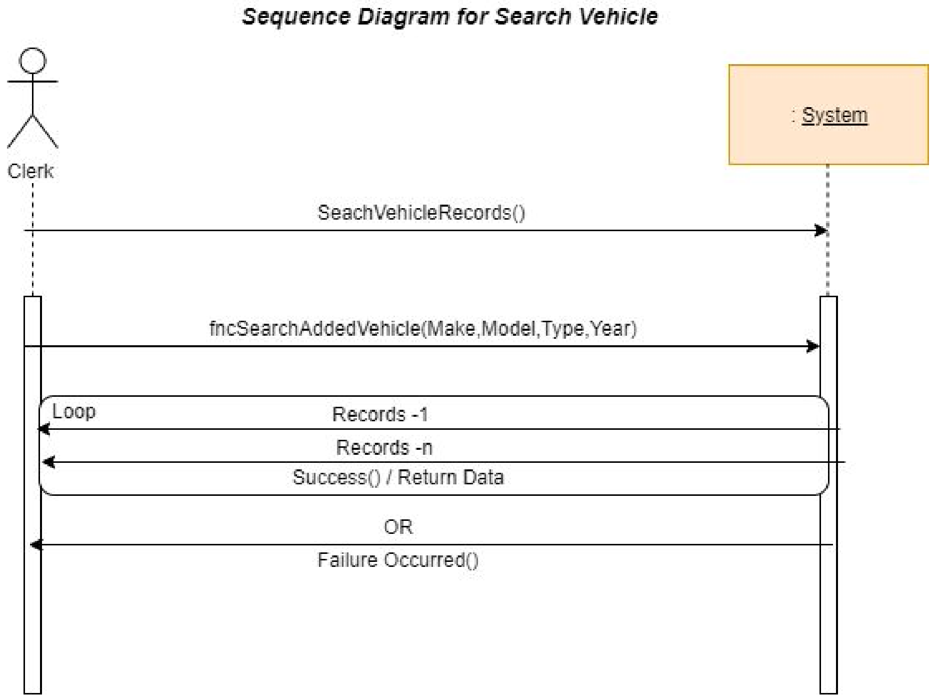


Figure 6.2:UC-4

5. UC 5: View Vehicle Details

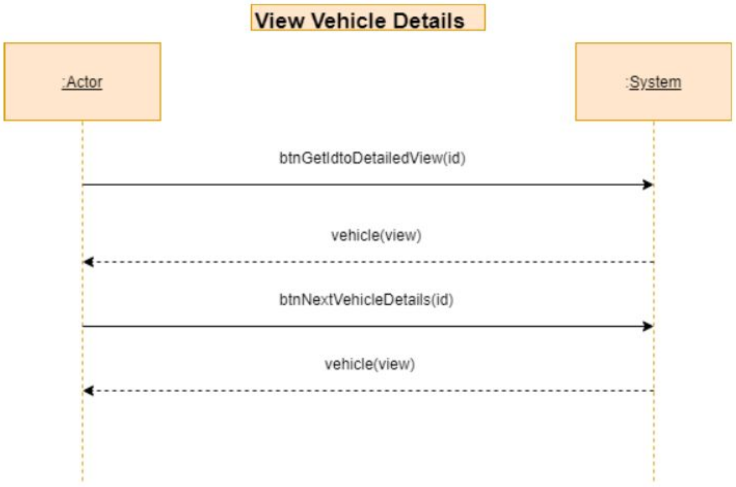


Figure 6.3:UC-5

6. UC 6: Book Vehicle

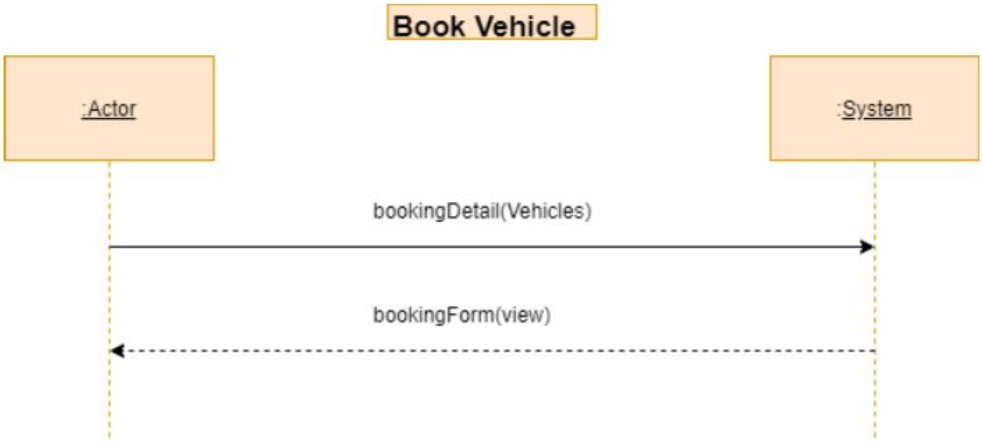


Figure 6.4:UC-6

7. UC7: Manage Client Records

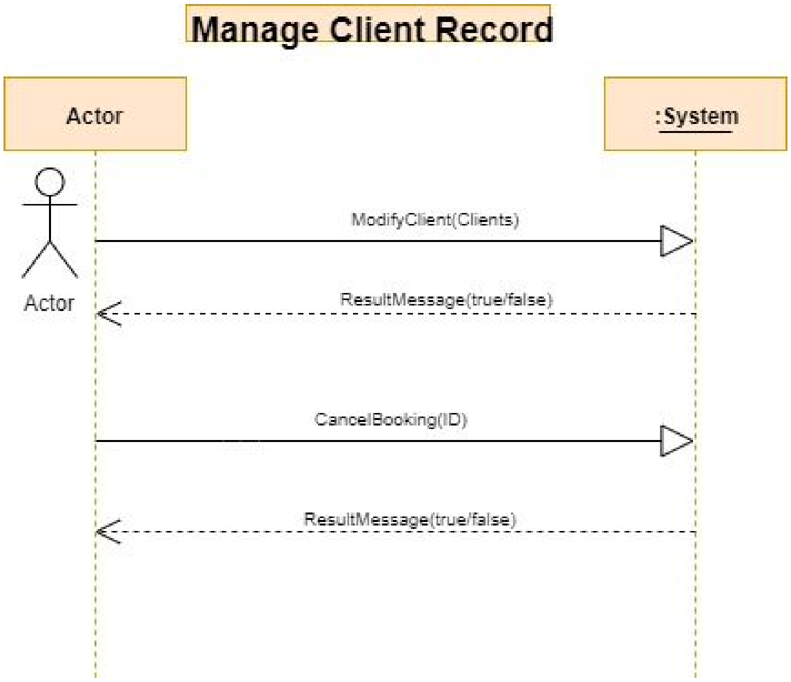


Figure 6.5:UC-7

8. UC8: View Bookings

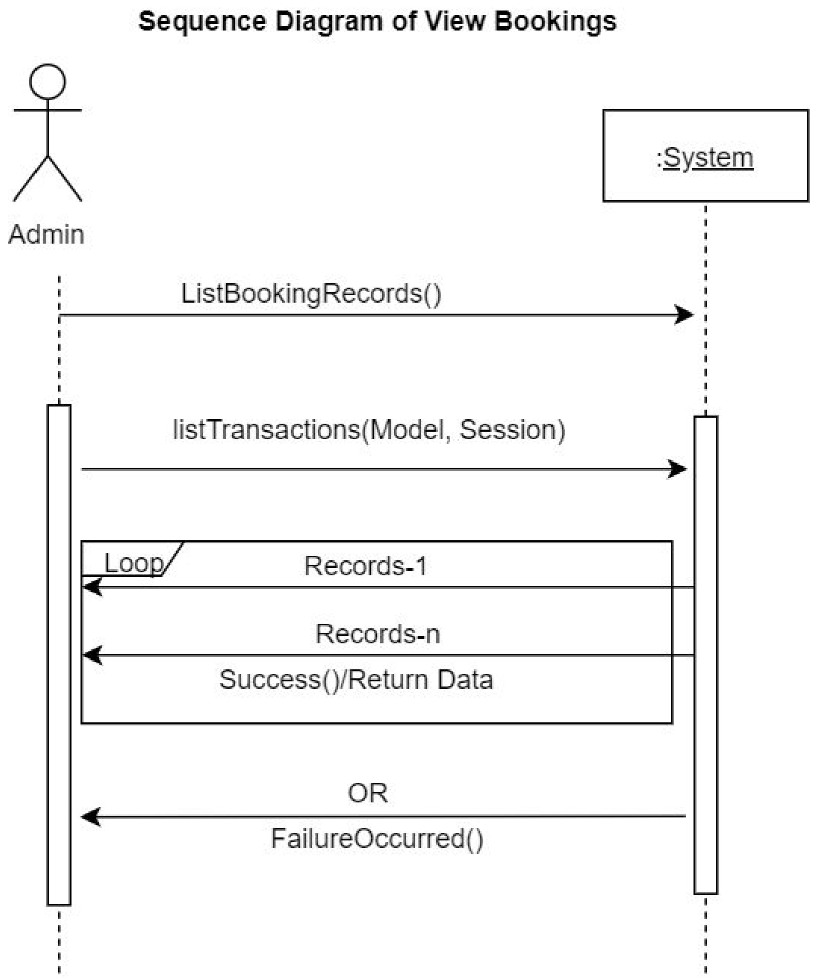


Figure 6.6:UC-8

9. UC9: Search Booking:

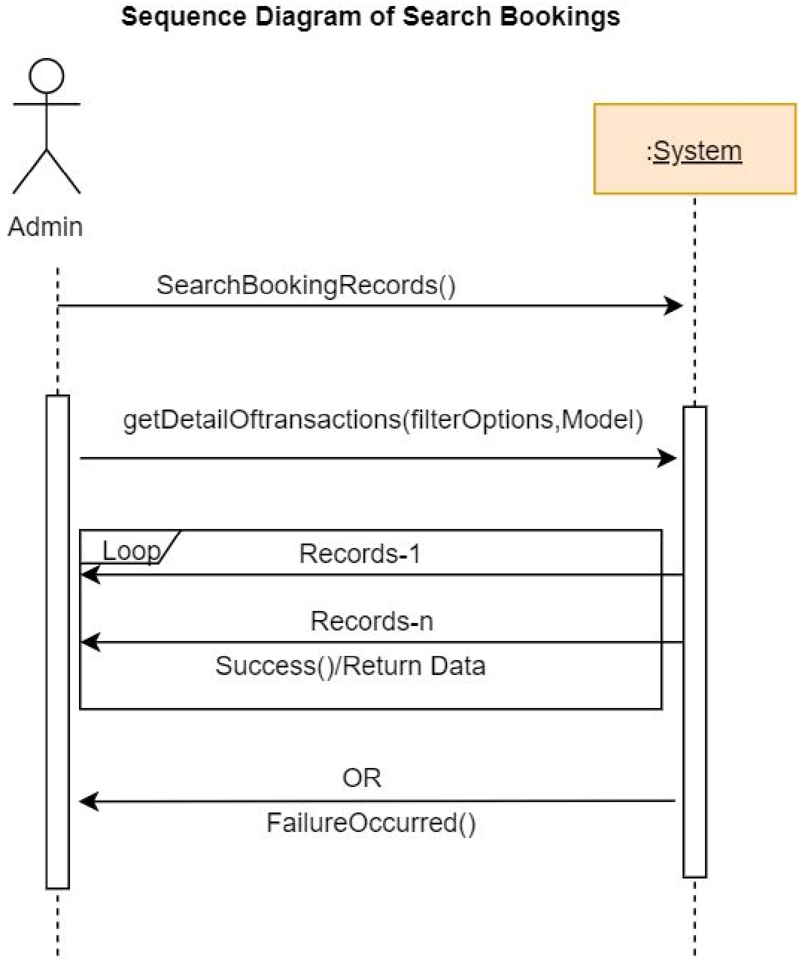


Figure 6.7:UC-9

10.UC10: Manage Vehicle Record

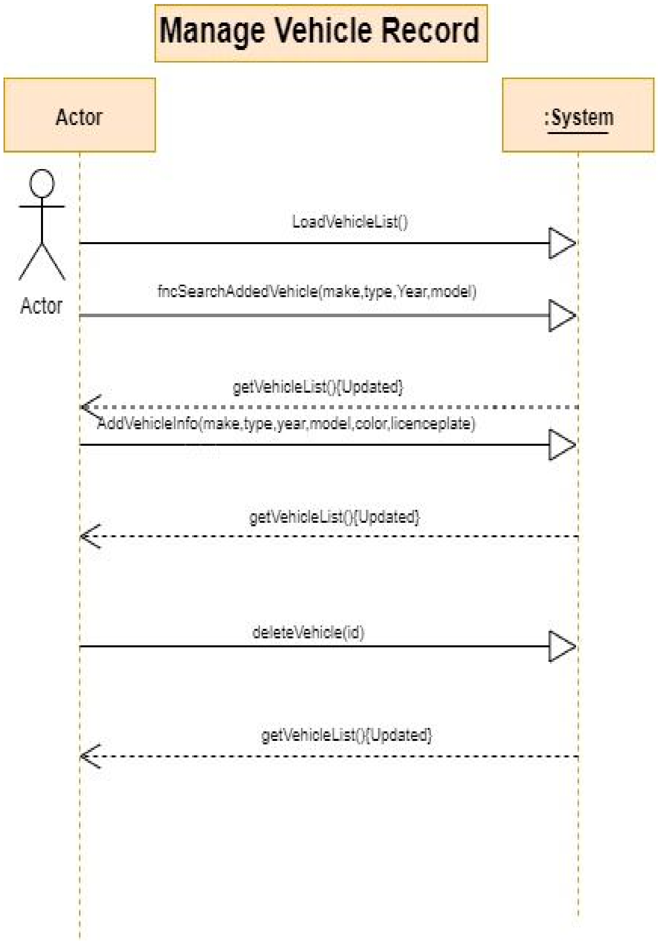

Manage Vehicle Record
Actor
:System
Actor
LoadVehicleList()
fncSearchAddedVehicle(make,type,Year,model)
getVehicleList(){Updated}
AddVehicleInfo(make,type,year,model,color,licenceplate)
getVehicleList(){Updated}
deleteVehicle(id)
getVehicleList(){Updated}

Figure 6.8:UC-10

System Domain Model

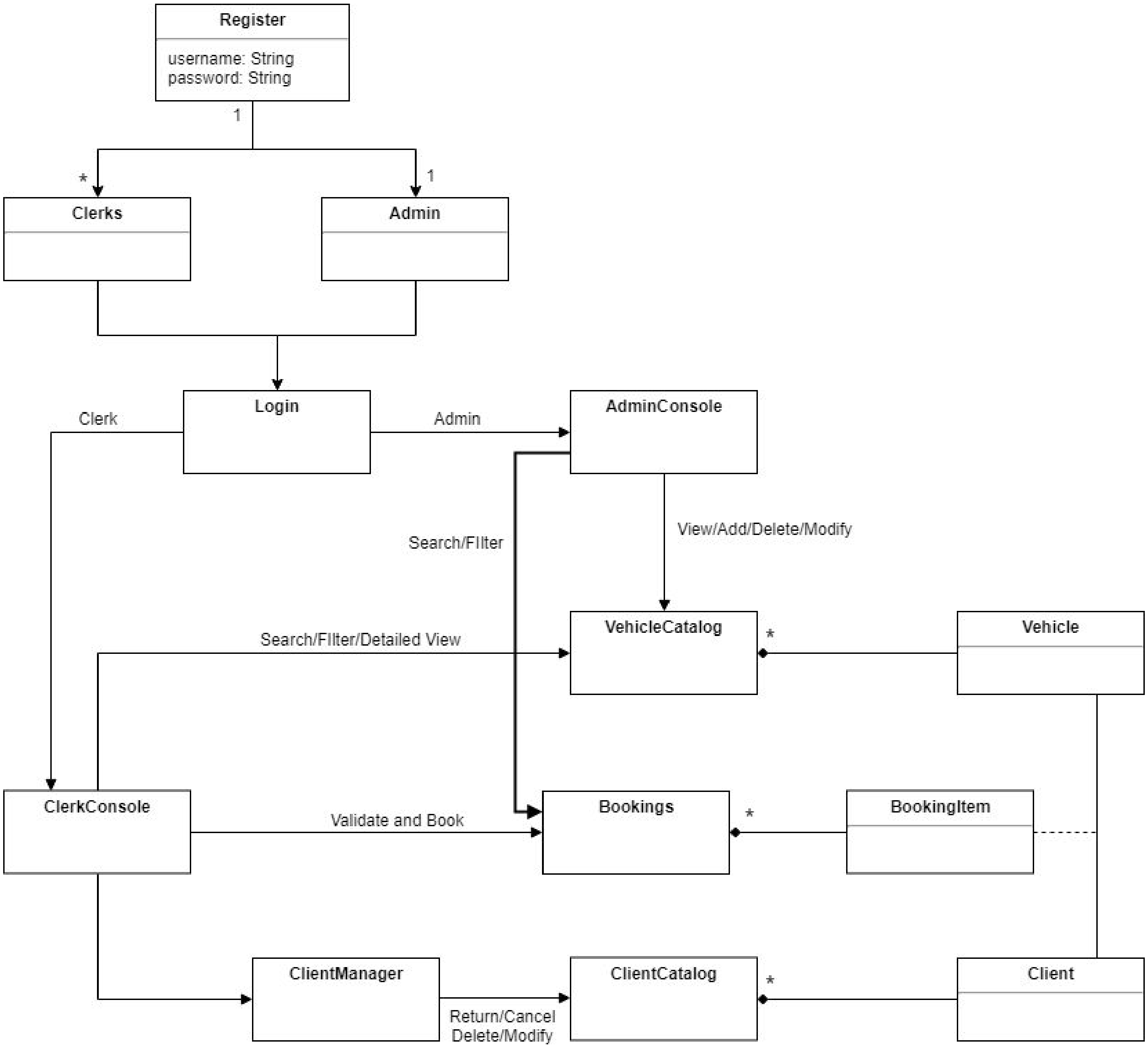
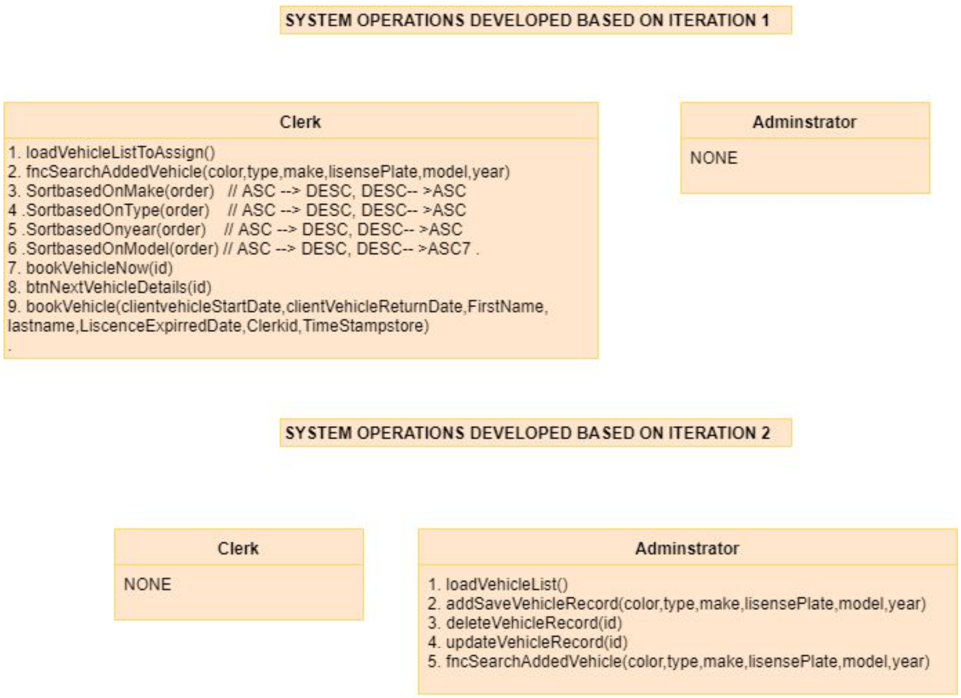


Figure 6.9:Domain Model

System Operations for vehicles



SYSTEM OPERATIONS DEVELOPED BASED ON ITERATION 1

Clerk

1. loadVehicleListToAssign()
2. fncSearchAddedVehicle(color,type,make,lisensePlate,model,year)
3. SortbasedOnMake(order) // ASC --> DESC, DESC-- >ASC
4 .SortbasedOnType(order) // ASC --> DESC, DESC-- >ASC
5 .SortbasedOnyear(order) // ASC --> DESC, DESC-- >ASC
6 .SortbasedOnModel(order) // ASC --> DESC, DESC-- >ASC7 .
7. bookVehicleNow(id)
8. btnNextVehicleDetails(id)
9. bookVehicle(clientvehicleStartDate,clientVehicleReturnDate,FirstName, lastname,LiscenceExpirredDate,Clerkid,TimeStampstore)

.

Adminstrator

NONE

SYSTEM OPERATIONS DEVELOPED BASED ON ITERATION 2

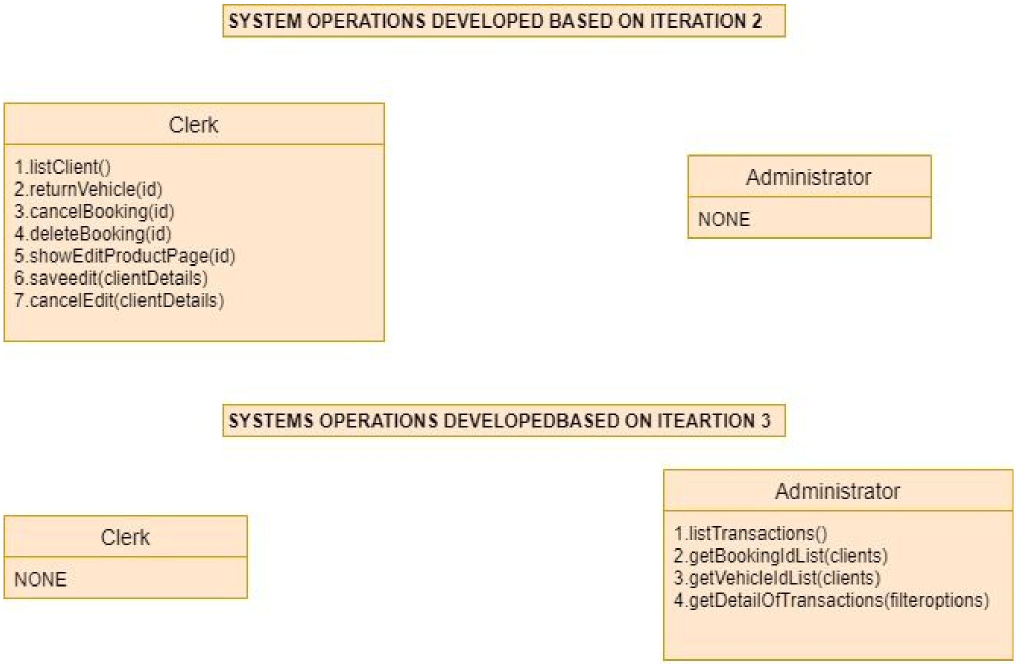
Clerk

NONE

Adminstrator

1. loadVehicleList()
2. addSaveVehicleRecord(color,type,make,lisensePlate,model,year)
3. deleteVehicleRecord(id)
4. updateVehicleRecord(id)
5. fncSearchAddedVehicle(color,type,make,lisensePlate,model,year)

Figure 7.0:System Operations for clients



SYSTEM OPERATIONS DEVELOPED BASED ON ITERATION 2

Clerk

1.listClient()
2.returnVehicle(id)
3.cancelBooking(id)
4.deleteBooking(id)
5.showEditProductPage(id)
6.saveedit(clientDetails)
7.cancelEdit(clientDetails)

Administrator

NONE

SYSTEMS OPERATIONS DEVELOPEDBASED ON ITEARTION 3

Clerk

NONE

Administrator

1.listTransactions()
2.getBookingIdList(clients)
3.getVehicleIdList(clients)
4.getDetailOfTransactions(filteroptions)

Figure 7.1:System Operations

Operation Contracts

Contract OC-001	loadVehicleListToAssign
Operation	loadVehicleListToAssign (Model, HttpSession session)
Cross Reference	**UC3: View Catalog**
Pre-Condition	After Login Clerk should able to view all the available/not available list of vehicles added by Administrator.
Post-Condition	Clerk can Search, Sort, View Details to book vehicle

Contract OC-002	fncSearchAddedVehicle
Operation	fncSearchAddedVehicle (String Make, String Type, String Model, Int Year)
Cross Reference	**UC 4:Search Catalog**
Pre-Condition	Table will get loaded automatically with vehicle list.

Post-Condition	Clerk can put parameters in textbox such as the Make, Model, Year, Type and search the vehicle what they are looking for.

Contract OC-003	sortbasedOnMake sortbasedOnModel sortbasedOnyear sortbasedOnType

Operation	sortbasedOnMake(Model model)
	sortbasedOnModel(Model model)
	sortbasedOnyear(Model model)
	sortbasedOnType(Model model)

Cross Reference	**UC4: Search Catalog**
Pre-Condition	Dataset/Vehicle record should be loaded into the table
Post-Condition	Clerk can click on any of the button name as follows: 1.Sort by make : (ASC->DESC),(DESC-->ASC) 2.Sort by model : (ASC->DESC),(DESC-->ASC) 3.Sort by year : (ASC->DESC),(DESC-->ASC) 4.Sort by Vehicle type : (ASC->DESC),(DESC-->ASC)

Contract OC-004	btnGetIdtoDetailedView
Operation	btnGetIdtoDetailedView(Long id)
Cross Reference	**UC5:View Vehicle Details**
Pre-Condition	Click on the record for which you wanted to view the details in Detailed view.
Post-Condition	Click the next button to view the details of next vehicle record.

Contract OC-005	bookingDetailsRental bookingDetailsReservation
Operation	bookingDetailsRental (Model model) bookingDetailsReservation (Model model)
Cross Reference	**UC6: Book Vehicle**
Pre-Condition	Vehicle should be available too book on Detailed View otherwise book Now button is disabled on Detailed view Screen.
Post-Condition	If clicked on <<book Now>> button the vehicle will get booked for the client and saves all the info which is shown on Client/ booking Catalog Screen for Modify/Return.

Contract OC-006	loadVehicleList
Operation	loadVehicleList (Model model, HTTP session)
Cross Reference	**UC5: View Vehicle Details**
Pre-Condition	After Login Administrator should able to view all the of vehicles added by Administrator.
Post-Condition	Administrator can Add/Edit/Delete the vehicle which can be viewed and booked by clerk for a client.

Contract OC-007	fncSearchAddedVehicle
Operation	fncSearchAddedVehicle (String Make, String Type, String Model, Int Year)
Cross Reference	**UC9: Search Bookings**
Pre-Condition	Table will get loaded automatically with vehicle list.

Post-Condition	Administrator can put parameters in textbox such as the Make, Model, Year, Type and search the vehicle what they are looking for.

Contract OC-008	addVehicleRecord
Operation	addVehicleRecord (Model model)
Cross Reference	**UC10: Manage Vehicle Record**
Pre-Condition	Load the Vehicle master table.
Post-Condition	After <<Save>> button administrator should be redirected back to vehicle master page.

Contract OC-009	updateVehicleRecord
Operation	updateVehicleRecord(long id)
Cross Reference	**UC10: Manage Vehicle Record**
Pre-Condition	Load the Vehicle master table.

Post-Condition	After clicking on <<Edit>> button administrator will update the fields and click on <<update>> button

Contract OC-010	deleteVehicleRecord
Operation	deleteVehicleRecord(long id)
Cross Reference	**UC10: Manage Vehicle Record**
Pre-Condition	Load the Vehicle master table.
Post-Condition	After <<Delete>> button administrator should delete the current vehicle record.

Contract OC-011	returnVehicle(long id)
Operation	returnVehicle(long id)
Cross Reference	**UC 7: Manage Client Record**
Pre-Condition	Load client manager table.
Post-Condition	After <<Return>> button clerk should set the current client has returned the vehicle and make available for future bookings.

Contract OC-012	cancelVehicle(long id)
Operation	cancelVehicle(long id)
Cross Reference	**UC 7: Manage Client Record**
Pre-Condition	Load client manager table.
Post-Condition	After <<Cancel>> button clerk should set the current client has cancelled the vehicle and make available for future bookings.

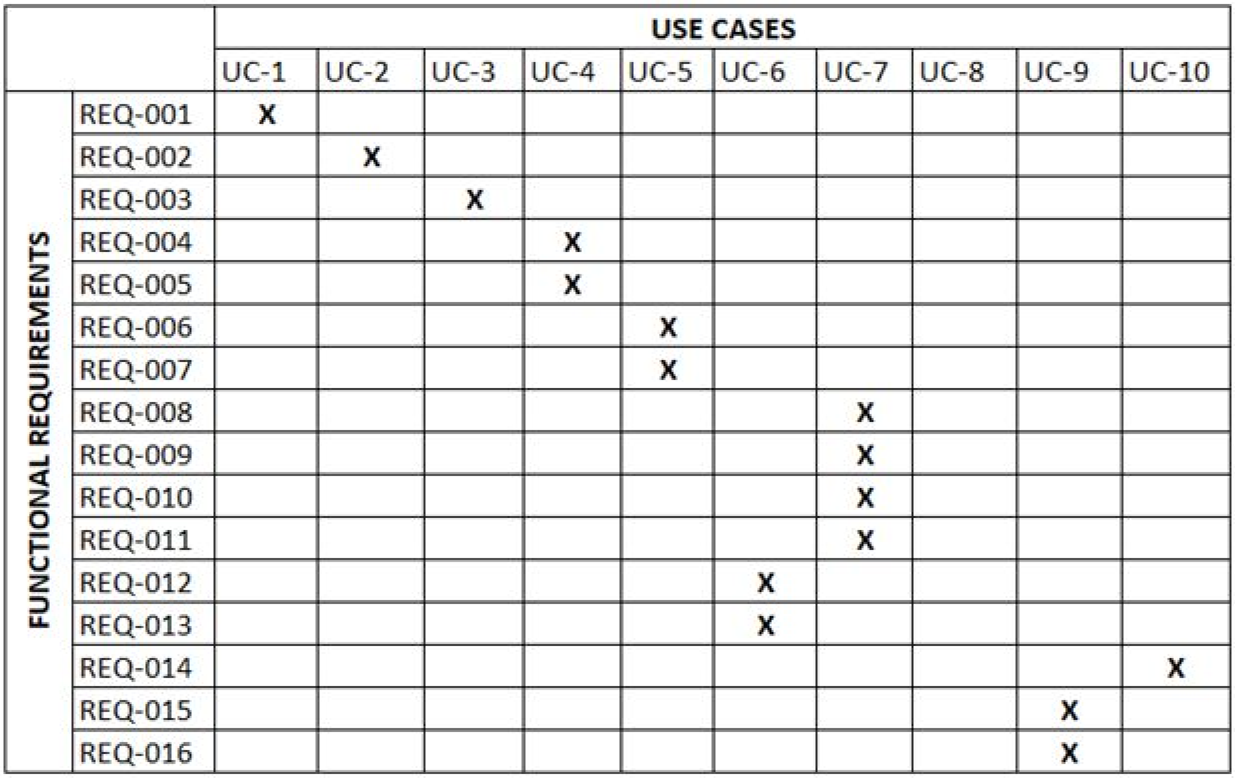
Contract OC-013	deletelVehicle(long id)
Operation	deletelVehicle(long id)
Cross Reference	**UC 7: Manage Client Record**
Pre-Condition	Load client manager table.
Post-Condition	After <<Delete>> button clerk removes current client record and makes the respective vehicle available for future bookings.

Contract OC-014	modifyClient(long id)
Operation	modify(long id)
Cross Reference	**UC 7: Manage Client Record**
Pre-Condition	Load client manager table.
Post-Condition	After <<Modify>> button clerk is directed to modify screen which allows him to change the selected clients details.

Contract OC-015	saveClient(long id)
Operation	save(long id)
Cross Reference	**UC 7: Manage Client Record**
Pre-Condition	Load client modify page.
Post-Condition	After <<Save>> the changes made by clerk are saved and updated in the database, he is redirected to client manager screen where he can view the changes.

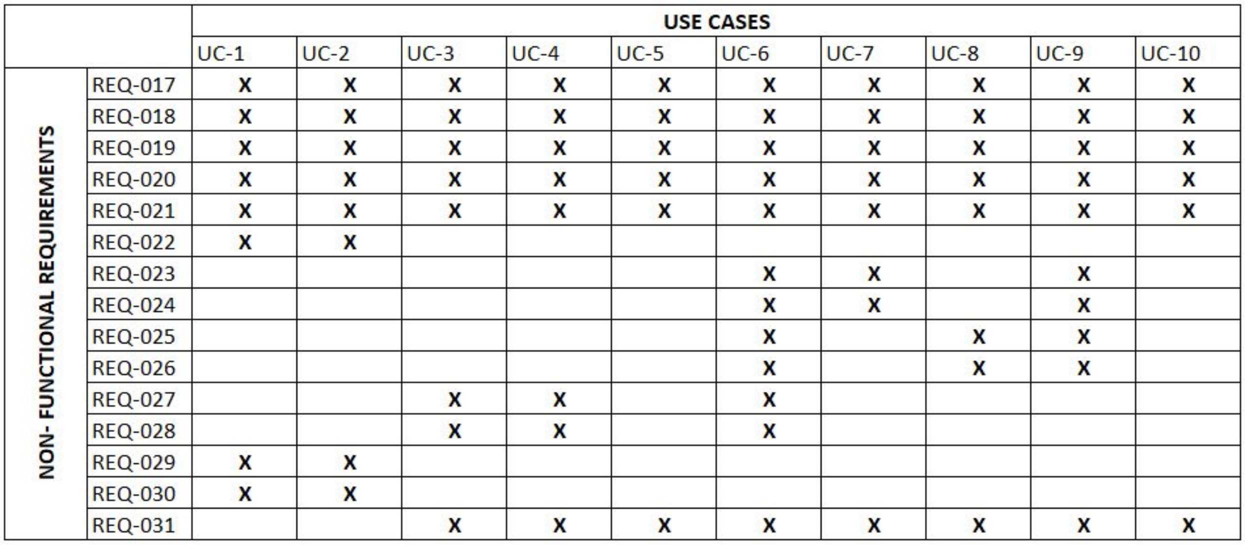
Contract OC-015	cancelModify(long id)
Operation	cancel(long id)
Cross Reference	**UC 7: Manage Client Record**
Pre-Condition	Load client modify page.
Post-Condition	After <<Cancel>> the changes made by clerk are discarded, he is redirected to client manager screen.

Use Case and Functional Requirements Mapping



		USE CASES									
		UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10
FUNCTIONAL REQUIREMENTS	REQ-001	X									
	REQ-002		X								
	REQ-003			X							
	REQ-004				X						
	REQ-005				X						
	REQ-006					X					
	REQ-007					X					
	REQ-008							X			
	REQ-009							X			
	REQ-010							X			
	REQ-011							X			
	REQ-012						X				
	REQ-013						X				
	REQ-014										X
	REQ-015									X	
	REQ-016									X	

Non-Functional Requirements Coverage



		USE CASES									
		UC-1	UC-2	UC-3	UC-4	UC-5	UC-6	UC-7	UC-8	UC-9	UC-10
NON- FUNCTIONAL REQUIREMENTS	REQ-017	X	X	X	X	X	X	X	X	X	X
	REQ-018	X	X	X	X	X	X	X	X	X	X
	REQ-019	X	X	X	X	X	X	X	X	X	X
	REQ-020	X	X	X	X	X	X	X	X	X	X
	REQ-021	X	X	X	X	X	X	X	X	X	X
	REQ-022	X	X								
	REQ-023						X	X		X	
	REQ-024						X	X		X	
	REQ-025						X		X	X	
	REQ-026						X		X	X	
	REQ-027			X	X		X				
	REQ-028			X	X		X				
	REQ-029	X	X								
	REQ-030	X	X								
	REQ-031			X	X	X	X	X	X	X	X