FINAL SEMESTER ASSESSMENT (FSA)
B.TECH. (CSE)
VI SEMESTER

UE18CS355 – OBJECT ORIENTED ANALYSIS AND DESIGN WITH SOFTWARE ENGINEERING LABORATORY

PROJECT REPORT

ON

Car Rental Management System

SUBMITTED BY

NAME	SRN
1) Akash Yadav	PES2201800415
2) Naveen Suresh	PES2201800508
3) Saiprakash L Shetty	PES2201800730

JANUARY – MAY 2021

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
EC CAMPUS,
BENGALURU – 560100, KARNATAKA, INDIA

TABLE OF CONTENTS

ABSTRACT

Transport facility is a headache for those people who do not have any personal transport. On occasions like weddings, Vacation, house shifting, and tours outside their area and in many other situations they feel the necessity of a vehicle to sort out the problems. So if it is possible to design or develop a web based application for availing transport whenever and wherever possible, then it will be beneficial for both renter and transport provider. Now a days, by some clicks only, we can get whatever you want at home.We already know about the online shopping, e-banking etc. Similarly, The Car Rental System is the facility to book cars within a few clicks only. Some people can not afford to have a car, for those people this system becomes very helpful. This system includes various cars, as per the customer order and comfort, it places the order and delivers the car as per the location around the country.

1.SOFTWARE REQUIREMENTS SPECIFICATION

1. INTRODUCTION

1.1 Purpose

The purpose of the car rental system is to design a Client friendly system that enables clients to check for availability of vehicle and book/reserve a vehicle. System that enables clients to pay their car rent and develop a system that stores bookings and reservation information as well as payment history to help the organization keep track of transactions. The individual who needs a car must contact a rental car company and contract out for a vehicle. This system increases customer retention and simplifies vehicle and staff management.

1.2 Intended Audience

This project is intended to serve the client that allows them to rent a car despite the fact that they don't have access to their own vehicle or don't own a vehicle at all. Allows the customer to book space for a group in the case of weddings or corporate meetings. Helps managing staff to maintain databases and control transactions.

1.3 Product Scope

This project traverses a lot of areas ranging from business concept to computing field, and requires several researchers to be able to achieve the project objectives. The area covers include:

● Car rental industry: This includes study on how the car rental business is being done, process involved and opportunity that exist for improvement.
● General Client and company’s staff will be able to use the system effectively.
● The system can be accessed 24/7 in a Client-friendly manner.
● Java programming language is incorporated for the development of the application.

1.4 References

● System, Online. 'Online Car Rental System'. Academia.edu. N.p., 2015. Web. 9 June 2015.
● Scribd.com, Online. '49930505 Car Rental System Project Report'. N.p., 2015. Web. 9 June 2015.
● Scribd.com, Online. 'Car Rental System Documentation'. N.p., 2015. Web. 9 June 2015.
● Freelancer, Online. 'Project Documentation Car Rental Company Software Development Freelancers and Jobs - Freelancer'. N.p., 2015. Web. 9 June 2015.
● Slideshare.net, Online. 'Zook Car Rental System Project'. N.p., 2015. Web. 9 June 2015.

2. Overall Description

2.1 Product Perspective

Car rental systems are used temporarily for a period of time with a fee. This system serves both foreign and local clients with the rental cars categorized based on their vehicle number. This organization uses a manual system for reserving, renting, registering and to keep record of all the rental activities and customer information. The detailed existing functions are listed as follows:

● Car reservations can be made through customer accounts on the application.
● During renting a car the customer's personal information and payments status are filled in order to hold legal contact between the customer and organization for renting the vehicle.
● The organisation offers flexibility to their clients in terms of the duration of renting a car of their choice.

2.2 Product Functions

● Car rental management: It provides car reservation facilities online. Client can visit the website and choose a car of their choice and booking can be done if it is feasible to their requirement.
● Payment system: Order cancellation, order finalize, these are all done and maintained by the admin/owner of the company.
● Admin: Maintain the data about the cars and their status.

2.3 Client Classes and Characteristics

● Admin
    1. Admin can login to the system.
    2. Verify the car information database.
    3. Generate price strategy.
    4. It updates the database.
    5. Update the database.
    6. Handle Payments

● Customer
    1. Clients can login to the system.
    2. Place the order

2.4 Operating Environment

Processor: Intel Core i3 and above
RAM: Minimum 4GB
OS: Windows 10
Application: Java, NetBeans, MySQL

2.5 Design and Implementation Constraints

● The application will use xml, sql as main web technologies and Java as programming language to code and execute programs.
● Since the Car Rental system is a web-based application, internet connection must be established.
● The Car Rental System will be used on PCs and will function via internet or intranet in any web browser.
● System should satisfy the requirement and have all the software installed to run efficiently and smoothly.

2.6 Assumptions and Dependencies

1) Regulatory Policies: Each center Client has an account created and authenticated by the

admin once the registration is complete. This application can be accessible within the company's intranet. Each Client has to first login and this will be done automatically. No Clients can share their Clientname and password to each other.

2) Hardware Limitations: There is no limitation in the operating system in which the Car Rental System will work. However, the system needs to meet the requirements to run a Java program and required software to code and run them.

3) There will be no Client manuals, online help or tutorials as it is made as simple as web beginners can also use it easily with best GUI functionality.
We have made the following assumptions to the development of the CRS:
● We assume that the development team has all the software's and tools that will be required to complete this project on time.
● We assume that at the time of the system deployment, the branches will have computers with the minimum system requirements as per required by the software for efficient runtime.
● We assume that our project team has been finalized and no new member will be added nor a present member will leave the group in the middle of the project.
● We assume that all the requirements we get from our sources are correct and achievable.
These are a few dependencies we have for the system
● The system has efficient runtime on Windows platform.

3. External Interface Requirements

3.1 Client Interfaces

There are 2 Client Interfaces in this system, Customer Interface, Admin Interface

3.1.1 Client Interface:

This is the interface that will be provided to all Clients on their systems. The elements included are:
1. Register as a member -The customer is initially given an option to either choose to register as a new member or log in.
2. Login - If the customer already has an account, they can log in using their registered credentials and be able to access the application, and make use of the features.
3. Make reservation - The cars available at that instance are displayed along with respective details. After placing an order and completing the payment, complete order details are displayed.

3.1.2 Admin Interface:

This is the interface at the backend (agency) which prepares the bookings and reservations. All the car reservations are sent to the staff and managed accordingly.
The elements included are:
1. Add a new car - New cars can be added to the list as an when there is an addition to the agency’s inventory.
2. Update - The staff has the option to remove or modify pre-existing car details.
3. Process Rental - Once a car is returned, this element will take care of all the simultaneous

updates that need to happen at the backend to make the car available for renting again.

This interface allows the admin to have an overall view of all the records along with necessary statistics to give an insight to the agency's sales, service and growth. This can help the administration make necessary decisions in the benefit of the agency.

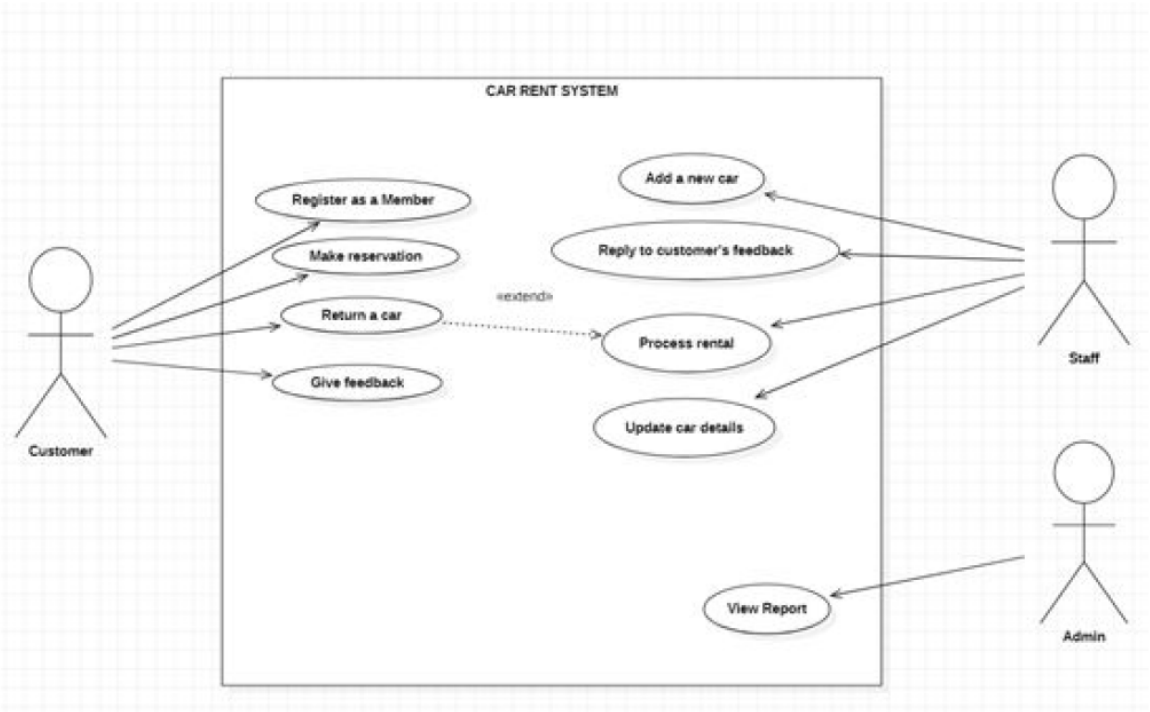
3.2 Software Interfaces

The Client interface will be simple, using terminology commonly understood by intended Clients
of the system. The system will have a simple interface, consistent with standard interface, to eliminate the need for Client training of infrequent Clients.
The software consists of the following technologies:
1. The Java application
2. The MySQL database
3. The database should remain consistent at all times in case of an error.

3.3 Communications Interfaces

1. This application will be enabled to the Clients through GUI.
2. Security will be enforced via encryption on communications involving payments.
3. A Client can make use of the services only if they are a authorised Client (verified by the admin)
4. The client data is secured and not shared with any third party organisations. (Except payments)

4. Analysis Models



5. System Features

5.1 Main Menu Tab

5.1.1 Description and Priority

This feature tab will allow clients to view various cars available on rent, search through applying filters and also view details and reviews on the specific car. This tab is where the Client will spend most of their time while using the application thus making it the highest prioritized component.

5.1.2 Stimulus/Response Sequences

1. The Client clicks on a car option.
2. A tab opens up showing details about the selected car along with a “back” and “make a reservation” option. These are statements of services the system should provide, how the system should react to particular inputs, and how the system should behave in particular situations. It specifies the application functionality that the developers must build into the product to enable Clients to accomplish their tasks.

5.1.3 Functional Requirements

ID REQUIREMENT
MM1 The customer should be able to view the menu with a display of all the available cars at that instance when they LOG IN.
MM2 The Main Menu Tab must always be updated in real time depending on reservations and when cars are returned.
MM3 The admin should be able to explicitly make changes to the cars displayed in this tab.

5.2 Payments Tab

5.2.1 Description and Priority

This feature tab will appear once a customer is sure about renting a particular car model. When an online payment method is chosen a secure gateway is established and choices between Card or Cash are given to the Client. This is a high priority feature.

5.2.2 Stimulus/Response Sequences

1. The customer reviews the car details they have chosen and fills in required details related to order and then clicks on “Pay”.
2. An option to choose between Online payment options or COD is given.
3. Upon successful transaction in case of an online payment, the reservation is confirmed.
4. When COD is chosen, the order is confirmed and payment is collected at the time of vehicle pickup .

5.2.3 Functional Requirements

ID REQUIREMENT
P1 Payment details shall not be lost under any circumstances.
P2 Application shall be capable of interfacing with a register to facilitate the accurate processing of a payment.

5.3 Reservations

5.3.1 Description and Priority

This feature tab will allow clients to proceed with reserving their car of choice once they have made their mind. This is of medium priority in terms of implementation but of high organisation value as it is what makes up for the organisation's profit as well as a high priority when it comes to information security.

5.3.2 Stimulus/Response Sequences

1. Client selects make a reservation once they make up their mind for a particular car.
2. The client is required to follow some protocols before getting a confirmation for their reservation. Once that is done, the client is directed to the payments tab.

5.3.3 Functional Requirements

ID REQUIREMENT
R1 The system must allow the customer to register for reservation.
R2 The system shall allow the customer to view a detailed description
of a particular car.
R3 The system must allow the customers to select a specific car using a different search category while booking.
R4 The system shall allow the employee to view reservations made by customers

6 . Other Nonfunctional Requirements

6.1 Performance Requirements

● Ability to maintain mass amount of customers on the website at once without crashing
● Speedy performance / transmission of data
● Send any emails immediately
● Being logged in should allow for customers to quickly make payments without reentering information, and allow for any potential registered perks the company may have
● Have a quick recovery time if anything were to go wrong

6.2 Safety Requirements

The system provides Clientname and password to prevent the system from unauthorized access. The subsystem should provide a high level of security and integrity of the data held by the system, only authorized personnel of the company can gain access to the company's secured page on the system; and only Clients with valid password and Clientname can login to view Client's page.

6.3 Security Requirements

● Secure any transmissions of private information between the customer and the company
● Prevent any potential threats such as SQL injections through the forms or search boxes.
● Prevent third party Clients at administrative level

6.4 Software Quality Attributes

● Maintain a Client-friendly environment that is visually appealing
● Easy to see and use navigation
● Maintain readable content

● Searching cars should be accessible to people who are and are not logged in
● Selecting and making a payment should be available to customers who are and are not logged in

6.5 Business Rules

● Information stored for each booking includes customer, car, date of hire.
● If a vehicle is available, the customer's details are recorded (if not stored already) and a new booking is made.
● Customers must pay for the vehicle at the time of hire.

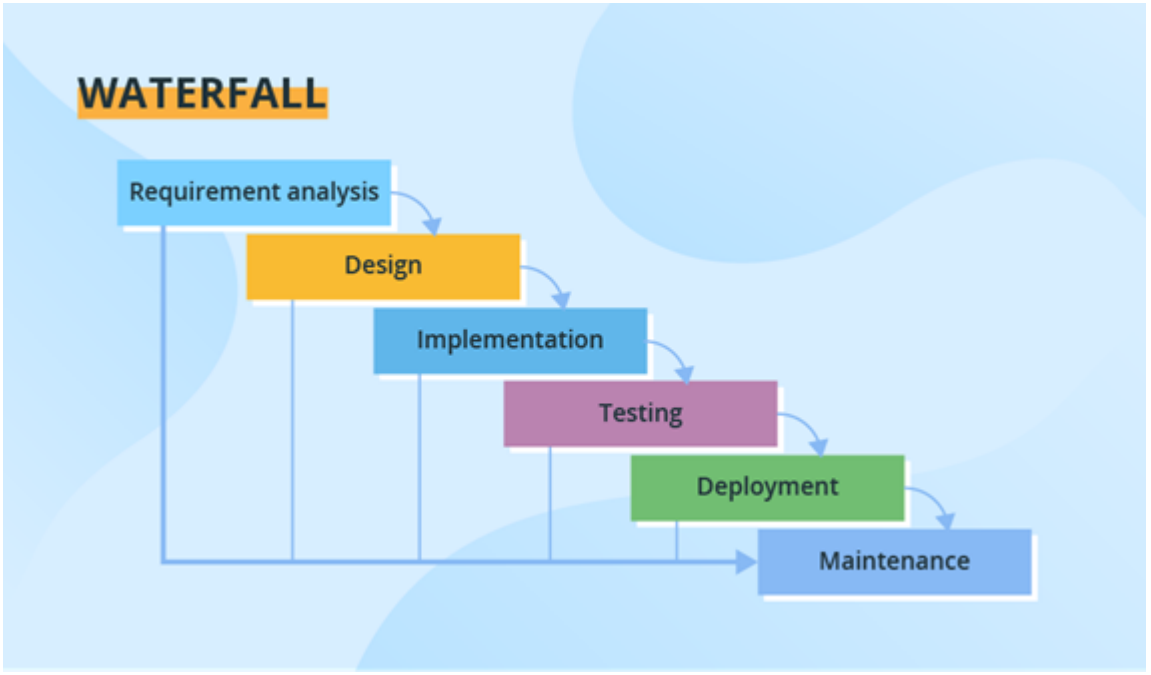
2. PROJECT PLAN

2. Software Development Methodology

2.1 MODEL

●After going through various models we have decided that the Waterfall Model is the Software development Lifecycle methodology we plan to follow for our project Online rental car management system.
●Through all development stages (analysis, design, coding, testing, deployment), the process moves in a cascade mode. Each stage has concrete deliverables and is strictly documented. The next stage cannot start before the previous one is fully completed for the development process.
●Plans regarding the number of iterations, the duration and the scope of each iteration are clearly defined in advance.

2.2 PHASES OF THE LIFE CYCLE



2.2.1 Advantages

●Simple small or mid-sized projects with clearly defined and unchanging requirements (small company website development).
●Projects that must adhere to multiple rules and regulations (healthcare projects).
●Projects where a well-known technology stack and tools are used.

2.2.2 Disadvantages

●No ability to see and try software until the last development stage is finished, which results

in high project risks and unpredictable project results. Testing is often rushed, and errors are costly to fix.
●Software requirements cannot be re-evaluated further in development.

2.3 Tools and Techniques

2.3.1 Design -StarUML
StarUML is an open source software modeling tool that supports the UML (Unified Modeling Language) framework for system and software modeling
StarUML supports the following diagram types that we are going to be using.
●Use Case Diagram
●Class Diagram
●Activity Diagram

2.3.2 Version Control - GitHub
Version control allows us to keep track of your work and helps us to easily explore the changes that we have made, be it data, coding scripts, notes, etc.
Each file on GitHub has a history, making it easy to explore the changes that occurred to it at different time points. You can review other people's code, add comments to certain lines or the overall document, and suggest changes.
Having a GitHub repo makes it to keep track of collaborative and personal projects - all files necessary for certain analyses can be held together and people can add in their code, graphs, etc. as the projects develop. You can also ask certain users to review your code. For personal projects, version control allows you to keep track of your work and easily navigate among the many versions of the files you create, whilst also maintaining an online backup.

2.3.3 Development
We will make use of PHP and Apache XAMPP servers for the purpose of log in and sign up.
MySQL will be used to store data of customers,cars etc. in a database.
Java is the basic language that we are going to use to code the front end part of our project.

2.3.4Planning - Microsoft Excel
Based on the Gantt chart model, this project plan template in Excel uses a simple visual representation to show how a project will be scheduled and managed over time from January to April.

2.4.Project Deliverables

2.4.1 Project Management

-Project Charter : Reuse
The project scope and structure has been discussed in our week 1 submission and will be used for the future development.
-Management Plan: Build
At its core, a management plan will define the approach and process the team will use to manage the project according to scope. This is to ensure that the long term trust of the

software in terms of accessibility and reusability is projected well to the project stakeholders.

2.4.2 Initiation

-Project planning - Build
Given the timeline and resources, a project plan has been made.

2.4.3Requirement Analysis

-Requirement Gathering - Reuse
Customer requirements taken into consideration during the project build of similar apps in the market like Zoomcar. This approach is inherited by this project in order to understand our clients.
-Requirements Analysis - Reuse
The gathered requirements are analyzed for feasibility as done by Zoomcar.
-Security Planning - Reuse
Basic security in terms of data integrity is maintained. Any future security threats can be handled by further modelling.

2.4.4 Design

-High Level Design - Build
A good software is one that can be easy to use by the clients (user) and which has self explanatory features.
-Proof Of Concept: Reuse
Zoomcar serves as a proof of concept as they are implemented and are feasible.
-Detailed Design - Build
A detailed explanation of the project features are documented including its action diagrams and use case diagrams which can be used by successive development and management teams to understand the project and its structure better.

2.4.5 Development

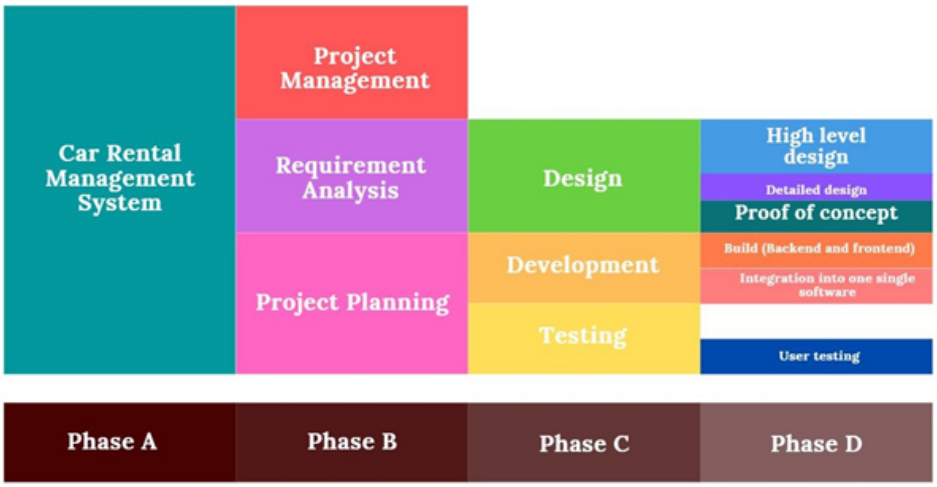
-Front end: Reuse
Front-end design can be inherited from the base design of similar apps in the present market.
-Backend: Build
The database is built from scratch based on the scope of the application.
-Integration Plan - Build
The frontend and backend would be combined using necessary Frameworks.

2.4.6 Testing

-User acceptance: Build
User acceptance testing would be done with a set of users in the proximity of the team and those who helped out with requirement gathering, and any changes to be made would be

taken from their responses.

2.5.WBS -Work Breakdown Structure



2.6: Do a rough estimate of the effort required to accomplish each task in terms of person months.

This project uses the Constructive Cost Model (CoCoMo model) specifically the organic method because the team size is small(3 members in a team) and the problem is well understood.

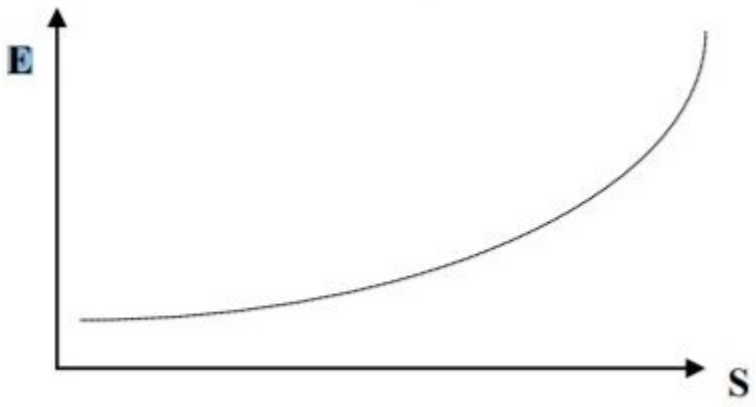
The effort required to accomplish the task is given by the following formula:

E = a(S)^b Person months Where:

E represents effort in person-months,

S is the size of the software development in KLOC and,

a and b are values dependent on the development mode, development mode: organic a = 2.4, b = 1.05



This graph demonstrates that the amount of effort(E) accelerates as size(S) increases.

The expected lines of code required for the project: Effort required for the different phases are as follows:

1)**Design**: Developing the login and register page using HTML and PHP.

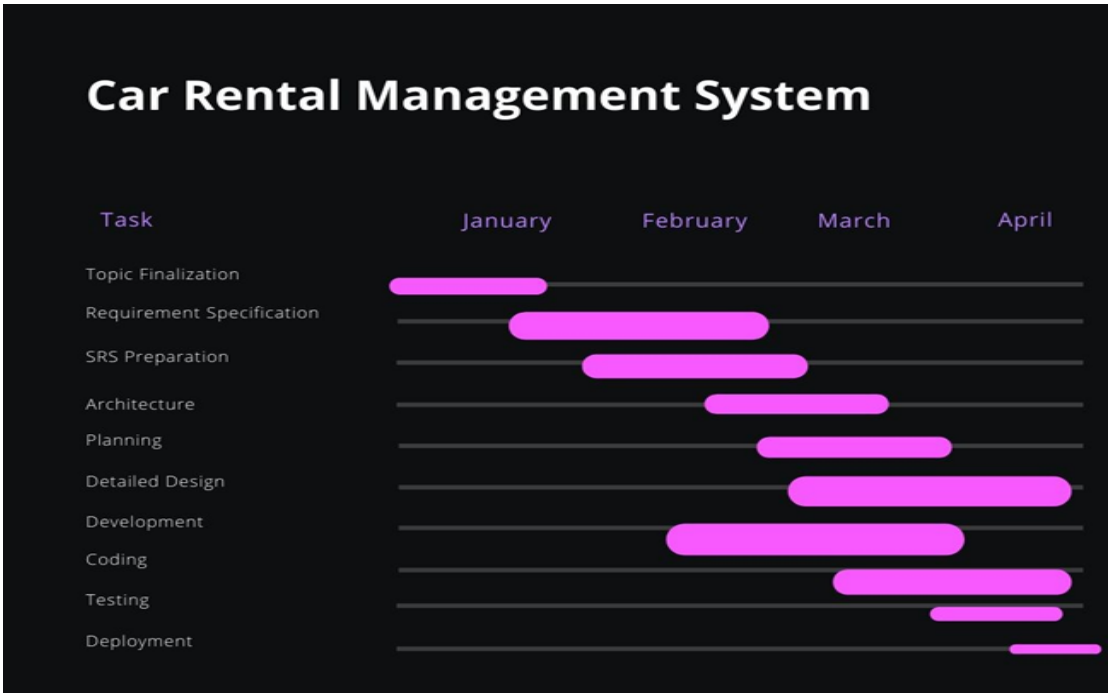
Effort = 2.4 * (95) ^ 1.05 = 287 PM

2)**Coding:** Using Java Programming language to build classes and functions.

Effort = 2.4 * (500) ^ 1.05 = 1638 PM

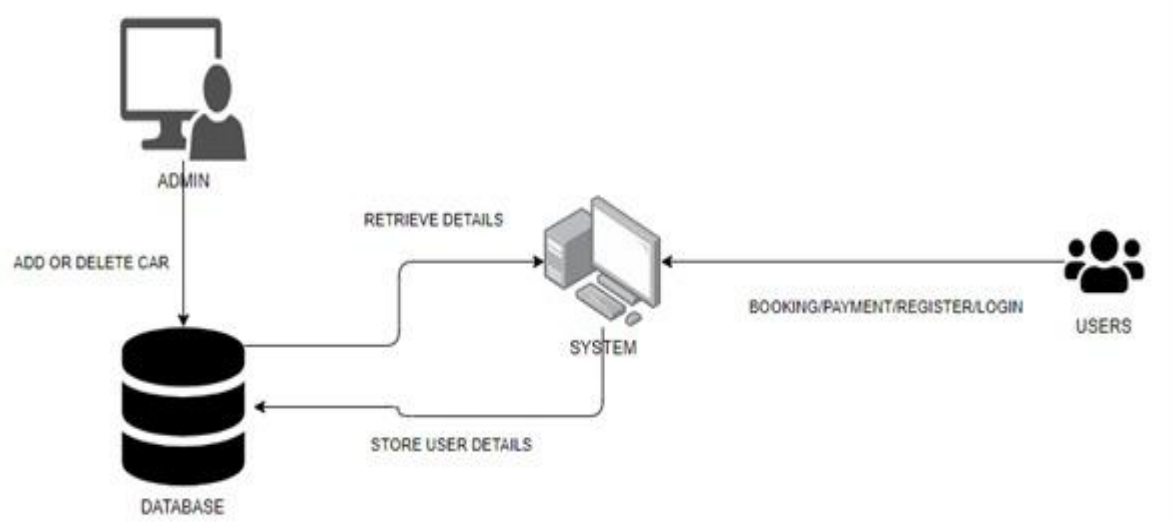
3)**Testing**:
Effort = 2.4 * (60) ^ 1.05 = 176 PM

6: Create the Gantt Chart for schedule.

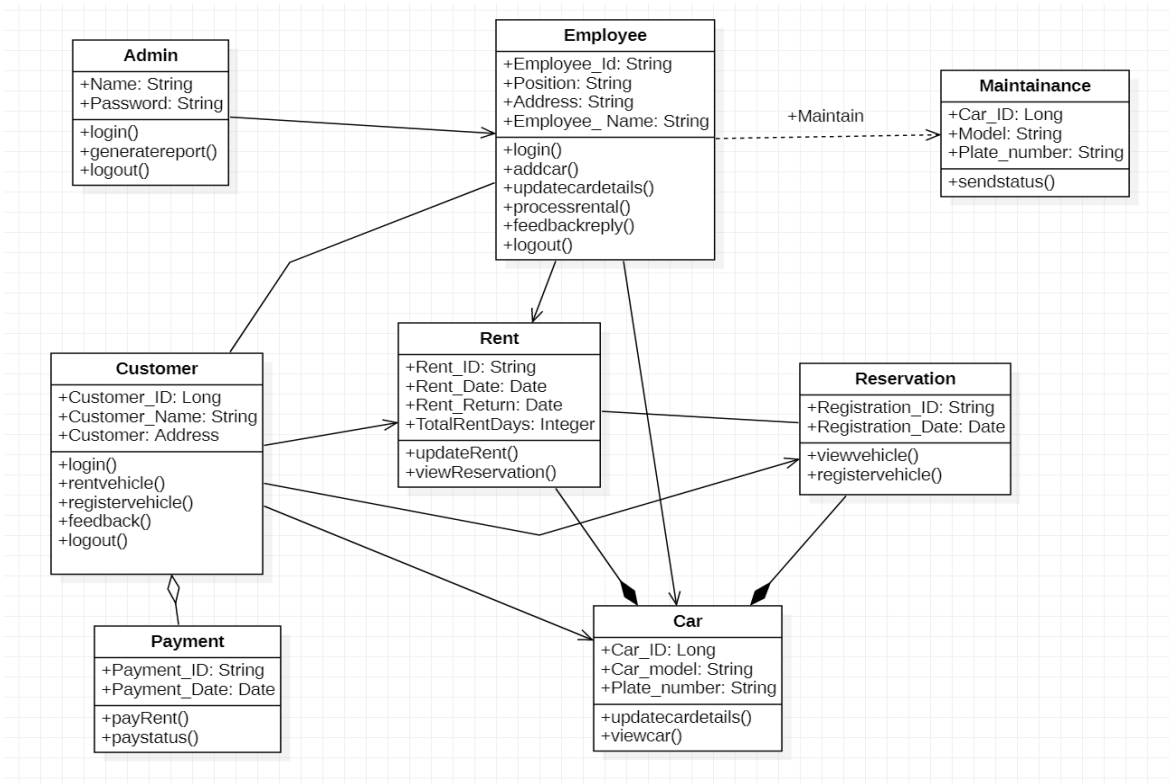


3. DESIGN DIAGRAMS

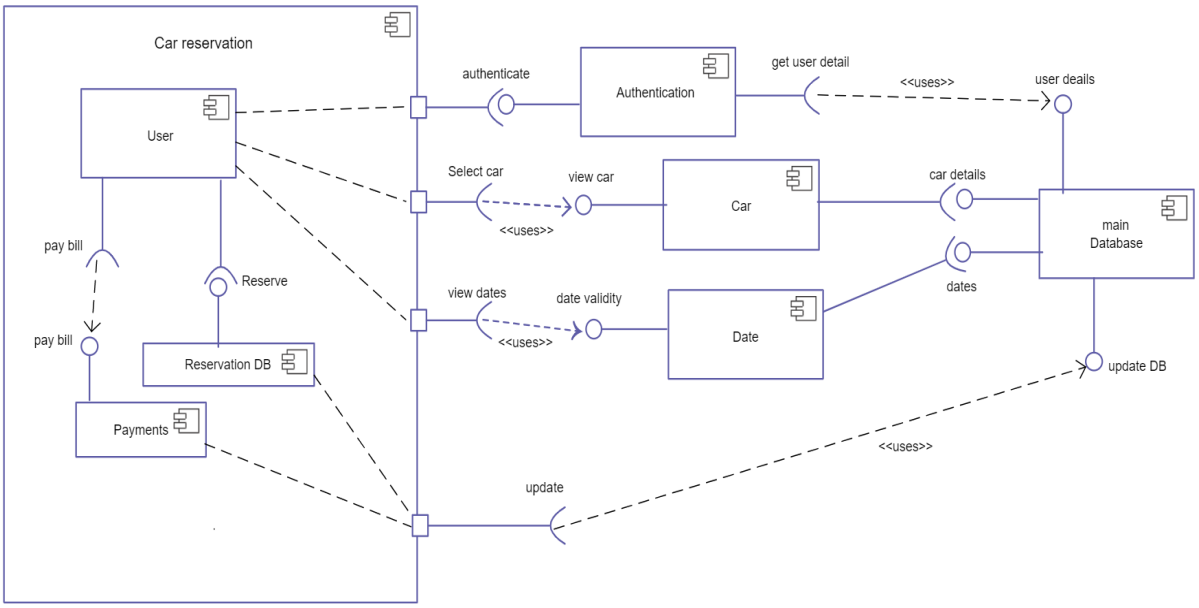
3.1 System Architecture(generic)



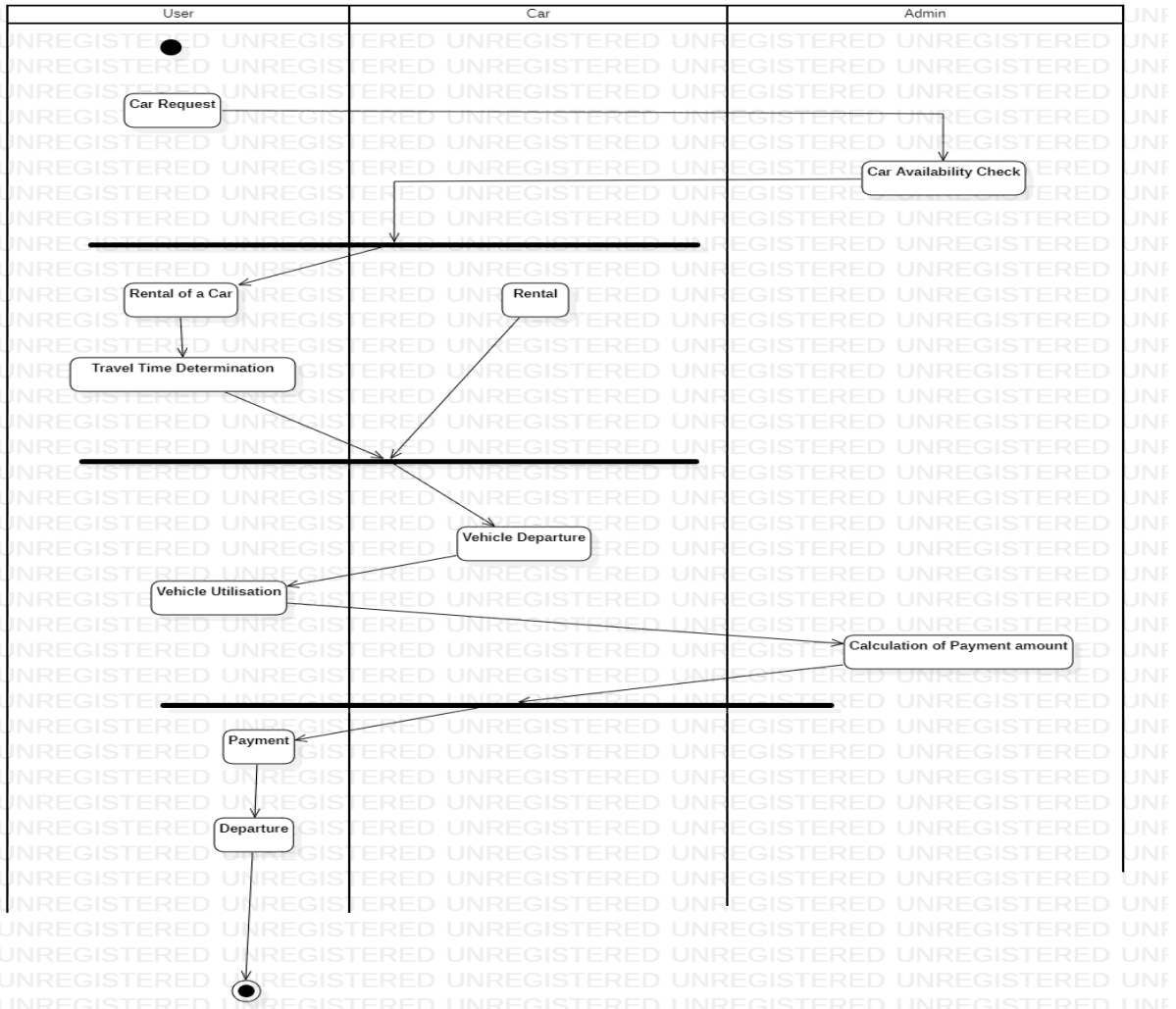
3.2 Class Diagram



3.3 Component diagram

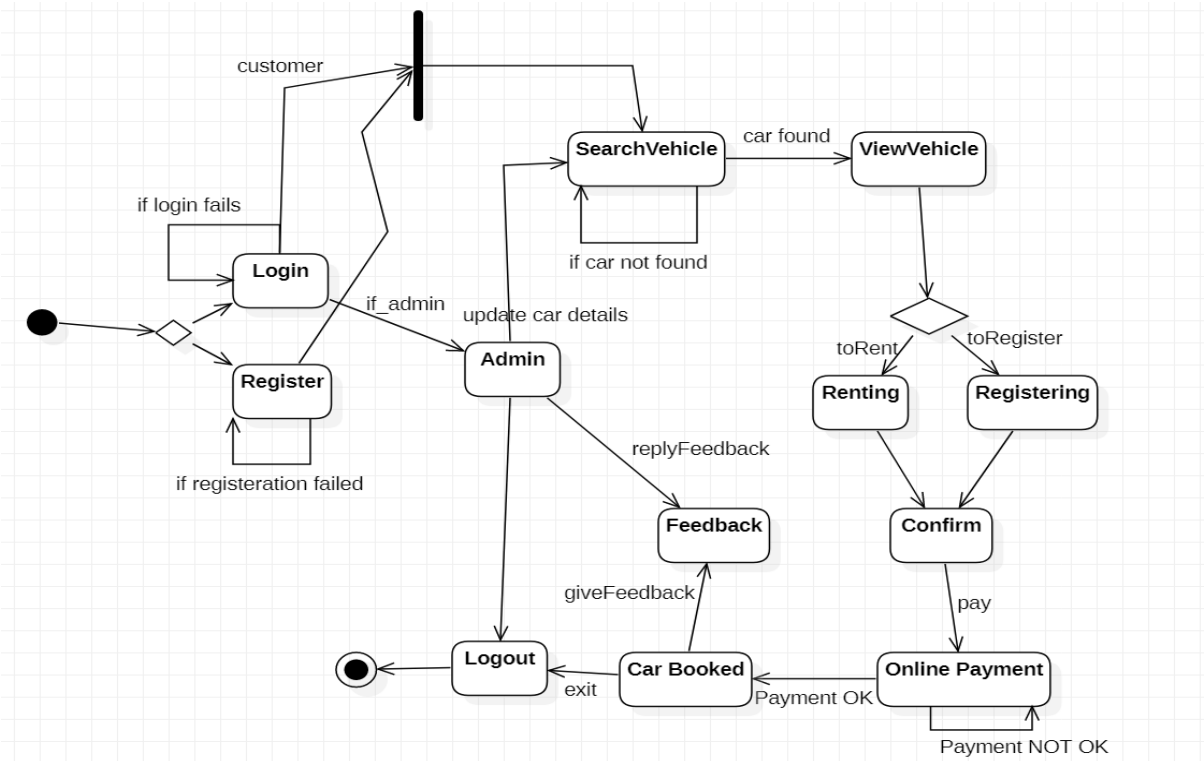


3.4 Activity diagram

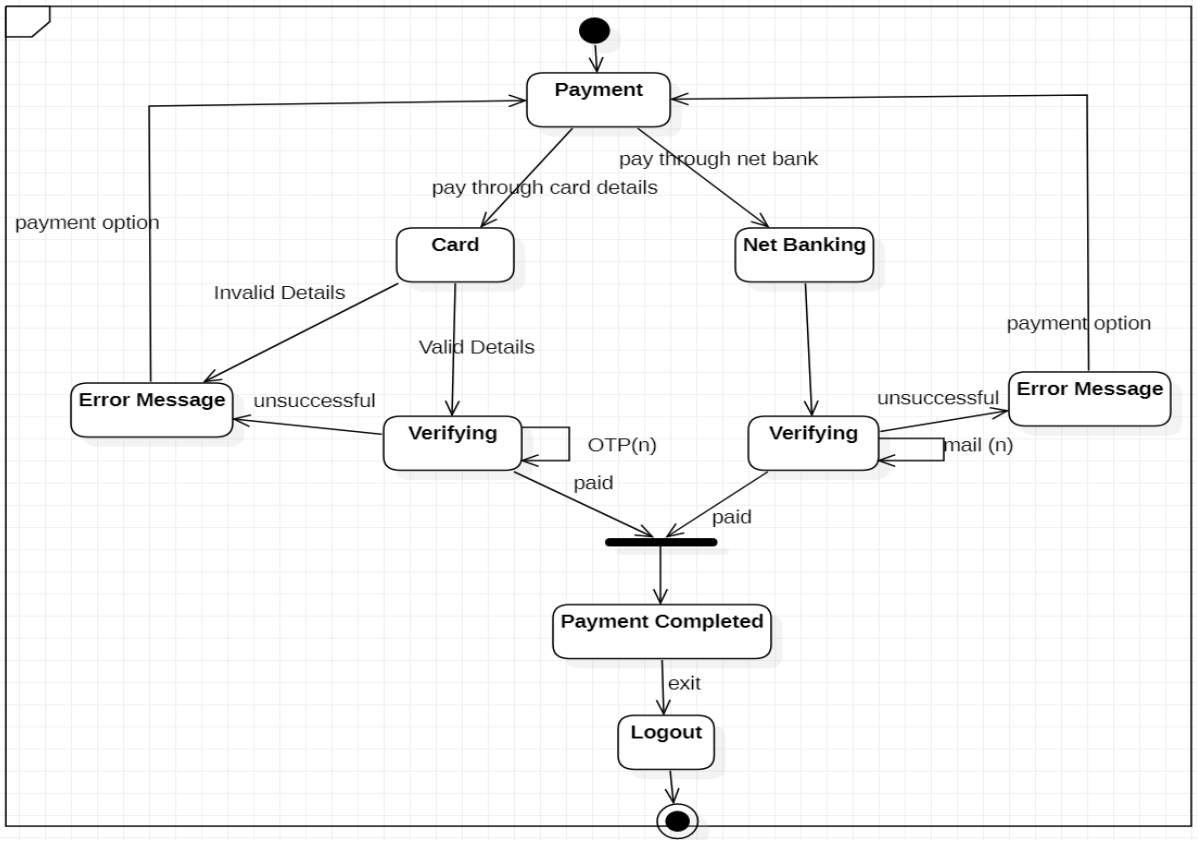


3.5 State diagram

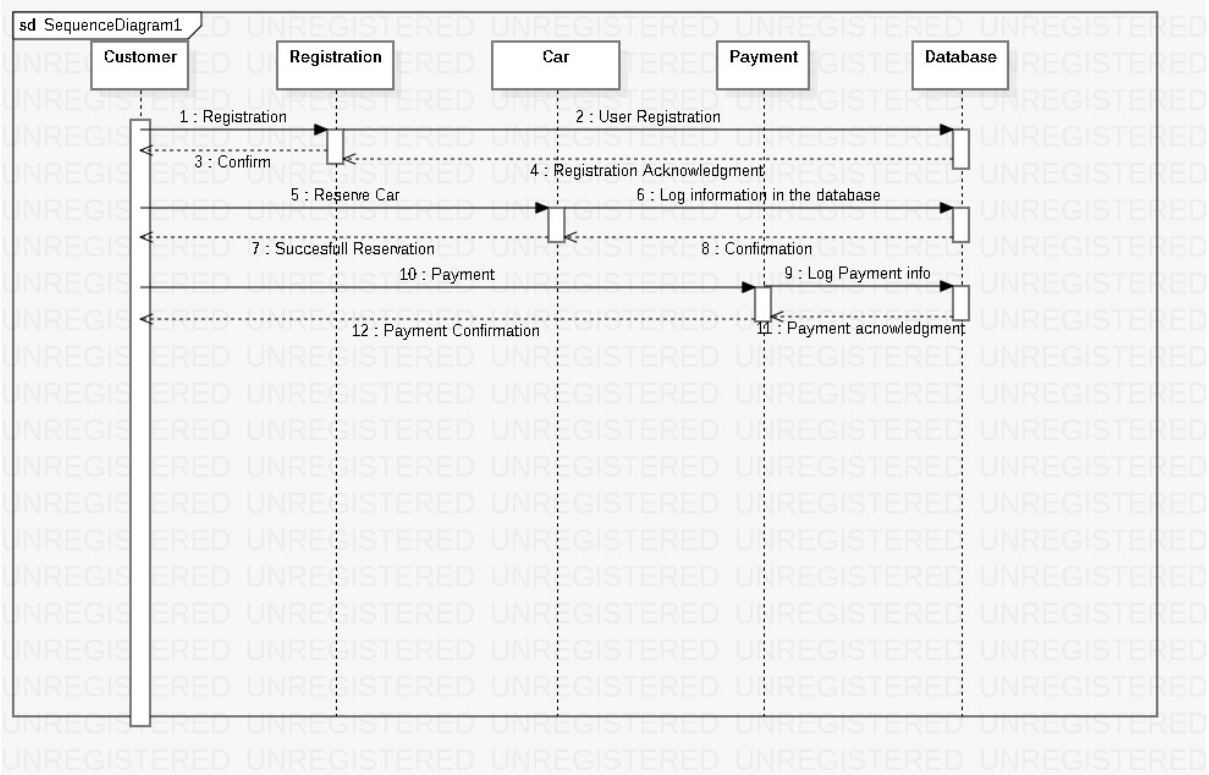
3.5.1 Overall



3.5.2 Payment



3.6 Sequence diagram



4. MODULE DESCRIPTION

USER REGISTRATION AND LOGIN MODULE

The user after entering the system can register if he/she is a new user. The user must provide a username and password in order to register. Once the registration is complete, the user can now proceed to the login page through the user option. In the login form, the user must enter a valid username and password in order to login to the system. The credentials entered must be the same as provided at the time of registration, else the user will not be allowed to enter the system for vehicle selection.

ADMIN AND TRACK BOOKING DETAILS MODULE

The admin's details have been stored in the "ope" table. The admin can login using the details present in this table. The functionalities that the admin can perform is to add a car or delete a car from the vehicles table in the e_reservations database. The admin can enter the details of the cars and add the car to the vehicles table. The admin can delete a car by entering a vehicle number of the car which needs to be deleted. The track booking details functionality helps the user to view the details of the car that has been booked by entering the vehicle number of the car. Once the user enters the vehicle number, the details of the vehicle booked will be shown.

CAR SELECTION AND PAYMENT MODULE

The details of the cars available will be provided to the user. Using the Vehicle number of the available car, the user can enter the details in the reservation form by providing details such as number of seats, journey date etc. Based on the number of seats, the function will calculate the cost required. The user can then proceed to the payment functionality. Here, the user must enter the valid name and select the mode of payment. If the mode of payment is Credit/debit Card then the user must enter valid credentials for the card number, journey date and vehicle number for the payment to be successful. If the selected mode is cash, then the user need not enter any details as it would not be electronic. After successful payment, the user can then exit the system.

5. TEST CASES

User Registration and Login Functionality (Implemented by Saiprakash L Shetty)

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
UL_01	User Registration and Login Functionality	To test whether the registration operation is successful.	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL.	Go to the New User page from the Homepage and enter the username and password.	Username -pesu123 Password - pesu@123	A dialog box with a message, “Thank you for registering” should pop up.	Performs as expected, to confirm successful registration a dialog box appears with a message, “Thank you for registering”.	PASS
UL_02	User Registration and Login Functionality	To test whether the registration operation is successful	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	Go to the New User page from the Homepage and enter the username and password	Username - saiprakashlaxmanshetty Password - pesu@123	A dialog box with a message, “Data Truncation: Data too long for column ‘username’ at row 1” should pop up	When the username is more than 20 characters the message “Data Truncation: Data too long for column ‘username’ at row 1” pops up, i.e. performs as expected	PASS
UL_03	User Registration and Login Functionality	To test whether the registration operation is successful	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	Go to the New User page from the Homepage and enter the username and password	Username - pesu123 Password - saiprakashlaxmanshetty	A dialog box with a message, “Data Truncation: Data too long for column ‘password’ at row 1” should pop up.	When the password is more than 20 characters the message “Data Truncation: Data too long for column ‘username’ at row 1”	PASS

							pops up, i.e. performs as expected	
UL_04	User Registration and Login Functionality	To test whether a registered user can successfully login with the credentials.	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	Enter user name and password to login to the system	Username -pesu123 Password - pesu@123	User should be able to successfull-y login to the system	Performs as expected and Login is successful thereby redirecting the users to the booking page.	PASS
UL_05	User Registration and Login Functionality	To test whether an unregistered user can login with the wrong credentials	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	Enter incorrect username and correct password in the login page.	Username -pesu456 Password - pesu@123	Users should not be able to login to the system and a dialog box with a message, “Invalid Username” should pop up.	performs as expected	PASS
UL_06	User Registration and Login Functionality	To test whether registered user can login or not with the wrong credentials	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	Enter correct username and incorrect password in the login page.	Username -pesu123 Password - pesse21	Users should not be able to login to the system and a dialog box with a message, “Invalid Password” should pop up	Performs as Expected	PASS
UL_07	User Registration and Login Functionality	To test functionality of “Go Back” Button	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	After Successful /unsuccessful registration, click on the “Go Back” Button.	Click on the “Go Back” Button after registration.	On clicking the “Go Back” Button after successful/unsuccessful registration the system must redirect the user to the Home Page	Performs as expected and redirects to the Home Page	PASS

UL_08	User Registration and Login Functionality	To test functionality of "OK" Button in the login page.	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	After successful Login, click on the "OK" Button.	Click on the "OK" Button after logging in.	On entering valid credentials and clicking on the "OK" the system must redirect the user to the Vehicle Selection Page	Performs as expected and redirects to the Vehicle Selection Page	PASS

Admin and Item List Functionality (Implemented by Akash Yadav)

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
AL_01	Admin and Item List Functionality	To test the Login functionality of the admin page with the credentials of the admin.	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL.	Enter the username and password of the admin to login to the admin account.	The username and password of the admin will be stored in the "ope table" in the "e_reservations" database. The credentials of the admin are: Username: naveen Password: naveen	On entering the valid credentials of the admin, the system must login successfully to the admin account	On entering the valid credentials, the system successfully logs in to the system, i.e. performs as expected	PASS
AL_02	Admin and Item List Functionality	To test the Login functionality of the admin page with the credentials of the admin	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	Enter the username and password of the admin to login to the admin account.	If there is no entry in the "ope table", which must otherwise contain the username and password of the admin.	The Admin Login page must be non editable and the textfield cannot be accessed	The admin page is non editable and the text field for username and password	PASS

							cannot be accessed, i.e. performs as expected	
AL_03	Admin and Item List Functionality	To check whether the login is unsuccessful when the username of the admin is incorrect.	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	Go to the Admin page and Enter the username and password of the admin to login to the admin account.	The username and password of the admin will be stored in the "ope table" in the "e_reservations" database. The credentials of the admin are: Username: naveen123 Password: naveen	The admin login should fail because this particular username is not stored in the "ope table". A message saying "Please enter a valid username or password"	Login fails as expected and the message "Please enter a valid username or password" is displayed.	PASS
AL_04	Admin and Item List Functionality	To check whether the login is unsuccessful when the password of the admin is incorrect.	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	Go to the Admin page and Enter the username and password of the admin to login to the admin account.	The username and password of the admin will be stored in the "ope table" in the "e_reservations" database. The credentials of the admin are: Username: naveen Password: naveen123	The admin login should fail because this particular password is not stored in the "ope table". A message saying "Please enter a valid username or password"	Login fails as expected and the message "Please enter a valid username or password" is displayed.	PASS
AL_05	Admin and Item List Functionality	To check whether admin can add a vehicle into the item	Install an IDE for Java, preferably NetBeans and a	Click on the "Add" Button to enter the vehicle details	The Admin must click on the "Add" button to add vehicles into the system.	Upon Clicking the "Add" Button, the system must	Upon Clicking the "Add" Button, the system redirects	PASS

		list successfully	Database Service such as MySQL			redirect the admin to the add vehicle page.	the admin to the add vehicle page, i.e. it performs as expected.	
AL_06	Admin and Item List Functionality	To check if the entered vehicle name is a valid string or not.	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	After logging in from the admin credentials, go to the ADD option in the admin page.	At this point, enter a valid vehicle name such as: - Tata Safari	Since the name is a valid one(i.e less than 20 characters, the form should successfully be submitted)	As expected, the vehicle name is accepted successfully and updated in the database.	PASS
AL_07	Admin and Item List Functionality	To check if the entered vehicle name is a valid string or not.	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	After logging in from the admin credentials, go to the ADD option in the admin page.	At this point, enter a invalid vehicle name such as: - Tata Maruti swift dzire Safari	Since the name is not a valid one(i.e more than 20 characters, the form should pop an error message)	As expected, the vehicle name is not accepted and an error message pops up.	PASS
AL_08	Admin and Item List Functionality	To check if the entered vehicle number is a valid alphanumeric string or not.	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	After logging in from the admin credentials, go to the ADD option in the admin page.	At this point, enter a valid vehicle number such as: - KA09MJ5634	Since the number is a valid one(i.e less than 10 characters, the form should successfully be submitted)	As expected, the vehicle number is accepted successfully and updated in the database.	PASS
AL_09	Admin and Item List Functionality	To check if the entered vehicle number is a valid alphanumeric string or not.	Install an IDE for Java, preferably NetBeans and a Database Service	After logging in from the admin credentials, go to the ADD option in	At this point, enter an invalid vehicle number such as:	Since the number is an invalid one(i.e greater than 10 characters, the form should pop	As expected, the vehicle number is not accepted and an error	PASS

			such as MySQL	the admin page.	- KA09MJ563 4456	an error message)	message pops up.	
AL_10	Admin and Item List Functionality	Enter From-To details create travel details	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL.	complete the login into the system and fill the From-To in ADD form which is filled by admin	Enter 'From' place and 'To' destination. The length of both the strings should be less than 20 characters. From : Mumbai To: Delhi	On entering valid From and To details, the destination details must get updated	On entering valid From and To details, the destination details get updated .i.e., it performs as expected.	PASS
AL_11	Admin and Item List Functionality	Enter From-To details while booking	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	complete the login into the system and fill the From-To in ADD form which is filled by admin	Enter 'From' place and 'To' destination. The length of both the strings should be less than 20 characters. From : Mamungkukumpurangkuntjunya To: Venkatanarasimharajuvaripeta	Since the String length is an invalid one(i.e greater than 20 characters, the form should pop an error message)	As expected, the From-To details is not accepted and an error message pops up	PASS
AL_12	Admin and Item List Functionality	Enter Cost details while entering travel details	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQ	complete the login into the system and fill the From-To in ADD form which is filled by admin	Enter Cost details. The cost should be less than or equal to 10000 Cost : 8000	Since the cost is valid the cost details for that particular car travel details must get updated	On entering valid cost details, the cost details get updated successfully	PASS
AL_13	Admin and Item List	To test whether the Arrival Time	Install an IDE for Java,	After logging into the	The Arrival Time and Departure	The system must accept the arrival	The system accepts	PASS

	Functionality	and Departure Time entered is valid or not	preferably NetBeans and a Database Service such as MySQL	admin page, enter the valid time for arrival time and departure time.	Time inputs must be in the form: HH:MM:SS, i.e. hours, minutes and seconds. Arrival Time: 08:09:12 Departure Time: 03:05:30	time and departure time input from the admin as it is in the right format	the arrival time and departure time input from the admin as it is in the right format, i.e. it performs as expected.	
AL_14	Admin and Item List Functionality	To test whether the Arrival Time and Departure Time entered is valid or not	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	After logging into the admin page, enter the arrival time and departure time.	The Arrival Time and Departure Time inputs must be in the form: HH:MM:SS, i.e. hours, minutes and seconds. But, the inputs given are Arrival Time: 180:109:25 Departure Time: 888:895:30	The system must not accept the arrival time and departure time input from the admin as it is not in the right format	The system does not accept the arrival time and departure time input from the admin as it is not in the right format, i.e. it performs as expected.	PASS
AL_15	Admin and Item List Functionality	To test whether the Driver Name entered is valid or not	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	After logging into the admin page, enter the valid Driver Name	The Driver name field input is “Dhanik Shetty”.	The entered value for driver name must be accepted as it is less than 20 characters	The entered value for driver name is accepted as it is less than 20 characters, i.e. it performs as expected	PASS
AL_16	Admin and Item List Functionality	To test whether the Driver Name entered is valid or not	Install an IDE for Java, preferably NetBeans and a	After logging into the admin page, enter the	The Driver name field input is “Novak Rafael Nadal Djokovic”.	The entered value for driver name must not be accepted as	The entered value for driver name is not	PASS

			Database Service such as MySQL	valid Driver Name		it is more than 20 characters	accepted as it is more than 20 characters, i.e. it performs as expected	
AL_17	Admin and Item List Functionality	To test whether the admin can delete the vehicle from the system or not	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	Select the Delete option from the select form page to delete a vehicle.	Enter the vehicle number of the vehicle to be deleted.	If the vehicle number entered exists then the delete operation must take place or else deletion of vehicle must not be executed	The vehicle will be deleted successfully from the system if the vehicle entered is valid.	PASS

Ticket Booking and Payment Functionality (Implemented by Naveen Suresh)

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
TP_1	Ticket Booking and Payment Functionality	To check whether the user can book the ticket	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	After logging into your user account successfully, click on the Book Button	User must click on the Book Button upon successful login	The system must redirect the user to the reservation form to book his ticket	The system redirects the user to the reservation form to book his ticket, i.e. performs as expected	PASS
TP_2	Ticket Booking and Payment Functionality	To check if the user enters a valid journey date	Install an IDE for Java, preferably NetBeans and a Database Service	Enter a valid user date in the journey date text field	User must enter the journey date in the format(yyyy-mm-dd):	The entered journey must be valid as it follows the format	The entered journey date is valid and the system proceeds	PASS

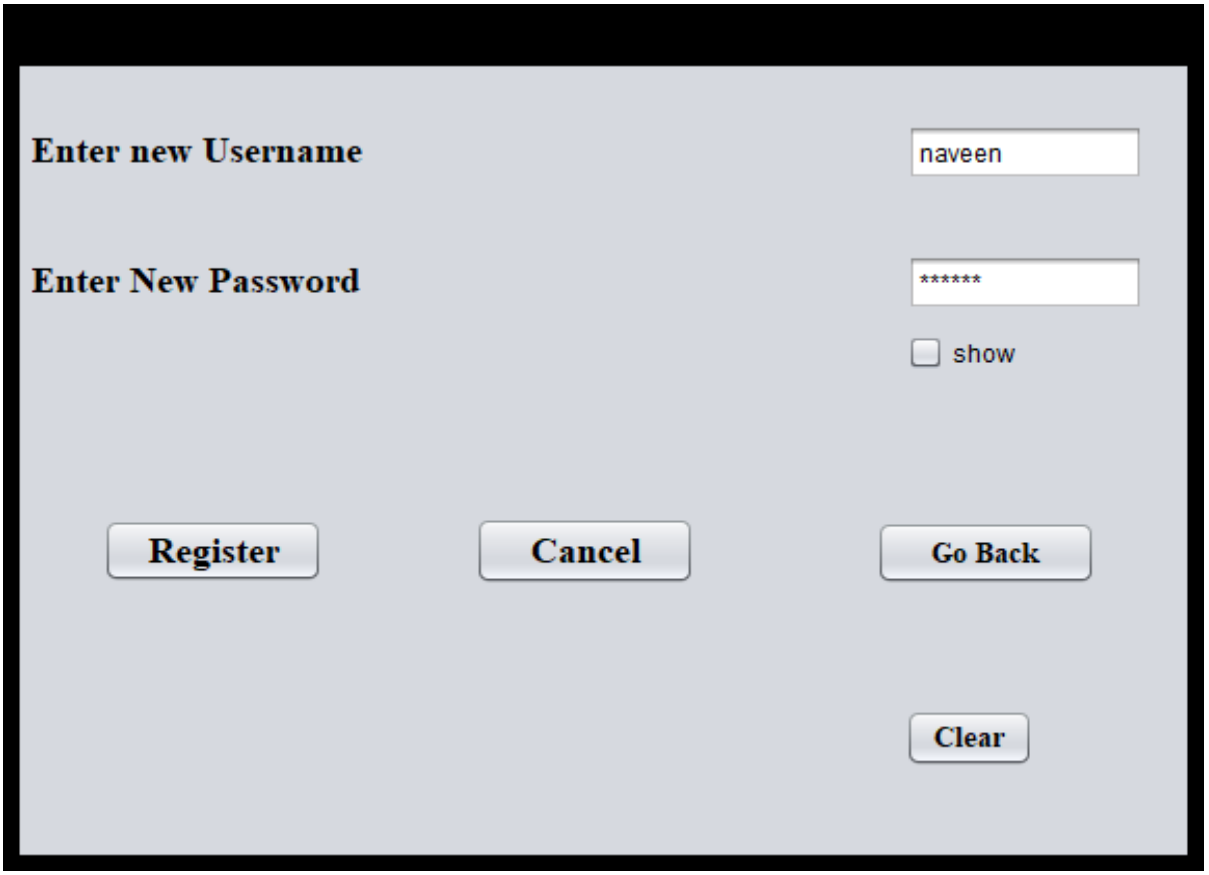
			such as MySQL		2021-05-15		with the booking process as expected	
TP_3	Ticket Booking and Payment Functionality	To check if the vehicle number entered is valid and present in the database.	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	After logging into your user account successfully, click on the Book Button	The user enters the vehicle number that they desire to book, such as; KA09MJ5634	Since the entered number is valid(i.e <=10 characters)and it's match is found in the database, it should be accepted	The match is found and therefore this test case is accepted and the user moves on.	PASS
TP_4	Ticket Booking and Payment Functionality	To check if the vehicle number entered is invalid or not present in the database so that the booking becomes invalid.	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	After logging into your user account successfully, click on the Book Button	An invalid vehicle number is entered such as; KA09MJ563489	Since the entered number is invalid(i.e <=10 characters)and it's match is not found in the database, it's not accepted.	As expected, the vehicle number is not accepted.	PASS
TP_5	Ticket Booking and Payment Functionality	To check if the number of seats entered is valid.	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	After logging into your user account successfully, click on the Book Button	The user enters the number of seats they want - 2,3,4 or 6	Since the entered number is valid(i.e <=6) , it should be accepted.	The seats are accepted and the booking phase carries on.	PASS
TP_6	Ticket Booking and Payment Functionality	To check if the number of seats entered is valid.	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	After logging into your user account successfully, click on the Book Button	An invalid number of seats are entered - Such as 8	Since the entered number is invalid(i.e >=6) , it should not be accepted.	The test case performs as expected and booking is not accepted.	PASS

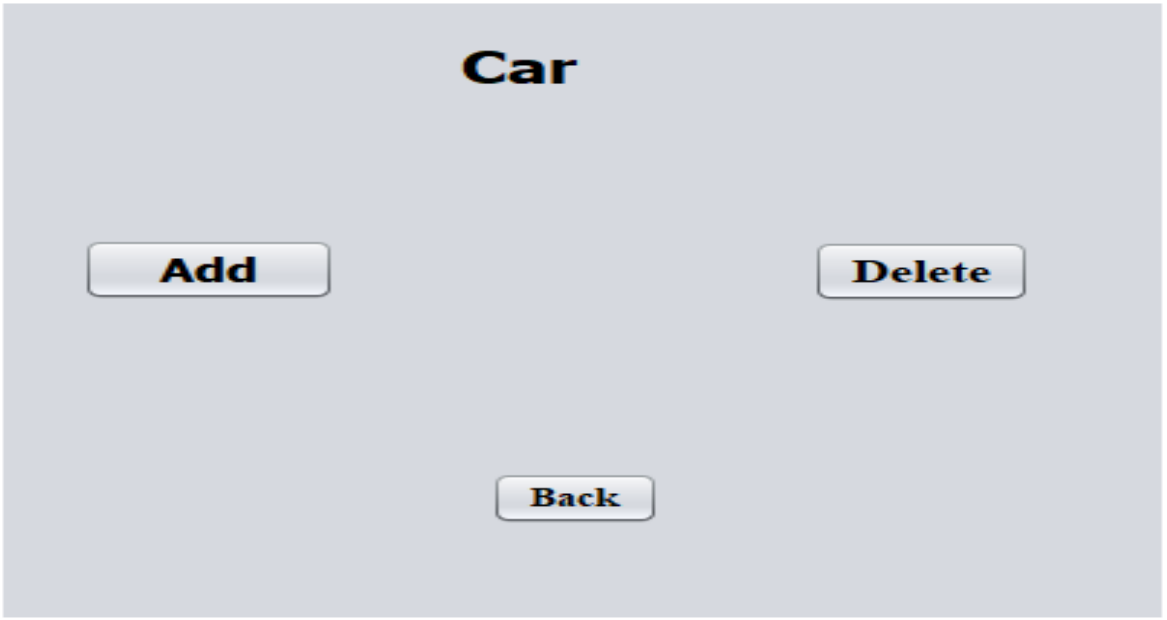
TP_7	Ticket Booking and Payment Functionality	To check if the system computes the cost appropriately	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	Click on the “cost” button to compute the cost for your journey	The system must be able to generate the cost based on the input given in the “number of seats required” test field and the facility level.	The cost must be generated based on the input given in the number of seats field and facility level field.	The cost is generated based on the input given in the number of seats field and facility level field, i.e. it performs as expected	PASS
TP_8	Ticket Booking and Payment Functionality	To test whether the user can perform the payment operation	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	Enter your name in the payment form	Enter the same name as given during reservation	The system must allow the user to proceed for payment if valid username is provided	The system allows the user to proceed for payment upon valid username	PASS
TP_9	Ticket Booking and Payment Functionalityand Payment Functionality	To test whether the user can perform the payment operationperform the payment operation	To test whether the user can perform the payment operationperform the payment operation	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL.	Enter a different name to what was given during reservation	The system must not allow the user to proceed for payment if invalid username is provided	The system does not allow the user to proceed for payment with invalid username, performs as expected	PASS

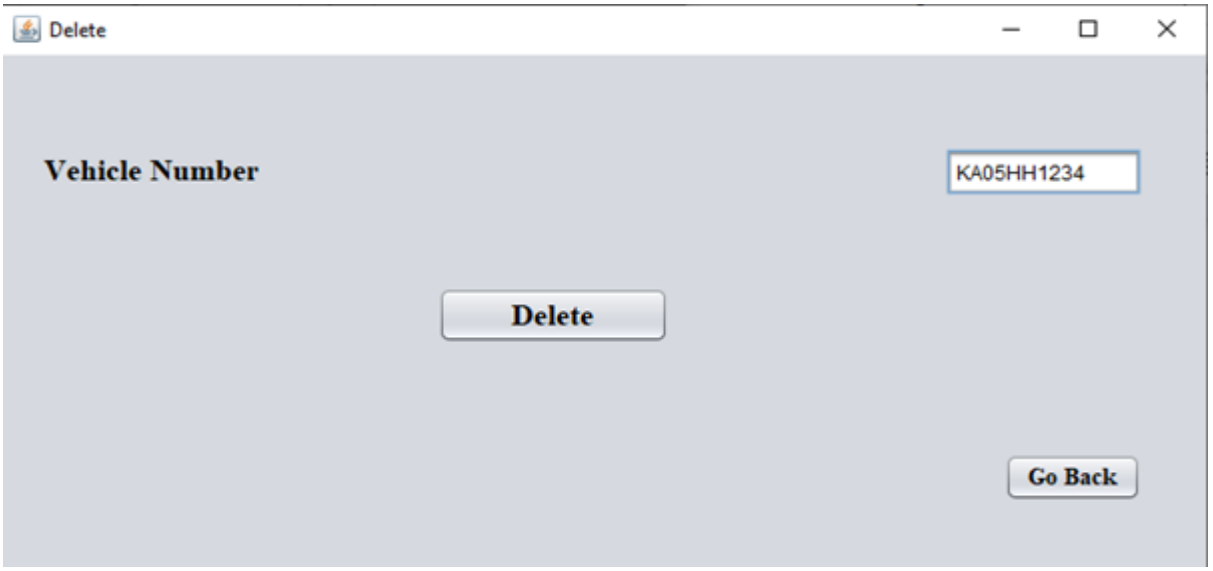
TP_10	Ticket Booking and Payment Functionality	To test whether the user can perform the payment operation using credit or debit card or cash/cheque	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	Enter the journey date and vehicle number in the text fields provided for the same	Enter the journey date in the format(yyyy-mm-dd) and a valid vehicle number in the respective fields	The system must proceed with they payment if the journey date and vehicle number is valid	The system proceeds with the payment, i.e. it performs as expected.	PASS
TP_11	Ticket Booking and Payment Functionality	To test whether the user can perform the payment operation using credit or debit card or cash/cheque	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	Enter the journey date and vehicle number in the text fields provided for the same	Enter the invalid journey date and vehicle number in the respective fields	The system must proceed with they payment if the journey date and vehicle number is invalid	The system must not proceed with the payment, i.e. it performs as expected.	PASS
TP_12	Ticket Booking and Payment Functionality	Vehicle Checking Dialog Window. To track vehicle details.	Install an IDE for Java, preferably NetBeans and a Database Service such as MySQL	In home page you can check about vehicle details by clicking Track Booking Details	Entering valid vehicle number i.e., given vehicle number was used booking before Vehicle Number : KA01EE1234	Since the vehicle number is valid and Show button is clicked Details regarding vehicles are displayed(Name,Vehicle no., Driver Name, From To, Departure, Arrival, Cost, Vehicle Name)	Details regarding vehicle booking are displayed (Name,Vehicle no., Driver Name, From To, Departure, Arrival, Cost, Vehicle Name) i.e., performed as expected.	PASS
TP_13	Ticket Booking and Payment	Vehicle Checking Dialog Window. To	Install an IDE for Java, preferably NetBeans	In home page you can check about	Entering invalid vehicle number i.e., given	Since the vehicle number entered is not present	Pop message saying ‘ Enter valid	PASS

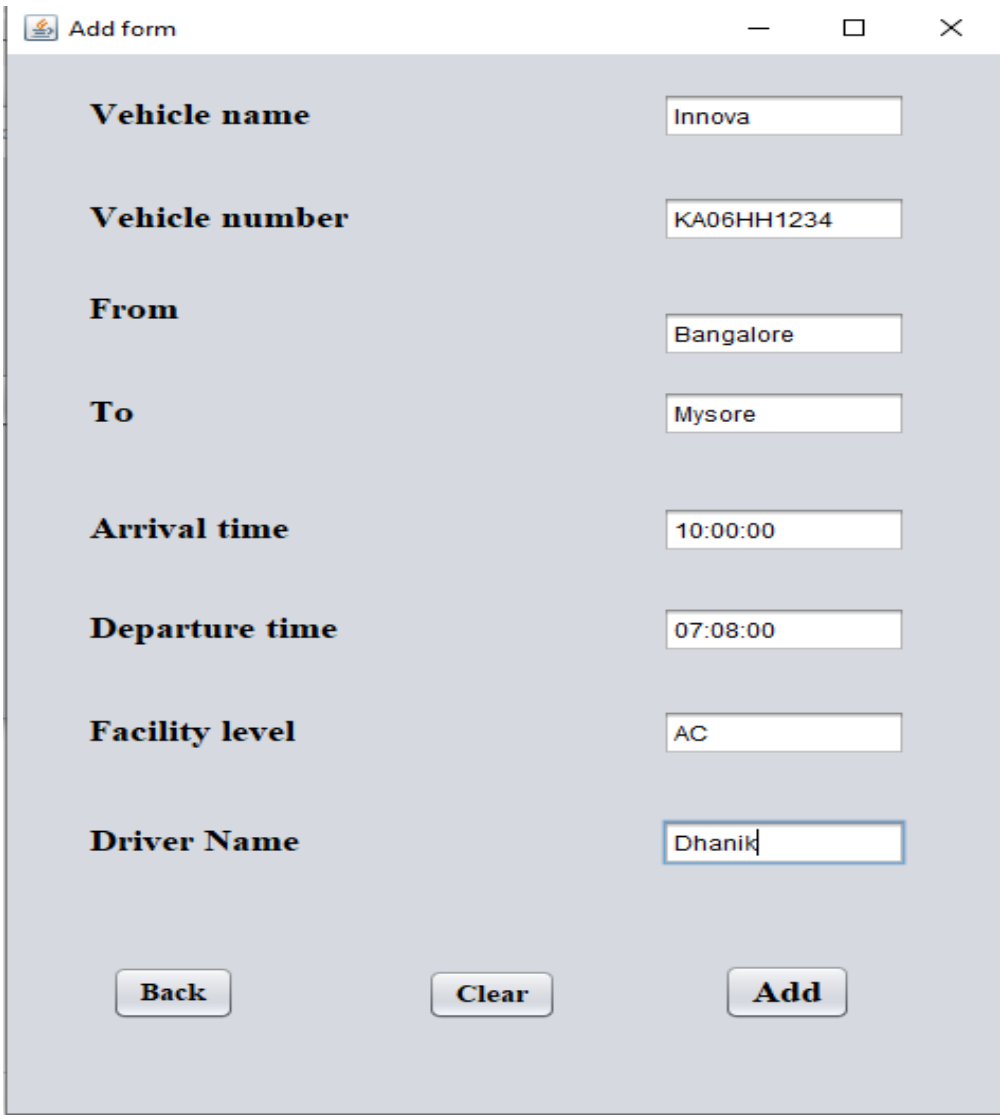
	Functionality	track vehicle details.	and a Database Service such as MySQL	vehicle details by clicking Track Booking Details	vehicle number is not present in database, not been used before Vehicle Number : MP12QW 1110	in database, when you click on 'Show' Button, a pop message saying 'Enter valid Vehicle number' shall be displayed	Vehicle Number' is displayed i.e., performs as expected.	

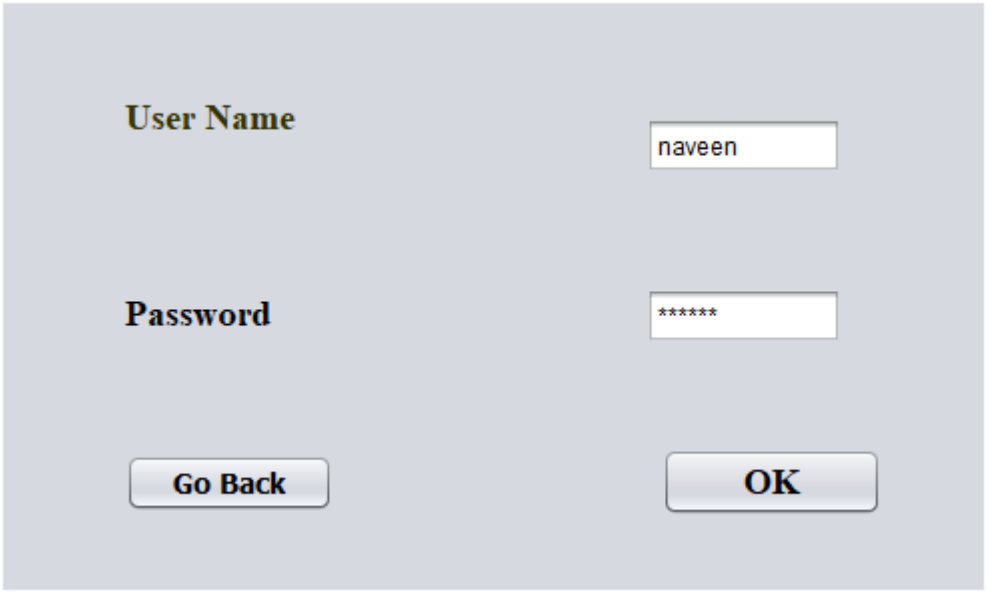
6. SCREENSHOTS OF OUTPUT

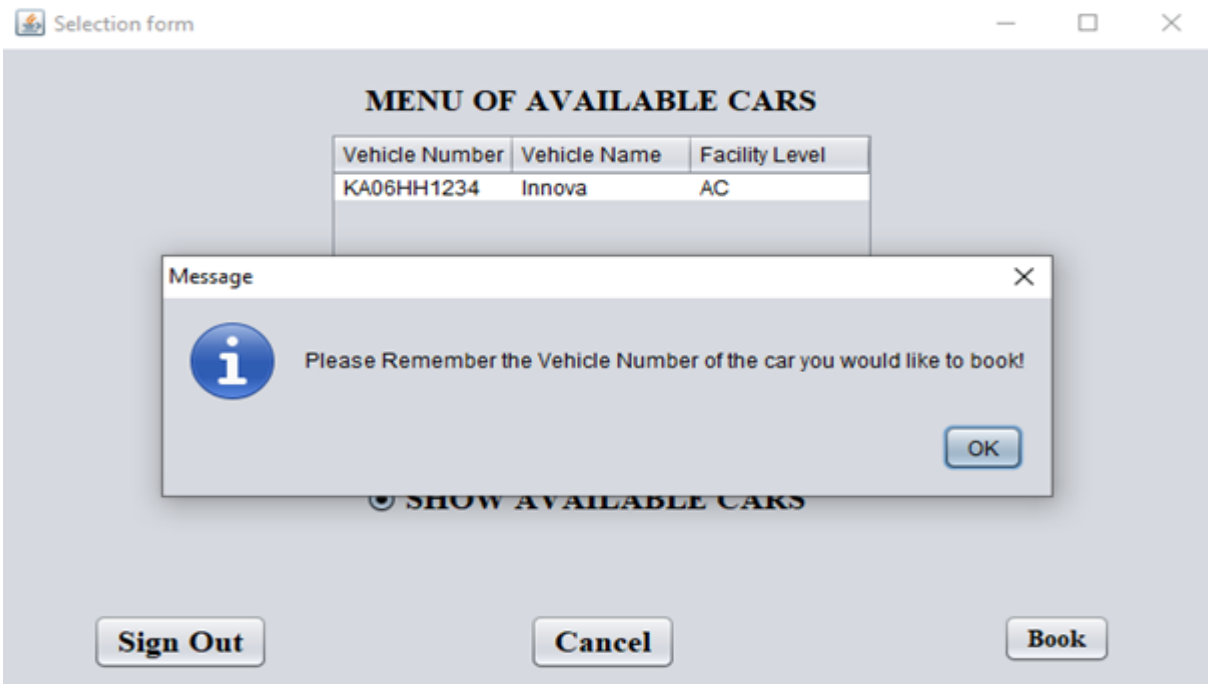



Car
Add
Delete
Back

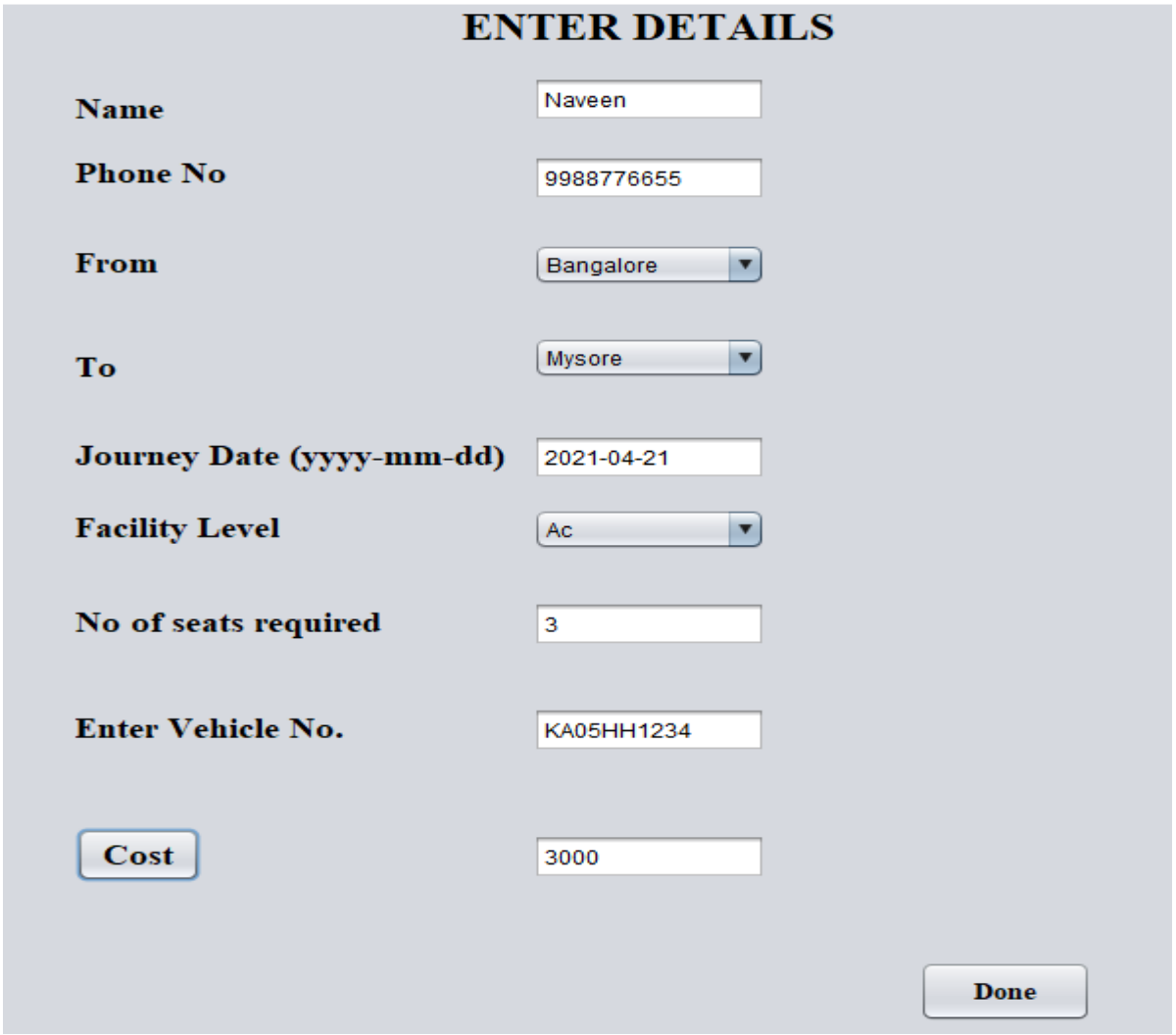

Delete
Vehicle Number
KA05HH1234
Delete
Go Back


Add form
Vehicle name
Innova
Vehicle number
KA06HH1234
From
Bangalore
To
Mysore
Arrival time
10:00:00
Departure time
07:08:00
Facility level
AC
Driver Name
Dhanik
Back
Clear
Add

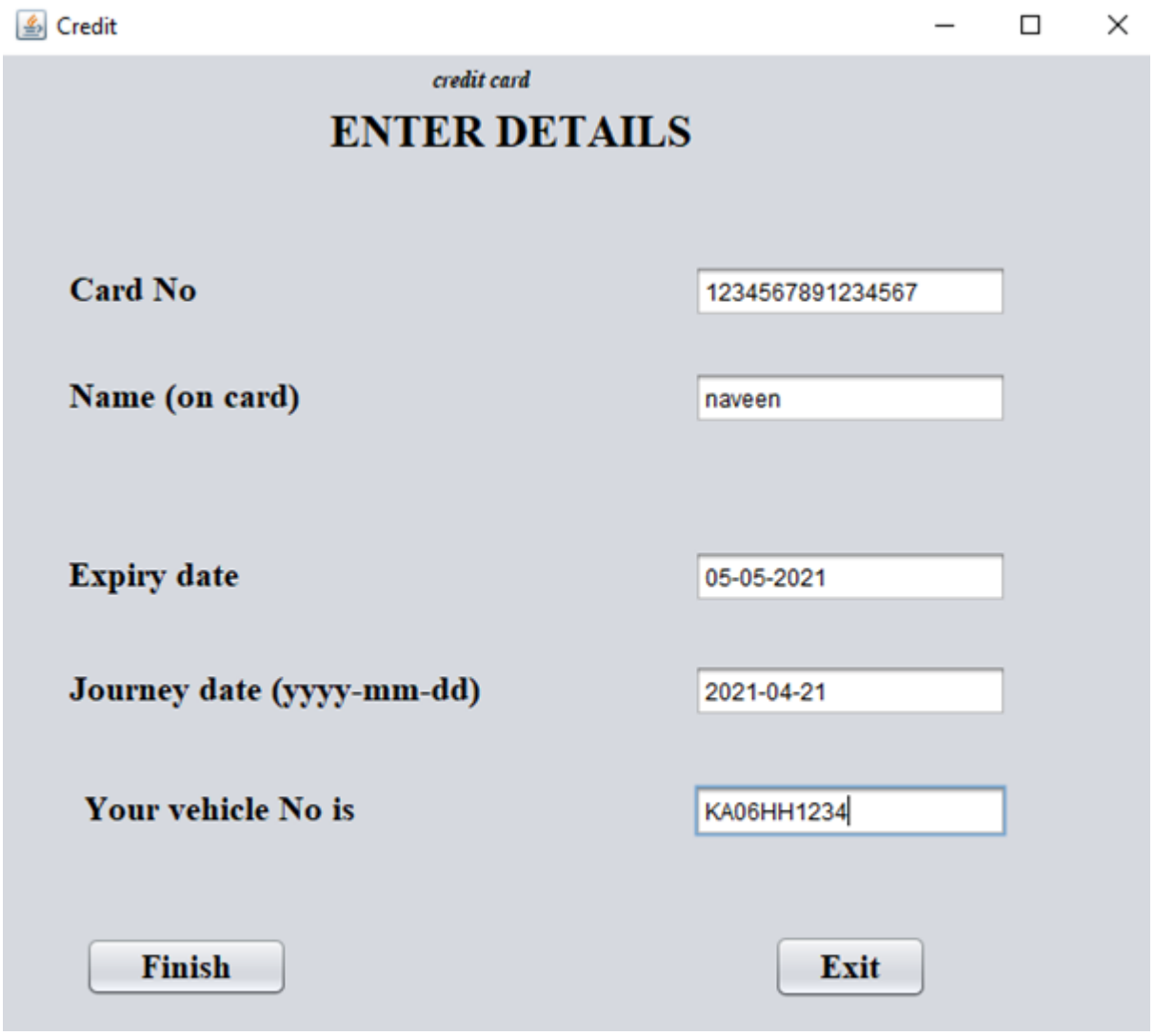

User Name
naveen
Password
******
Go Back
OK



Reservation:




Payment
Enter Name to Confirm:
******
Payment Through
Credit Card
Debit Card
Cash/dd/Cheque
Submit


Credit
credit card
ENTER DETAILS
Card No
1234567891234567
Name (on card)
naveen
Expiry date
05-05-2021
Journey date (yyyy-mm-dd)
2021-04-21
Your vehicle No is
KA06HH1234
Finish
Exit

TRACK BOOKING DETAILS: