Software Design Specifications

GoDriver - Car Rental & Driver Booking App

Version: 00.01

Project Code	*CS4091 / SE4091 (PROJECT-1)*
Supervisor	*Miss Nida Munawwar*
Co-Supervisor	*Sir Farrukh Hassan Syed*
Project Team	★ *Shaheer Mehmood (18K-0128)* ★ *Ali Ahmed (19K-1423)* ★ *Shayan Ahmed Khan (19K-1097)*
Submission Date	*April 21st, 2023*

[Instructions]

- *No section of the template should be deleted. You can write 'Not applicable' if a section is not applicable to your project. But all sections must exist in the final document.*
- *All comments/examples mentioned in square brackets ([]) are in the template for explanation purposes and must be replaced/removed in the final document.*
- *This Instruction' section should also be removed in the final document.*
- *MS Word Reviewing feature must be used to get the document reviewed by PMs or supervisors.*

[

Document Information

Category	Information
Customer	FAST-NU
Project	*GoDriver - Car Rental & Driver Booking App*
Document	Software Design Specification
Document Version	1.0
Status	Draft
Author(s)	1. *Shaheer Mehmood (18K-0128)* 2. *Ali Ahmed (19K-1423)* 3. *Shayan Ahmed Khan (19K-1097)*
Approver(s)	
Issue Date	
Document Location	
Distribution	Advisor Project Coordinator's Office (through Advisor)

Table of Contents

1 Introduction **8**

1.1 Purpose of Document *8*

1.2 Intended Audience *8*

1.3 Document Convention *8*

1.4 Project Overview *8*

1.5 Scope *8*

2 Design Considerations **9**

2.1 Assumptions and Dependencies *9*

2.2 Risks and Volatile Areas *9*

3 System Architecture **10**

3.1 System Level Architecture *10*

3.2 Software Architecture *10*

4 Design Strategy **11**

5 Detailed System Design **12**

5.1 Database Design *12*

5.1.1 ER Diagram 12

5.1.2 Data Dictionary 12

5.1.2.1 Data 1 12

5.1.2.2 Data 2 12

5.1.2.3 Data n 12

5.2 Application Design *14*

5.2.1 Sequence Diagram 14

5.2.1.1 <Sequence Diagram 1> 14

5.2.1.2 <Sequence Diagram 2> 14

5.2.1.3 <Sequence Diagram n> 14

5.2.2 State Diagram

5.2.3 Data Flow Diagram

5.2.4 Activity Diagram 14

6 References **15**

1 Introduction

1.1 Purpose of Document

The purpose of this document is to provide a detailed design specification for the development of "MyDrive - Car Rental & Driver Booking App". It outlines the system architecture, software architecture, and design considerations required for the development of the application.

1.2 Intended Audience

This document is intended for the development team and stakeholders who will be involved in the development, deployment, and maintenance of the application.

1.3 Document Convention

This document follows the IEEE standard for Software Design Specifications.

1.4 Project Overview

"MyDrive - Car Rental & Driver Booking App" is a mobile application that allows users to rent a car and book a driver on demand. The application provides a seamless user experience by integrating various functionalities such as real-time tracking, secure payments, and user feedback

1.5 Scope

The scope of this project includes the development of a mobile application for Android and iOS platforms. The application will provide a simple and user-friendly interface for users to search, book and rent cars with drivers. The application will also include features for drivers to manage their cars, schedule bookings, and view their earnings

2 Design Considerations

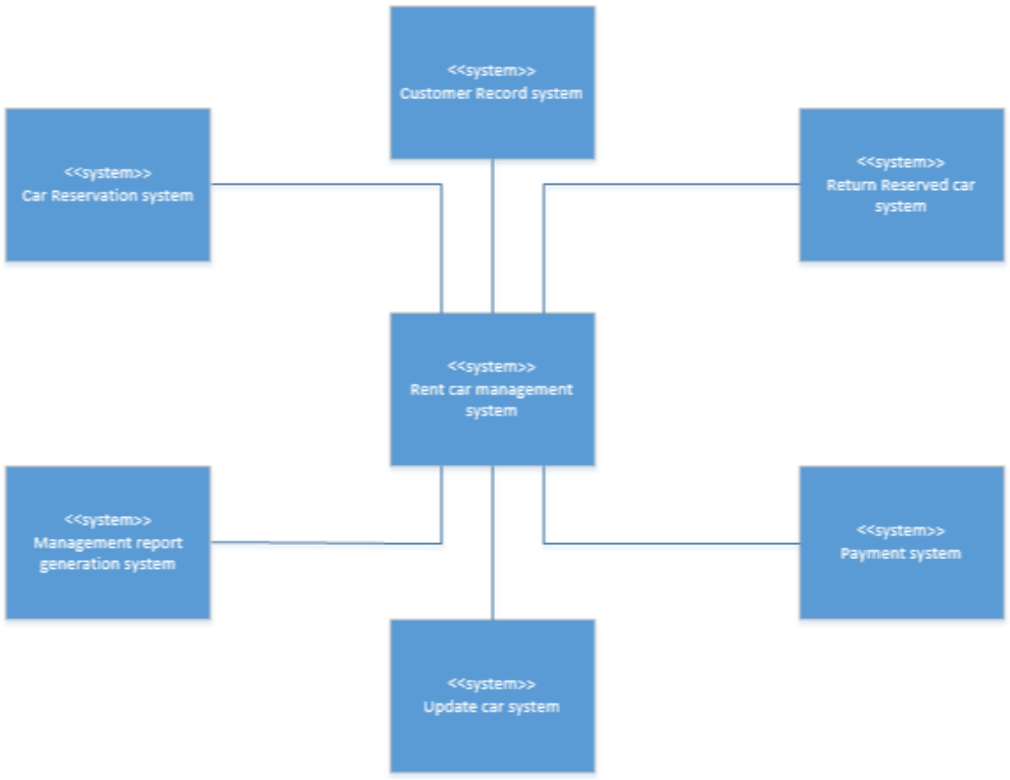
2.1 Assumptions and Dependencies

- *The application will require an internet connection to access the server and perform various operations.*
- *Users must have a valid driving license to rent a car.*
- *Drivers must have a valid driver's license and insurance to provide services.*
- *The application will use third-party APIs for payment processing and location tracking.*
- *The application will be developed using React Native framework.*

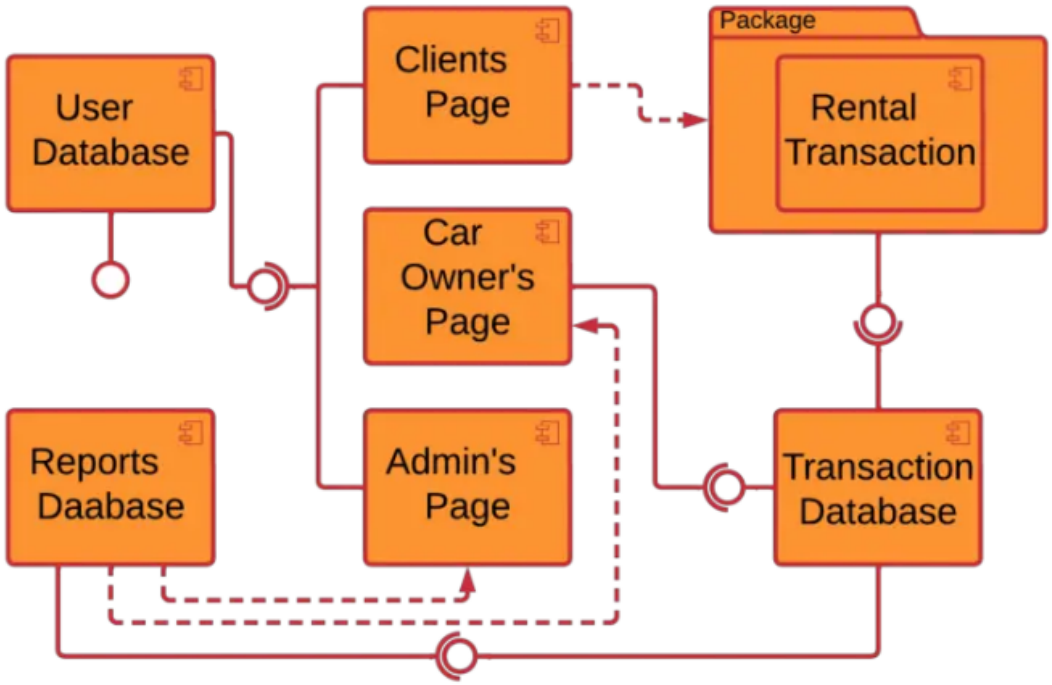
2.2 Risks and Volatile Areas

- *The application may face challenges related to the integration of third-party APIs.*
- *Security risks such as data breaches and unauthorized access to the system.*
- *Technical issues related to server downtime and network connectivity.*
- *The application may face competition from existing car rental and driver booking services.*

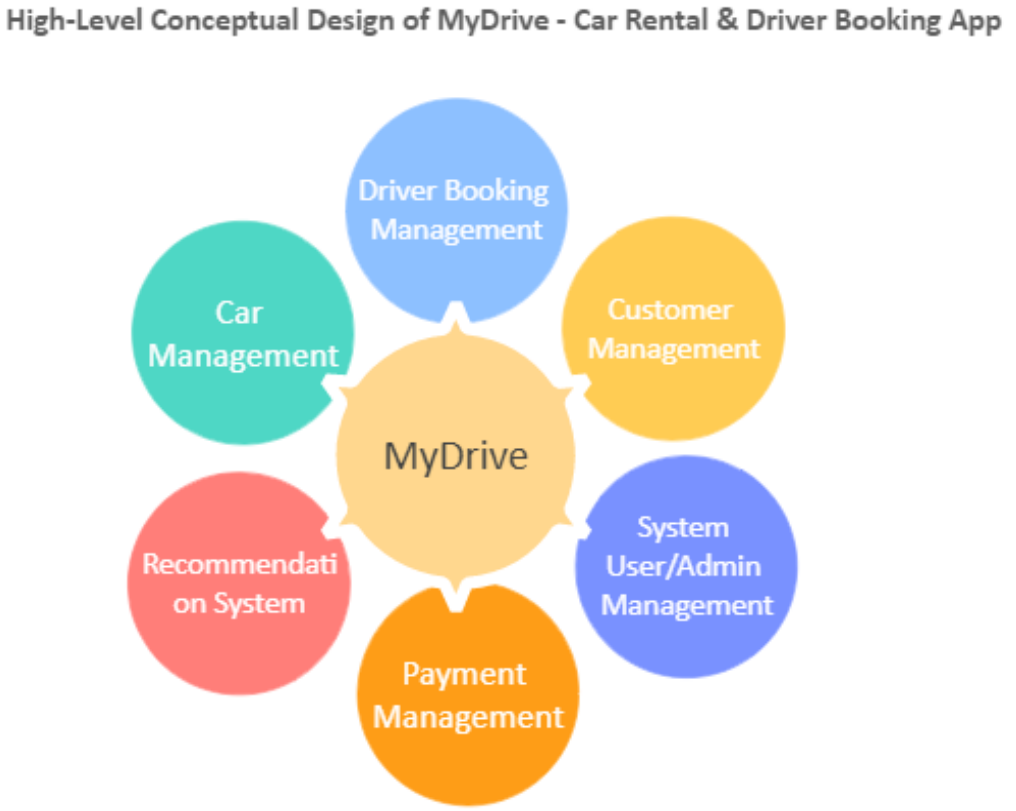
3 System Architecture



Component Diagram



3.1 System Level Architecture



The system will be composed of the following components:

Mobile Application

The mobile application will be developed using React Native framework. It will provide the user interface that allows users to rent a car and book a driver. The application will have the following features:

- *User registration and login*
- *Car rental booking*
- *Driver booking*
- *Real-time tracking*
- *Secure payments*
- *User feedback and ratings*
- *Push notifications*

Server

The server will handle requests from the mobile application and communicate with the third-party APIs. It will be responsible for the following tasks:

- *User and driver management*
- *Car rental and driver booking management*
- *Payment processing*
- *Location tracking*
- *Push notifications*

Database

The database will store user data, driver data, and other relevant information. It will be responsible for the following tasks:

- *User and driver data storage*
- *Car rental and driver booking data storage*
- *Data retrieval and manipulation*

Third-Party APIs

The application will use third-party APIs for payment processing and location tracking. The following APIs will be integrated into the application:

- *Payment Gateway API: The payment gateway API will be used to process secure payments for car rental and driver booking services.*
- *Location Tracking API: The location tracking API will be used to track the location of the rented car and the driver in real-time.*

3.2 Software Architecture

The software architecture of the application will follow the Model-View-Controller (MVC) design pattern. The application will be divided into three main components

Model

The data layer will handle data storage and retrieval from the database. It will be responsible for the following tasks:

- ➢ *User and driver data storage and retrieval*
- ➢ *Car rental and driver booking data storage and retrieval*

View

The user interface layer will display data to the user and handle user input. It will be responsible for the following tasks:

- ➢ *User registration and login*
- ➢ *Car rental booking*
- ➢ *Driver booking*
- ➢ *Recommendation System*
- ➢ *Real-time tracking*
- ➢ *Secure payments*

Controller

The controller layer will handle user input and manipulate data in response to user actions.The controller will interact with the model and the view layers to handle user input and manipulate data. It will be responsible for the following tasks:

- ➢ *User and driver management*

- *Car rental and driver booking management*
- *Payment processing*
- *Location tracking*
- *Push notification*

The application will also use a RESTful API architecture to communicate between the mobile application and the server. The API will use HTTP methods to perform CRUD (Create, Read, Update, Delete) operations on the server.

The following are the endpoints of the API:

- */user: For user management operations*
- */driver: For driver management operations*
- */car-rental: For car rental management operations*
- */driver-booking: For driver booking management operations*
- */payment: For payment processing operations*
- */location-tracking: For location tracking operations*

The API will use JSON (JavaScript Object Notation) format to exchange data between the mobile application and the server.

In conclusion, the SDS provides a detailed specification for the development of "MyDrive - Car Rental & Driver Booking App". The document outlines the design considerations, system architecture, and software architecture required for the development of the application. With this specification, the development team and stakeholders can have a clear understanding of the requirements and design of the application, which will aid in the development, deployment, and maintenance of the application.

4 Design Strategy

The design strategy for the "MyDrive - Car Rental & Driver Booking App" will be centered around modularity, scalability, and reusability. The overall architecture will be divided into different layers, each with its own responsibilities, to ensure that the system is modular and easy to maintain. The application will also be designed to be scalable, allowing for future system extension or enhancement.

To achieve these goals, the following design decisions will be employed:

- *Layered Architecture: The application will be organized into different layers, including the presentation layer, business layer, and data layer. Each layer will have its own responsibilities, making the system more modular and easier to maintain. The presentation layer will handle user interaction, while the business layer will implement the application's logic, and the data layer will manage data storage and retrieval.*

- *Service-Oriented Architecture (SOA): The application will use a service-oriented architecture, with each service representing a specific functionality. This approach will make it easier to add or modify services, making the system more scalable and flexible.*

- *User Interface: The user interface will be designed to be intuitive and easy to use, with a focus on user experience. The application will employ modern UI paradigms and design patterns, such as responsive design, to ensure that the application works seamlessly on different devices.*

- *Data Management: Data will be stored in a centralized database, which will ensure data consistency and make it easier to manage data. The application will use techniques like caching to optimize data access and improve performance.*

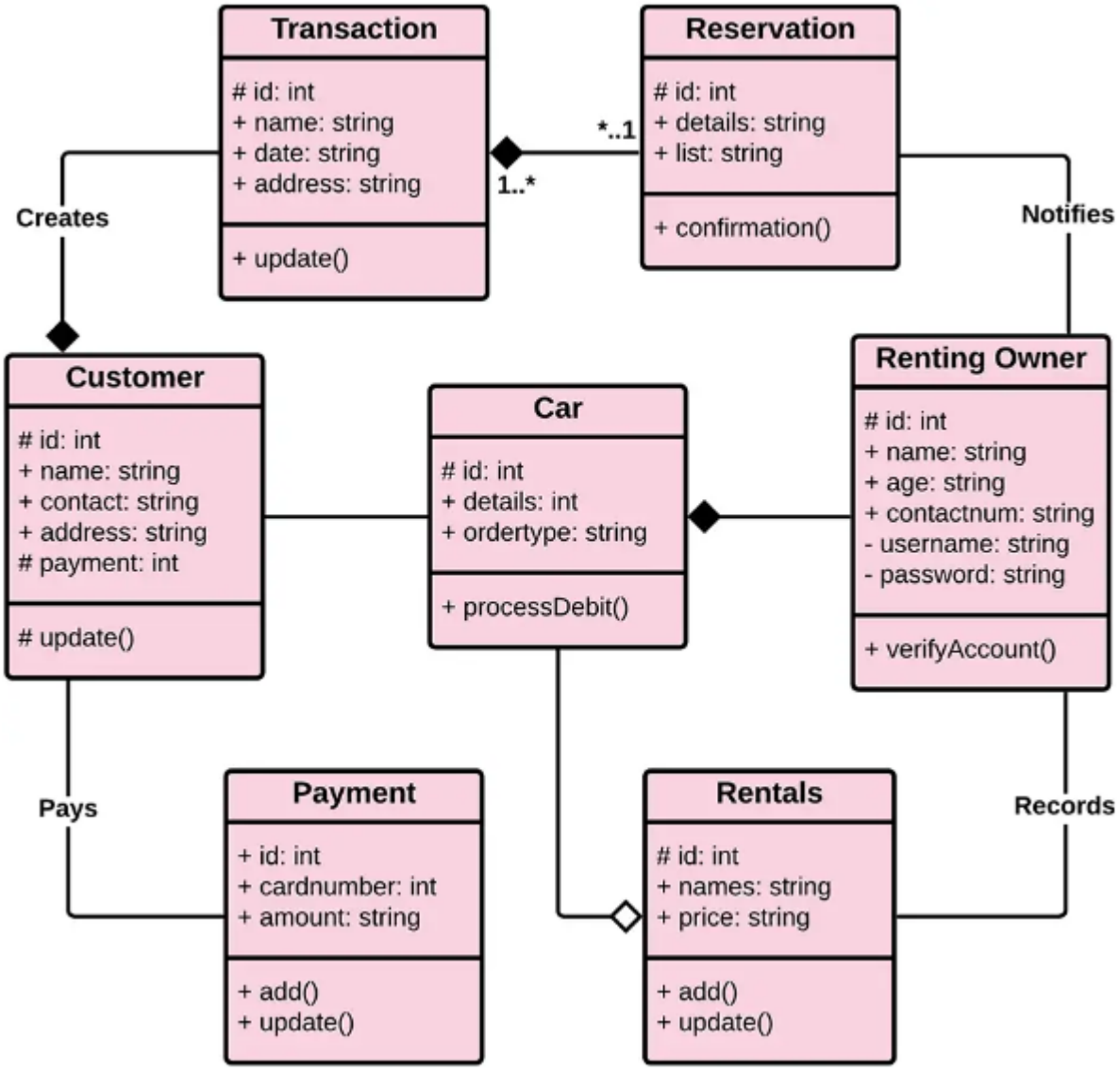
- *Concurrency and Synchronization: The application will e designed to handle concurrent requests and synchronize data access to ensure data integrity. Techniques like locking and multi-threading will be employed to achieve this goal.*

Trade-offs:

The main trade-off associated with the design strategy is that it may result in increased development time and complexity. However, the benefits of modularity, scalability, and reusability will outweigh these drawbacks in the long run. By adopting a layered and service-oriented architecture, the application will be easier to maintain, and it will be more adaptable to future changes or enhancements. Similarly, by investing in user interface design, the application will provide a better user experience and be more attractive to potential users

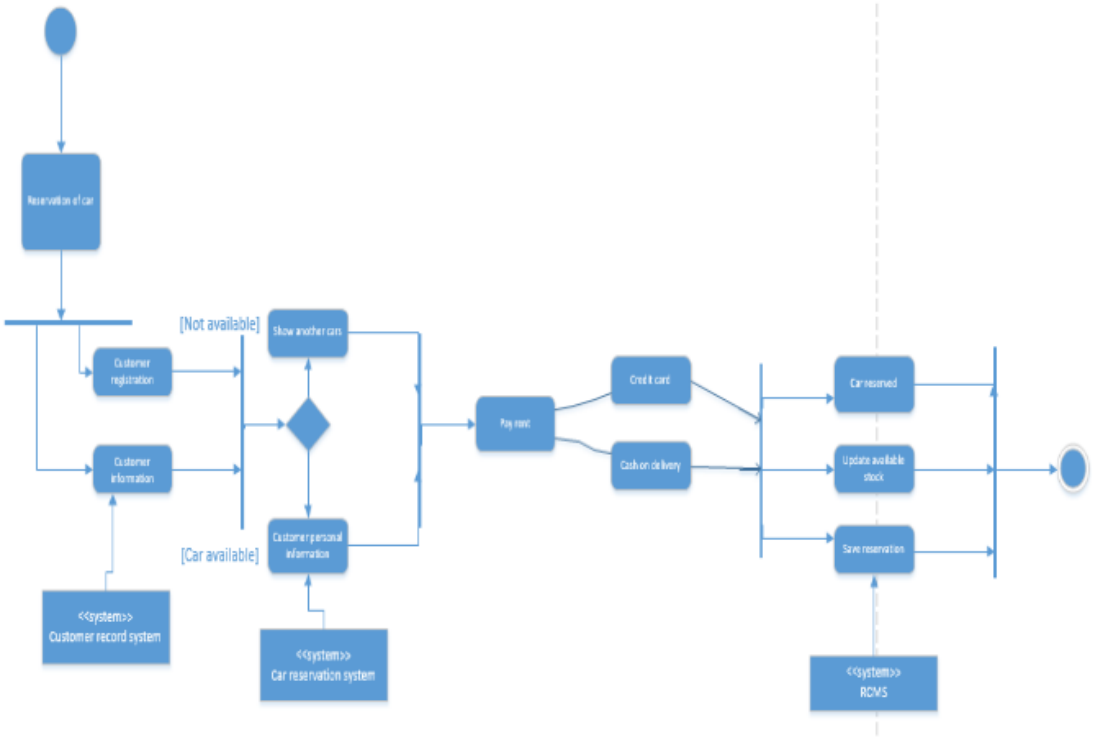
5 Detailed System Design

The following is a detailed class diagram for the "MyDrive - Car Rental & Driver Booking App":



The above class diagram shows the different classes involved in the application, their attributes, and methods. The classes are organized into different layers, including the presentation layer, business layer, and data layer. The presentation layer consists of classes that handle user interaction, while the business layer contains classes that implement the application's logic. The data layer includes classes that manage data storage and retrieval.

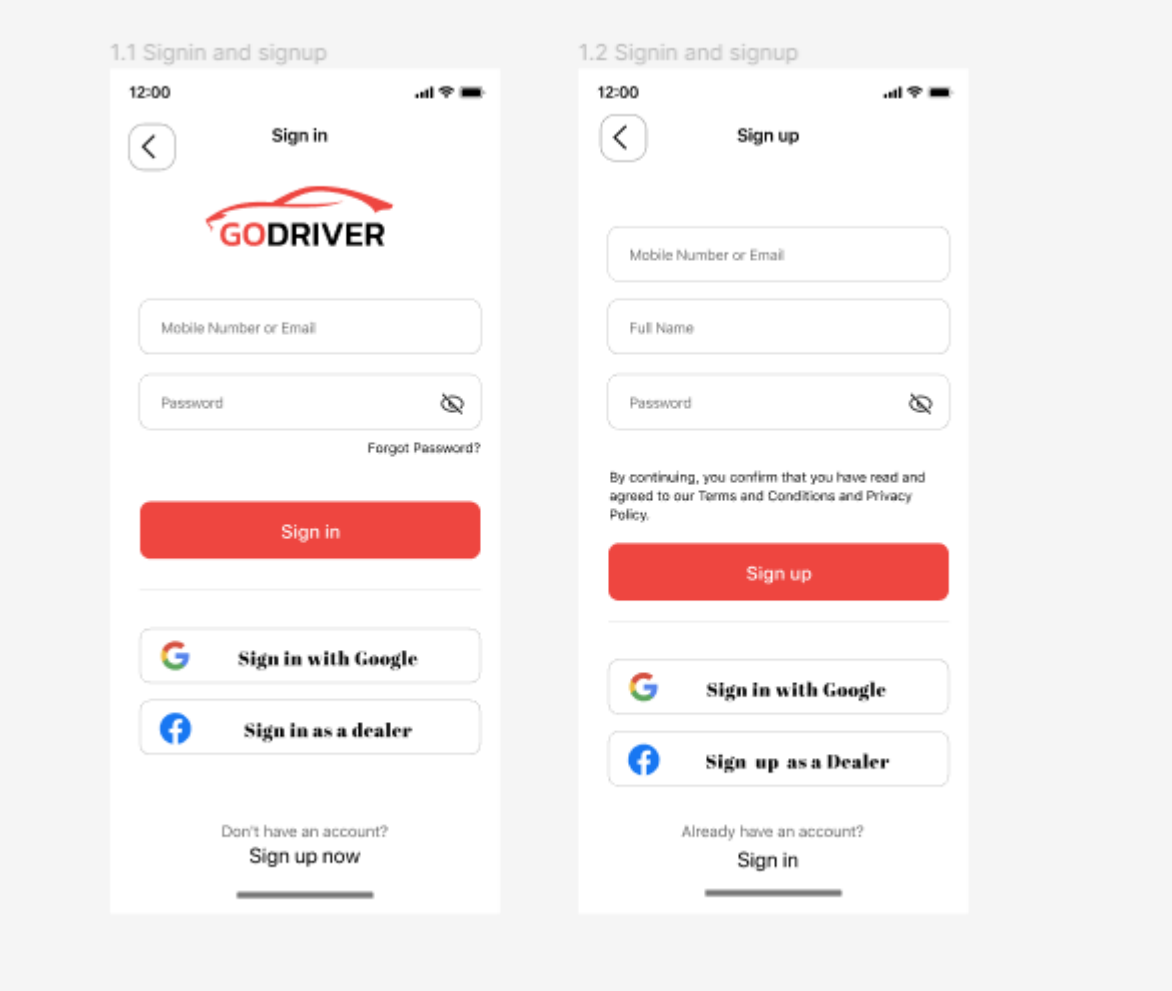
Process Model Flow Diagram



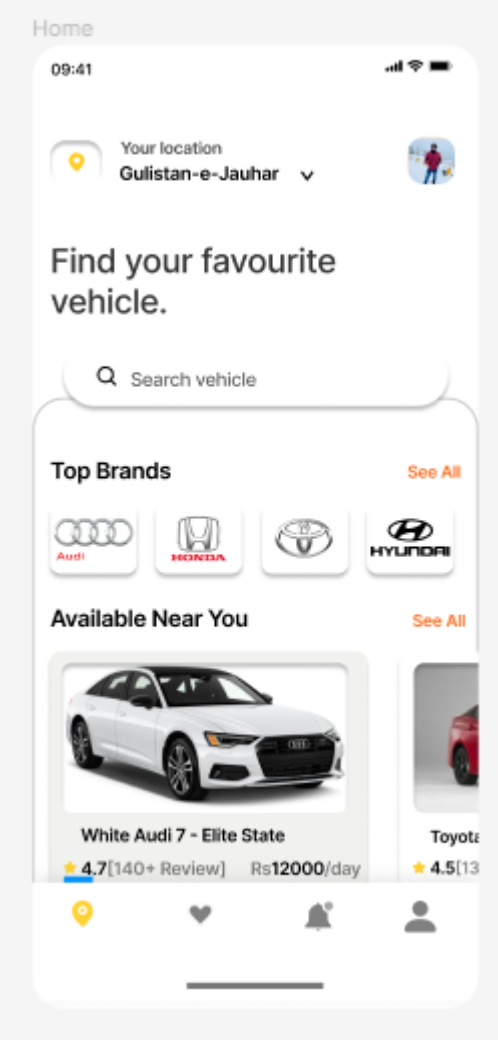
Detailed GUI Design

The following are detailed GUIs for the "GoDrive - Car Rental & Driver Booking App":

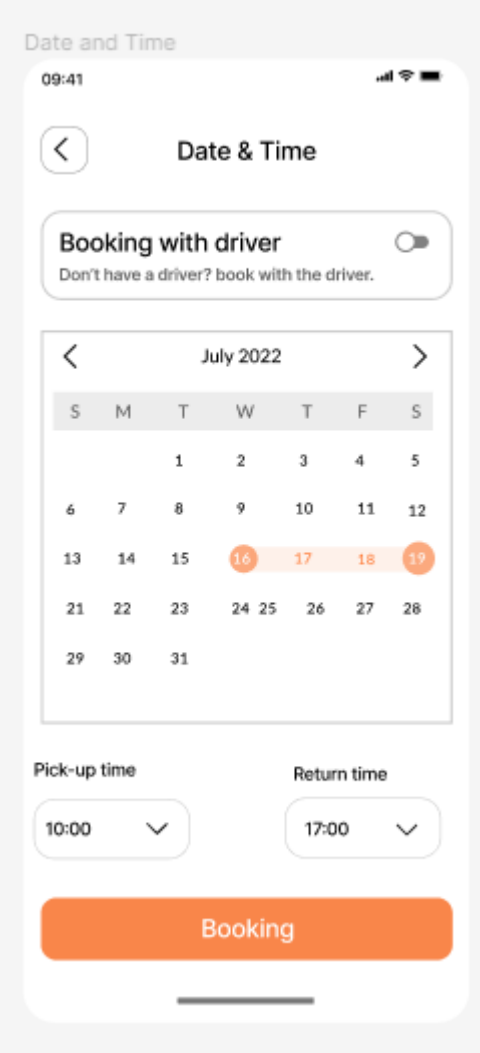
- Login Screen:



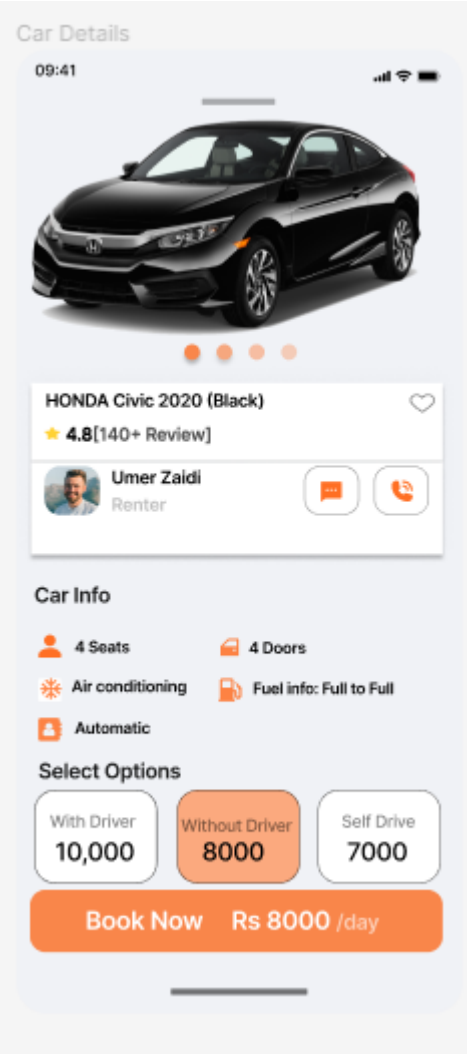
- User Dashboard:



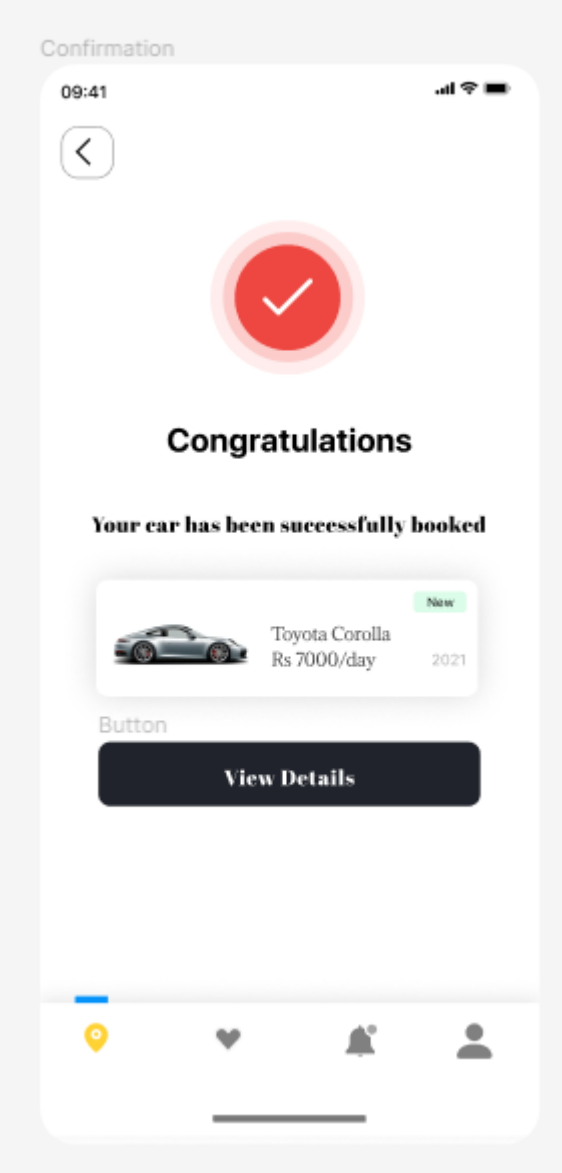
- Date and Time Selection Screen:



- Driver Booking Screen:



- Confirmation Screen:



The above GUIs show the different screens and interfaces involved in the application. They are designed to be intuitive and easy to use, focusing on user experience. The application will employ modern UI paradigms and design patterns, such as responsive design, to ensure that the application works seamlessly on different devices.

5.1 Database Design

User Table

user_id	name	email	password	phone	address
1	Divyansh Chanda	divyanshchanda195@gmail.com	wRaP8	3370567147	North Nazimabad 7117 Block H
2	Abram Dass	abramdass989@gmail.com	L4WYc	3341470870	Karimabad 8345 Allama Iqbal Road
3	Aarav Lala	aaravlala487@gmail.com	FSBtx	3431306032	Lyari 6231 Chakiwara
4	Purab Madan	purabmadan503@gmail.com	ersp6	3177897684	Malir 5729 Moinabad
5	Gokul Bose	gokulbose396@gmail.com	S5juq	3087274914	Gulshan-e-Iqbal 8730 Block 13-D
6	Stuvan Varkey	stuvanvarkey678@gmail.com	vwHfy	3247008540	Nazimabad 1954 Block 4
7	Samar Walla	samarwalla802@gmail.com	S2JMr	3234141769	KDA Scheme 1 060 Tipu Sultan Road
8	Madhup Deshmukh	madhupdeshmukh422@gmail.com	FMRTc	3289728684	Malir 5194 Model Colony

Driver Table

driver_id	name	photo	rating	location	phone
1001	Nitara Sarma	NULL	4	M.A. Jinnah Road, 19221, North Nazimabad	3461534732
1002	Dhruv Krishna	NULL	1	Khalid Bin Waleed Road, 44983, Johar	3295061494
1003	Jiya Saraf	NULL	1	Zainab Market, 15499, Defence	3007769601
1004	Suhana Kashyap	NULL	2	M.A. Jinnah Road, 36103, Gulshan	3382117100
1005	Dhanush Dhar	NULL	2	Khalid Bin Waleed Road, 60200, Gulshan	3141482471
1006	Shlok Kala	NULL	2	Khalid Bin Waleed Road, 03457, Defence	3330714268
1007	Vanya Kala	NULL	4	Khalid Bin Waleed Road, 01559, Johar	3426647026
1008	Tushar Vohra	NULL	2	Khalid Bin Waleed Road, 93795, Saddar	3371045147
1009	Raunak Ratta	NULL	3	Zainab Market, 10315, Johar	3435593385

Car Table

car_id	make	model	year	color	registration_number	rental_price_per_day	location	rental_status
2001	Toyota	yaris	2016	White	NGV-931	10000		Available
2002	Toyota	Prius	2009	White	ANF-445	7000		Available
2003	Toyota	Revo	2021	Black	LBY-278	7000		Available
2004	Suzuki	swift	2009	White	TRX-475	8500		Available
2005	Toyota	Revo	2005	White	ARM-105	8000		Available
2006	Toyota	yaris	2019	Black	BEG-163	11000		Available
2007	Audi	A3	2005	White	ENY-047	6500		Available
2008	Suzuki	cultus	2017	White	MJB-263	10000		Available
2009	Honda	Civic	2010	White	GYR-189	7000		Rented

Booking Table

booking_id	driver_id	user_id	booking_type	booking_date	car_id	booking_time
1	1154	727	both	2022-10-31	2669	20:33:00
2	1164	834	both	2021-03-31	2235	04:23:00
3	1199	781	both	2021-11-25	2878	13:26:00
4	1071	375	both	2021-07-25	2877	04:38:00
5	NULL	717	car	2022-07-31	2524	02:29:00
6	1949	972	both	2022-10-18	2211	01:59:00
7	1696	201	both	2021-10-23	2306	21:59:00
8	1178	516	driver	2020-08-16	NULL	12:32:00
9	1284	845	both	2020-04-25	2125	18:37:00

Payment Table

payment_id	booking_id	amount	payment_method	transaction_date	transaction_time
501	14	6000	easypaisa	2021-07-07	04:45:00
502	5	8500	card	2022-07-31	02:29:00
503	29	7500	card	2022-08-04	01:21:00
504	36	8000	easypaisa	2022-12-31	21:11:00
505	40	8500	card	2021-07-20	01:33:00
506	10	5500	card	2022-06-23	16:51:00
507	7	5500	easypaisa	2021-10-23	21:59:00
508	3	6000	easypaisa	2021-11-25	13:26:00
509	16	6000	card	2020-10-29	08:11:00

5.1.1 ER Diagram

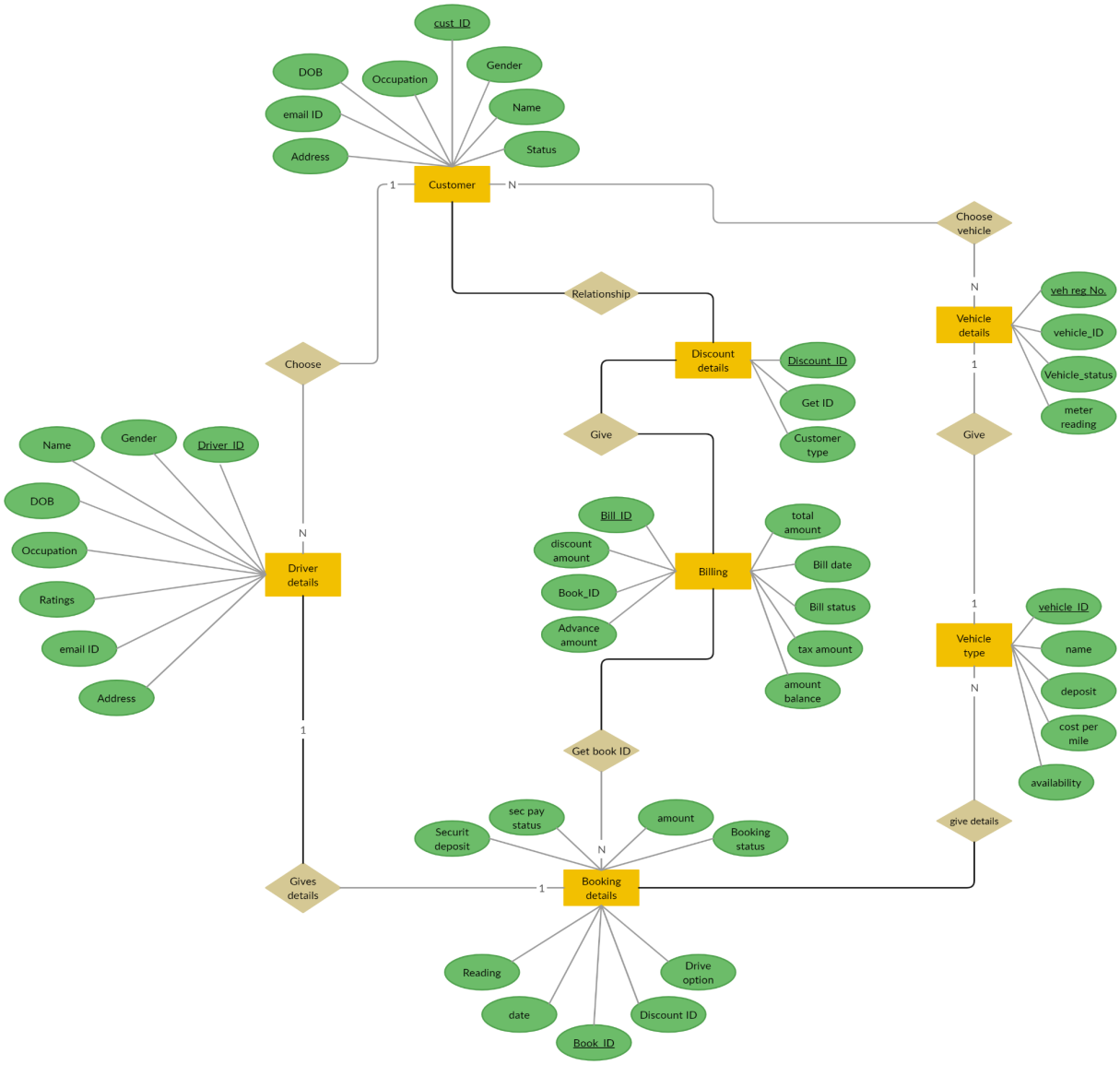
In this ER diagram, we have the following entities and relationships:

- ☐ User: Represents a user of the application. A user can rent cars and book drivers.
- ☐ Car: Represents a car available for rental. A car can be rented by many users.
- ☐ Driver: Represents a driver available for booking. A driver can be booked by many users.
- ☐ Booking: Represents a booking made by a user for a car or driver. A booking is associated with one user, one car or driver, and one payment.
- ☐ Payment: Represents a payment made for a booking. A payment is associated with one booking.

The relationships between these entities are as follows:

- ☐ User can rent many cars (one-to-many)
- ☐ User can book many drivers (one-to-many)
- ☐ Car can be rented by many users (many-to-many)

- [ ] Driver can be booked by many users (many-to-many)
- [ ] Booking is associated with one user (one-to-many)
- [ ] Booking is associated with one car or driver (polymorphic one-to-many)
- [ ] Booking is associated with one payment (one-to-one)



5.1.2 Data Dictionary

5.1.2.1 User Table

Field Name	Data Type	Length	Constraints	Description
UserID	int		PRIMARY KEY	Unique identifier for the user
Name	varchar	50	NOT NULL	Name of the user

Email	varchar	100	NOT NULL, UNIQUE	Email address of the user
Phone	varchar	20	NOT NULL	Phone number of the user
Password	varchar	50	NOT NULL	Password for the user's account

5.1.2.2 Car Table

Field Name	Data Type	Length	Nullable	Description
car_id	Integer	11	No	Unique identifier for the car
make	Varchar	50	No	Car make
model	Varchar	50	No	Car model
year	Integer	4	No	Car manufacturing year
color	Varchar	50	No	Car color
daily_rental_rate	Decimal	10,2	No	Daily rental rate for the car

5.1.2.3 Driver Table

Field Name	Data Type	Length	Constraints	Description
DriverID	int		PRIMARY KEY	Unique identifier for the driver
Name	varchar	50	NOT NULL	Name of the driver
Email	varchar	100	NOT NULL, UNIQUE	Email address of the driver
Phone	varchar	20	NOT NULL	Phone number of the driver
LicenseNo	varchar	50	NOT NULL	License number of the driver
CarID	int		FOREIGN KEY	ID of the car assigned to the driver

5.1.2.4 Booking Table

Field Name	Data Type	Length	Constraints	Description
BookingID	int	10	PRIMARY KEY	Unique identifier for the booking
CarID	int	10	FOREIGN KEY	ID of the car rented in the booking
DriverID	int	10	FOREIGN KEY	ID of the driver assigned to the booking
UserID	int	10	FOREIGN KEY	ID of the user who made the booking

PickupTime	datetime		NOT NULL	Time when the car will be picked up
ReturnTime	datetime		NOT NULL	Time when the car will be returned
PickupLocation	varchar	200	NOT NULL	Location where the car will be picked up
ReturnLocation	varchar	200	NOT NULL	Location where the car will be returned
TotalAmount	decimal	10,2	NOT NULL	Total amount charged for the booking

5.1.2.5 Car Rental Table

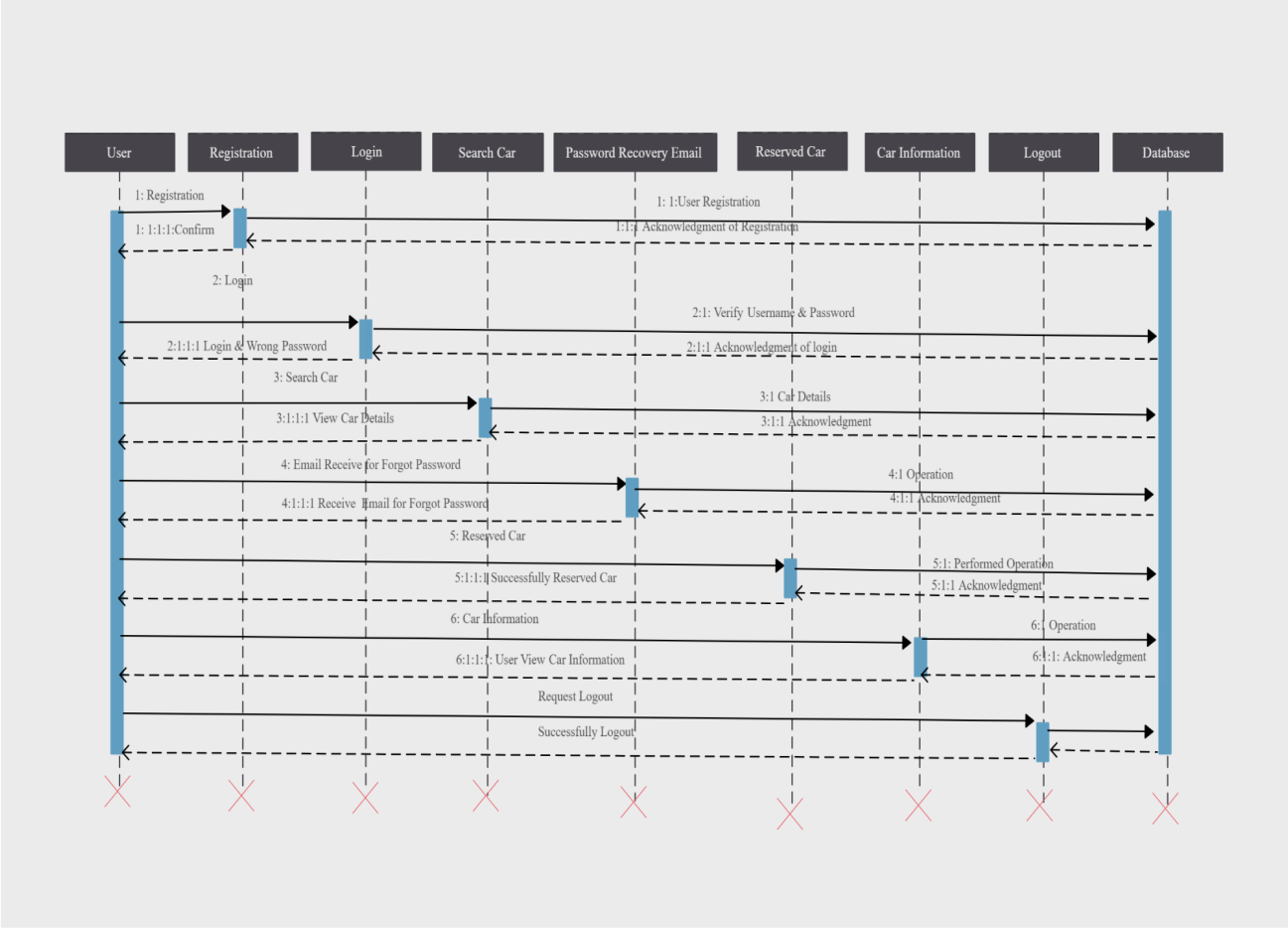
Field Name	Data Type	Length	Nullable	Description
rental_id	Integer	11	No	Unique identifier for the rental
user_id	Integer	11	No	Foreign key to the User table
car_id	Integer	11	No	Foreign key to the Car table
start_date	Date	-	No	Start date of the rental
end_date	Date	-	No	End date of the rental
rental_status	Varchar	50	No	Status of the rental (e.g., active, cancelled, completed

5.1.2.6 Payment Table

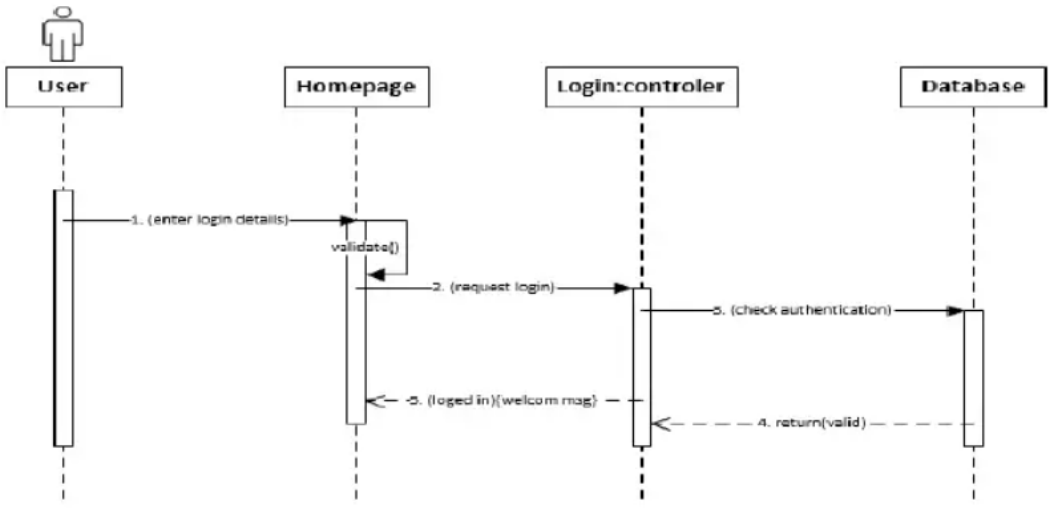
Field Name	Type	Length	Nullable	Description
payment_id	integer	11	No (Primary Key)	Unique identifier for the payment
booking_id	integer	11	No (Foreign Key)	Foreign key referencing the Booking table
amount	decimal	11	No	Amount paid for the booking
payment_date	date		No	Date of the payment
payment_method	varchar	50	No	Payment method used (e.g. credit card, PayPal)

Application Design

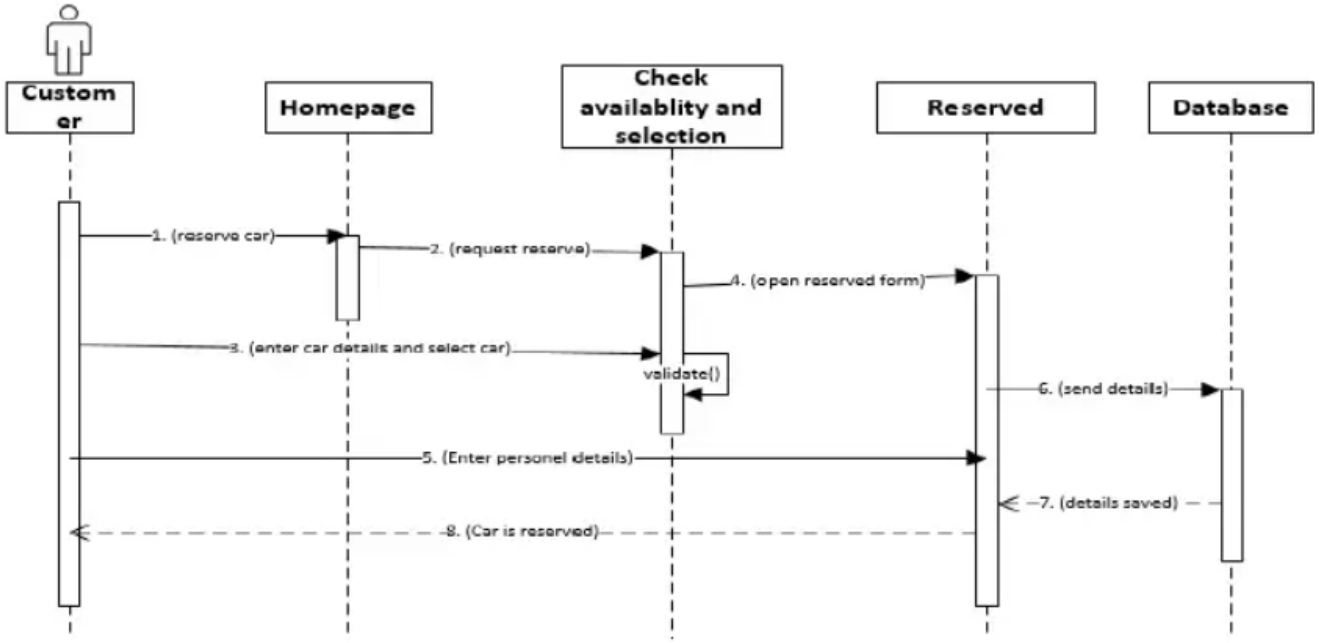
5.2.1 Sequence Diagram



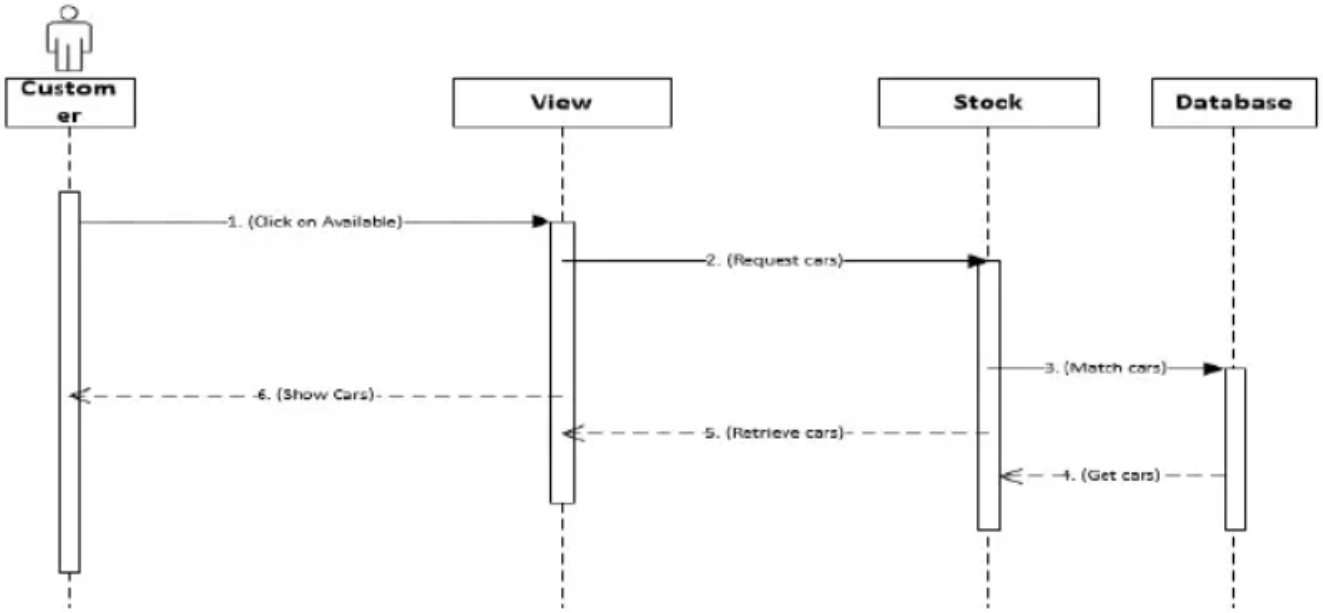
Login



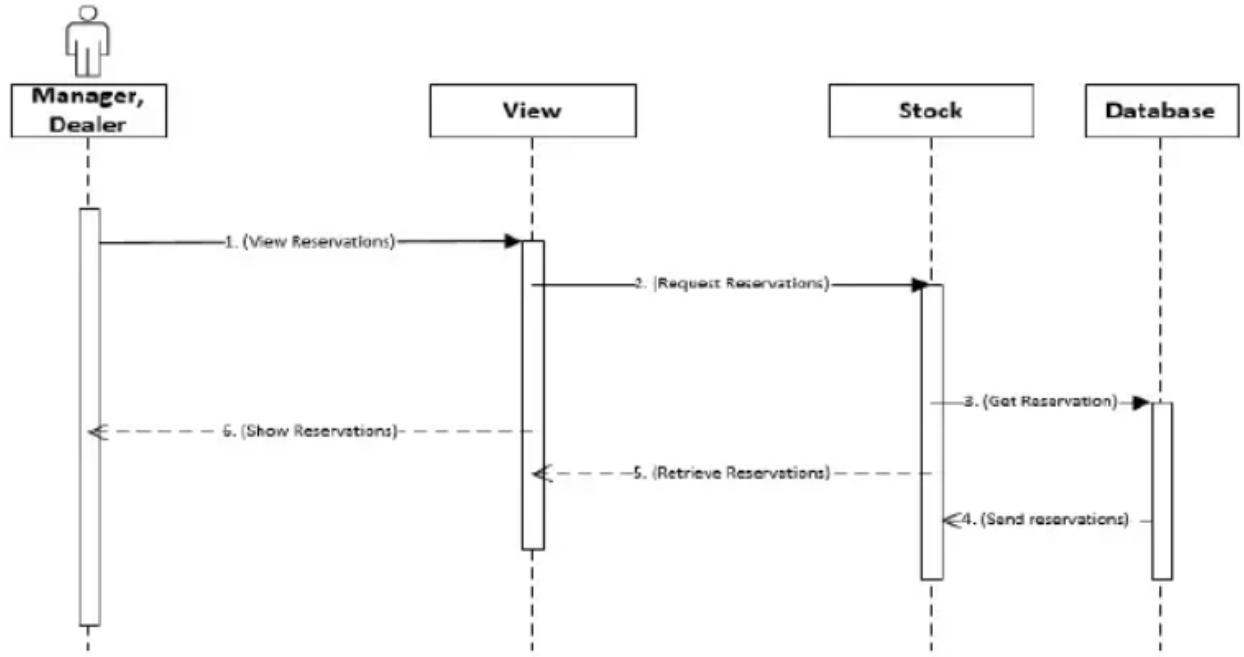
Car Reservation



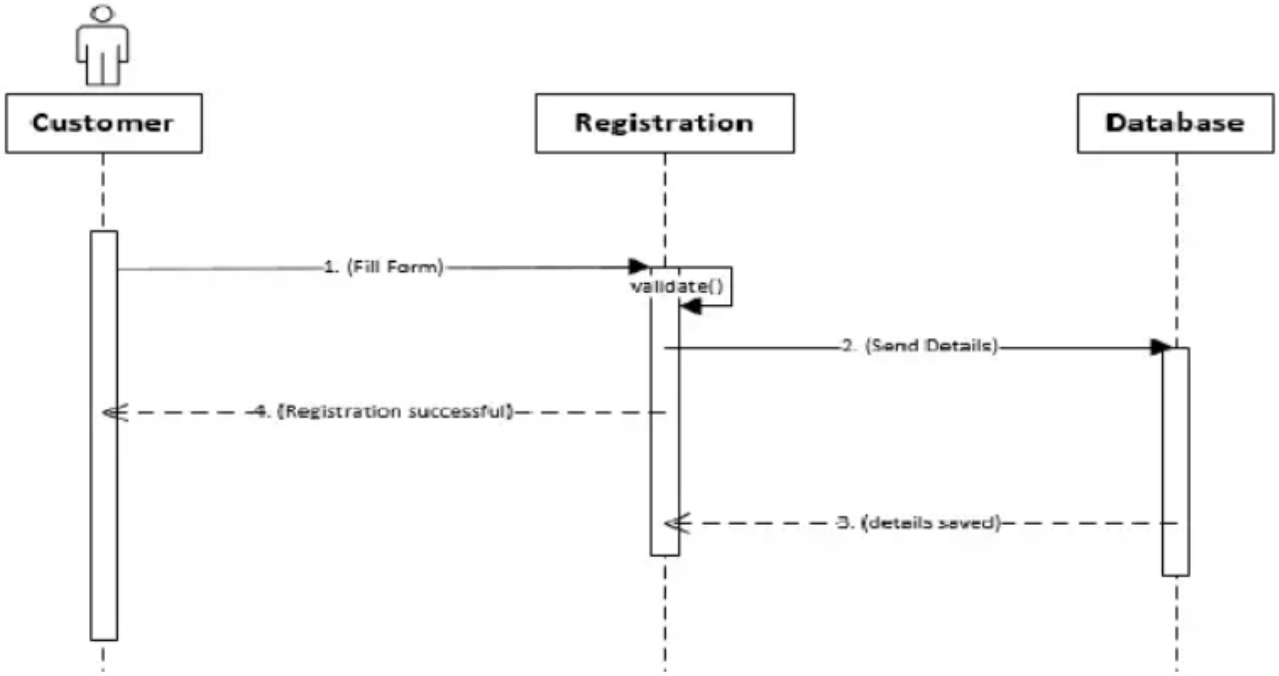
View Available Cars / Drivers



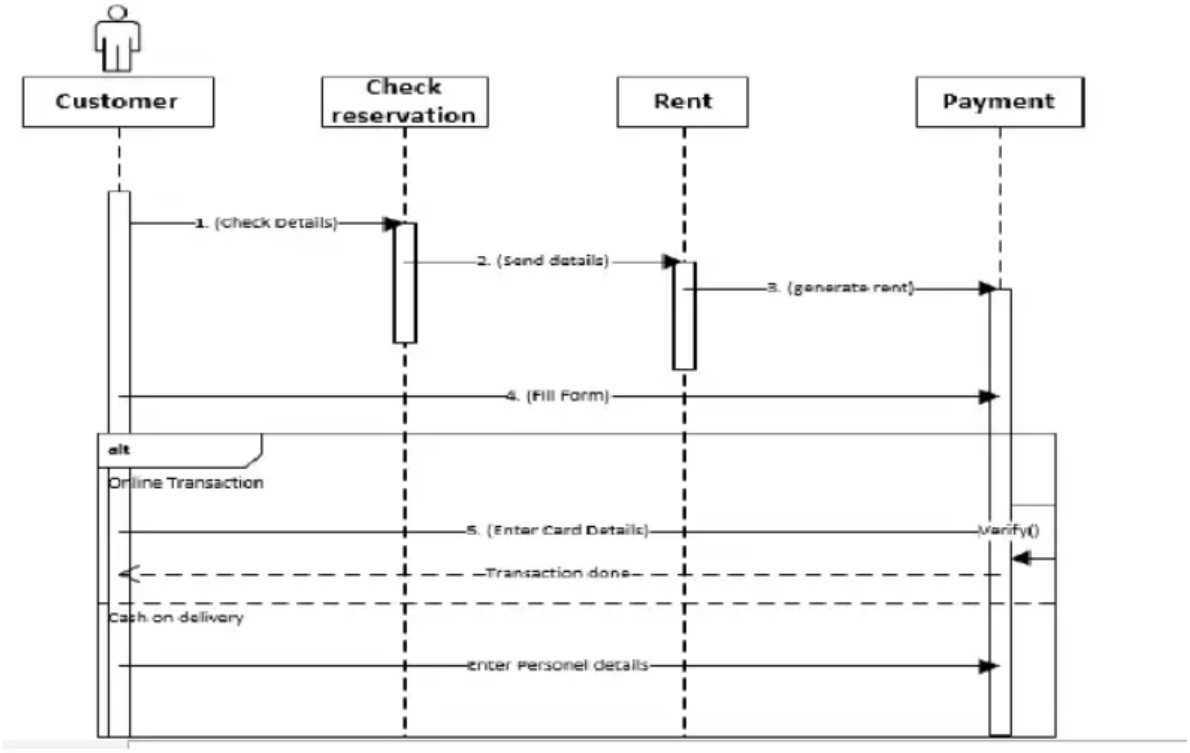
Book driver



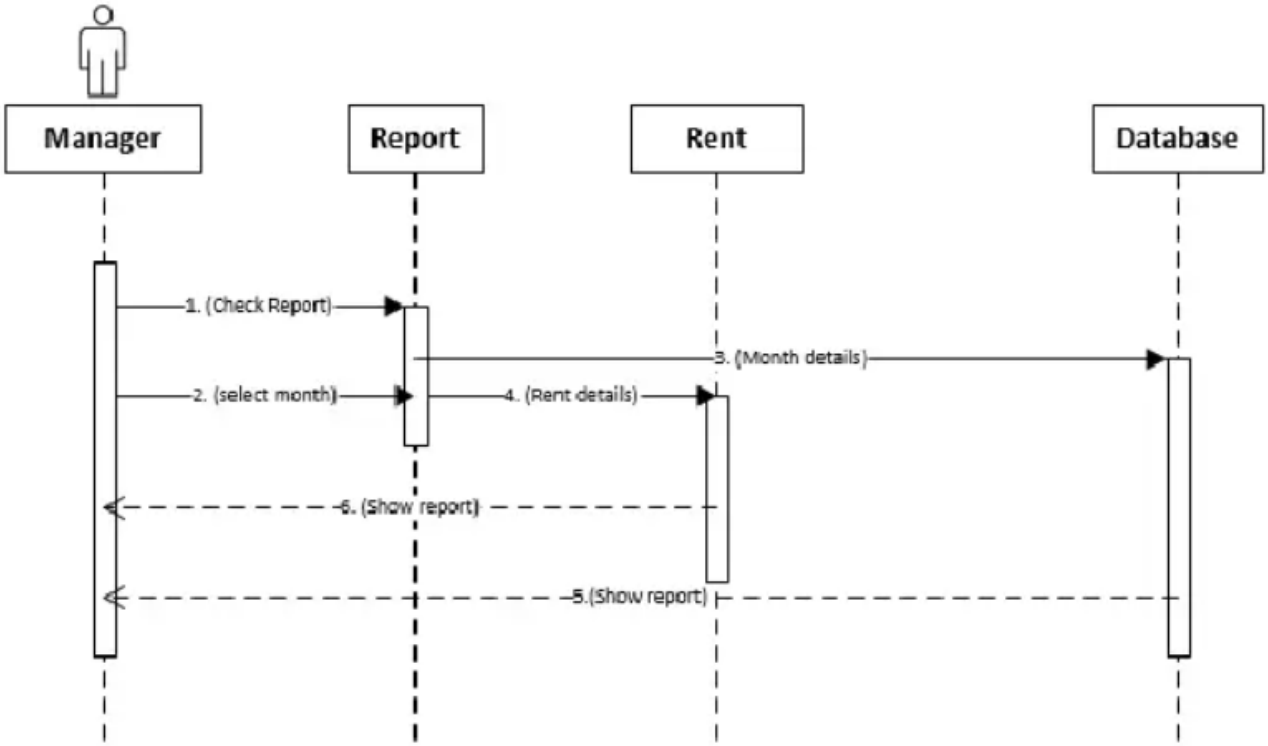
Registration



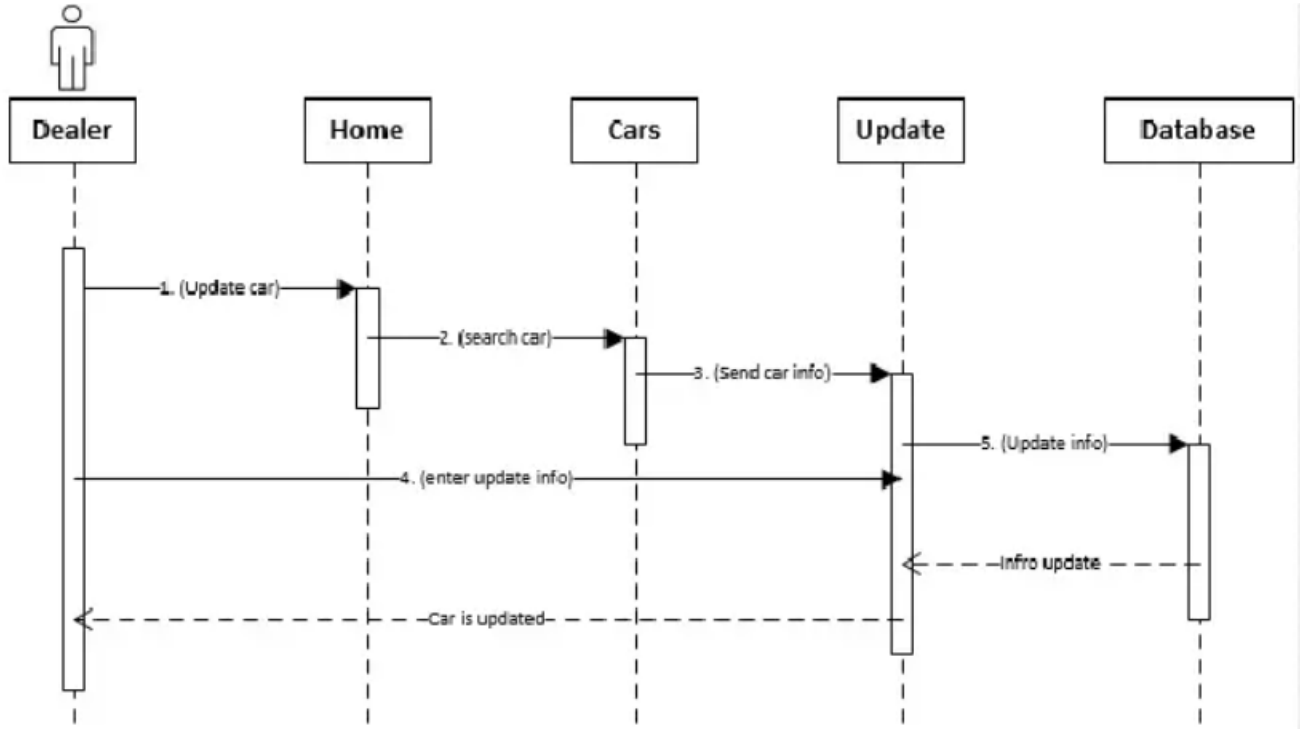
Payment Module



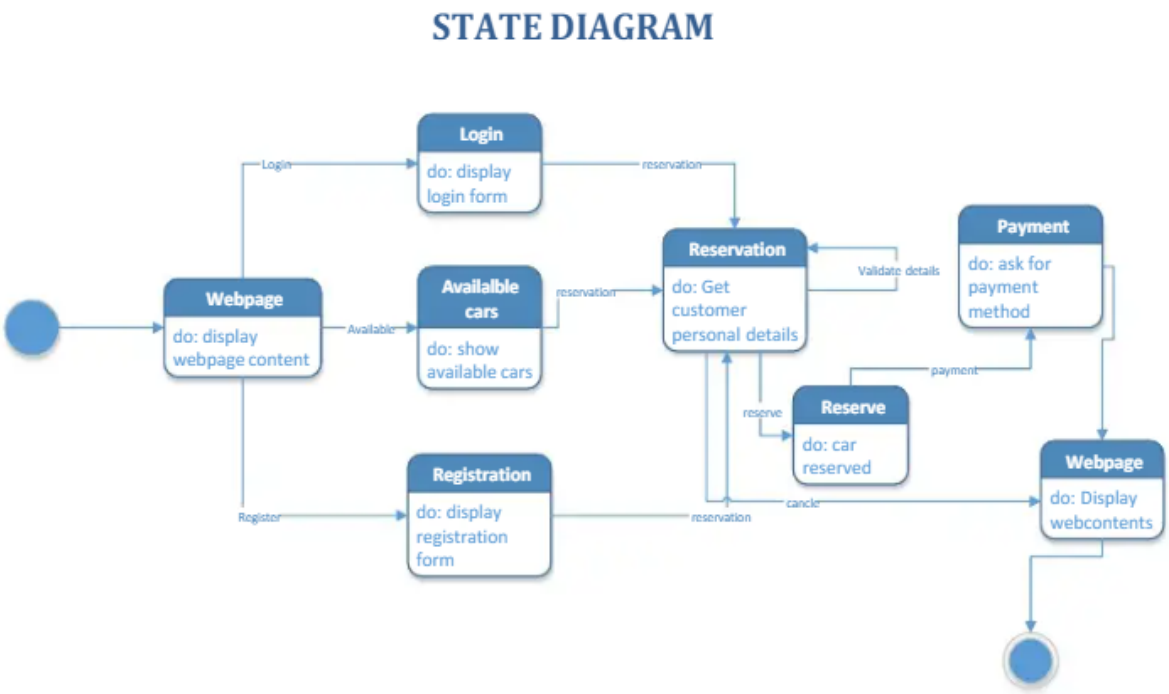
Generate report



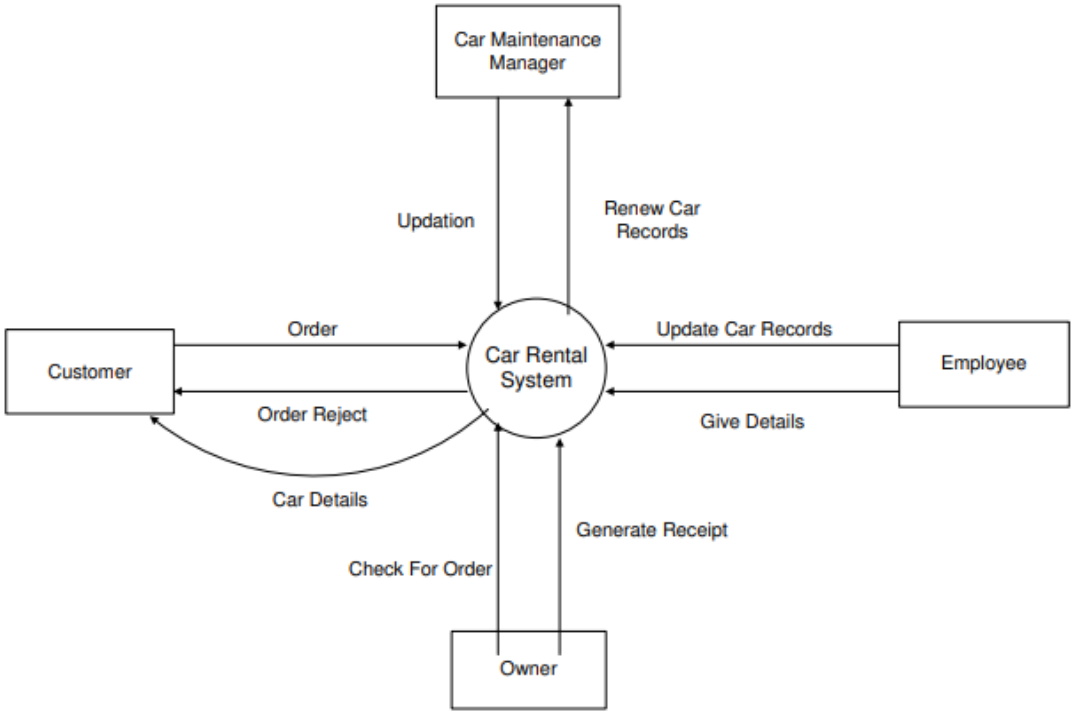
Update car



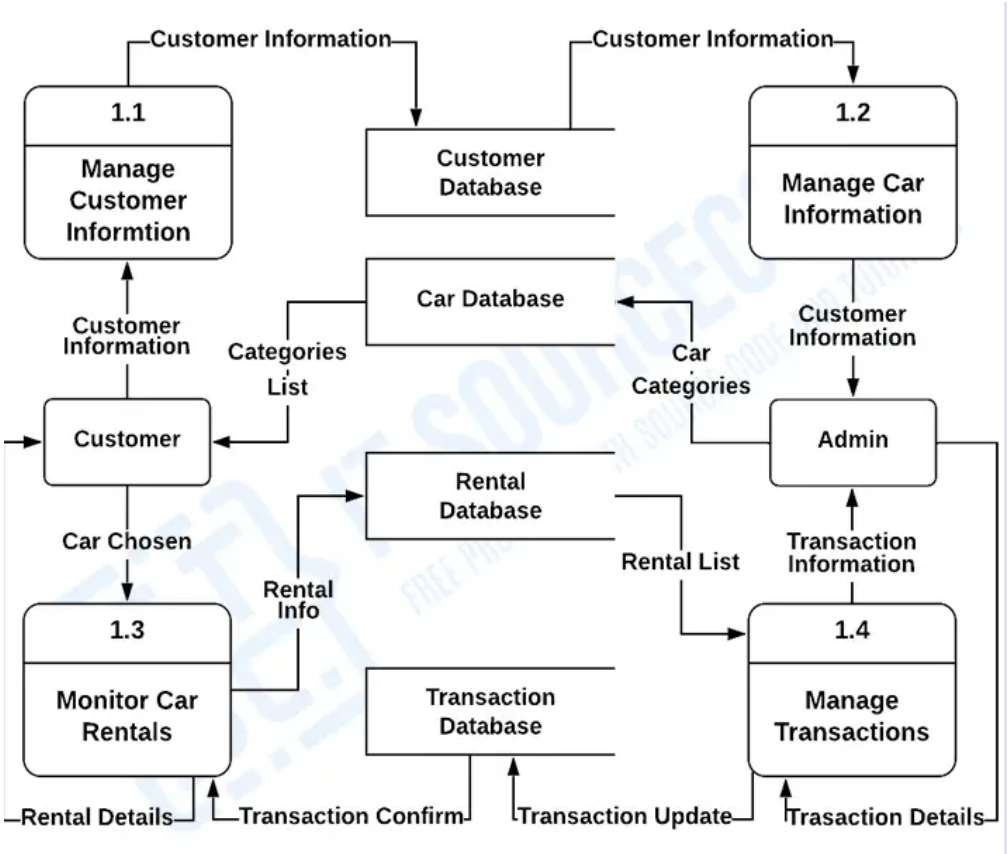
5.2.2 State Diagram



5.2.3 Data Flow Diagram (Level 1)

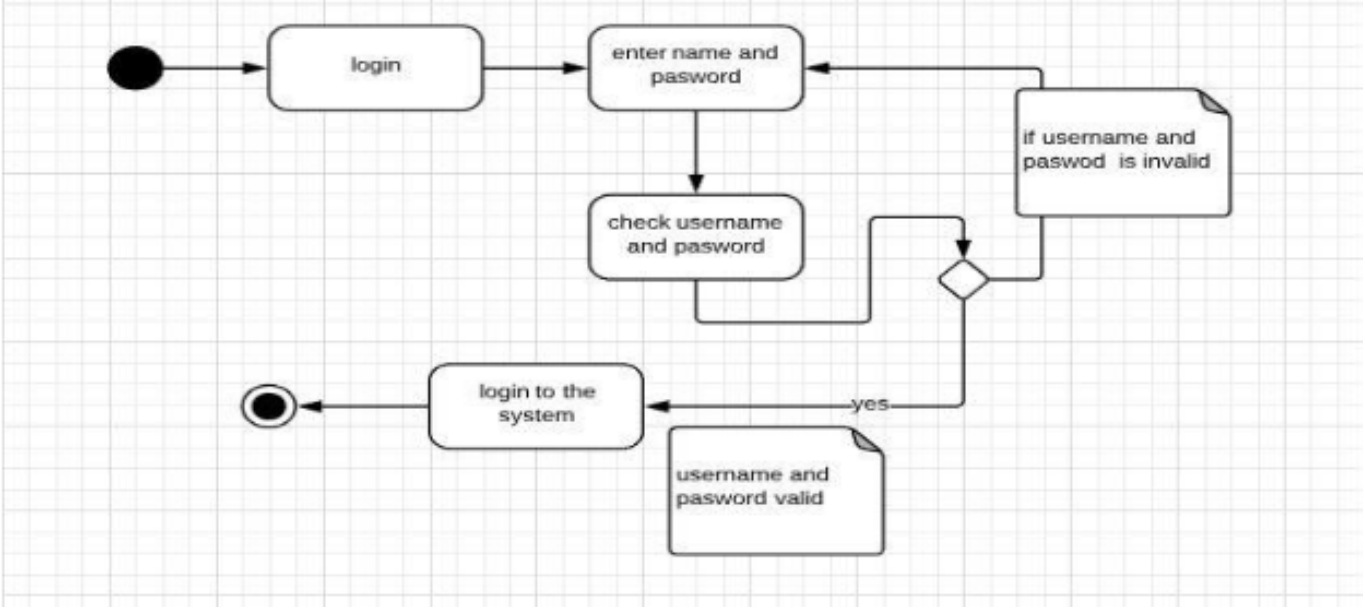


5.2.4 Detailed Data Flow Diagram (Level 2)

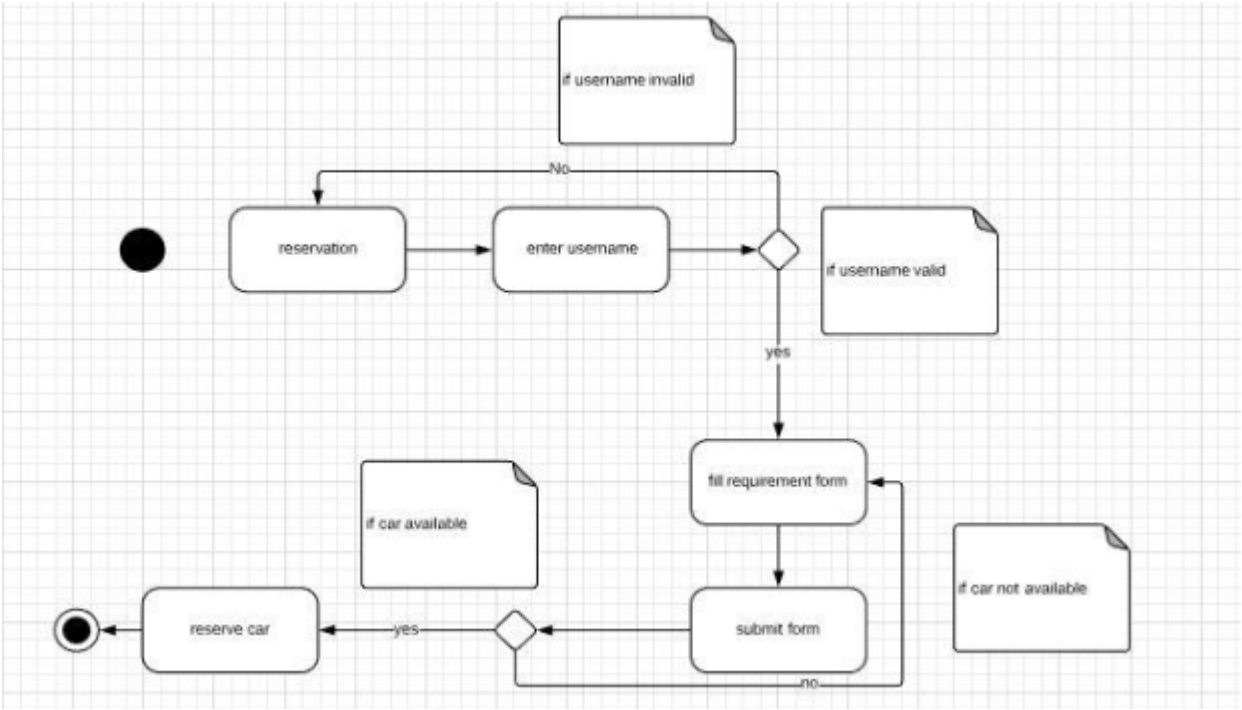


5.2.5 Activity Diagrams

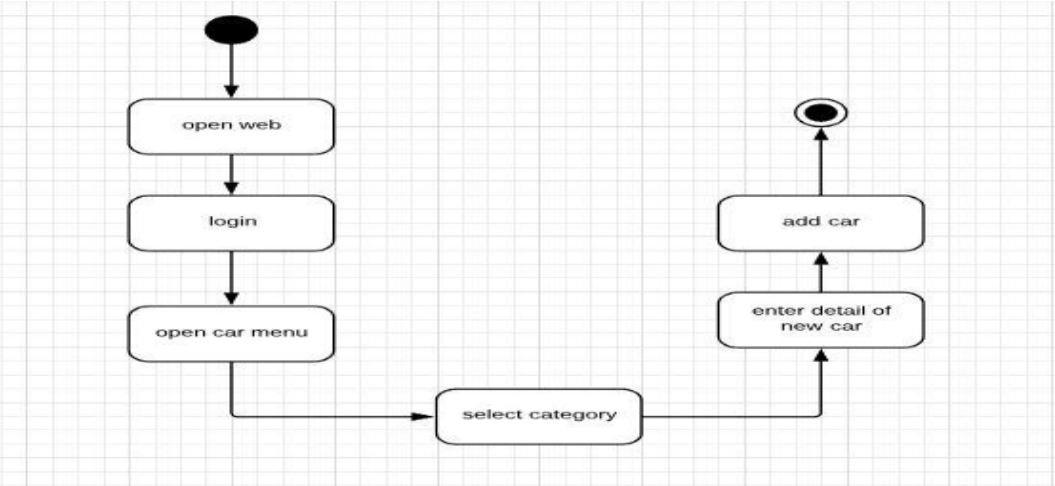
Login



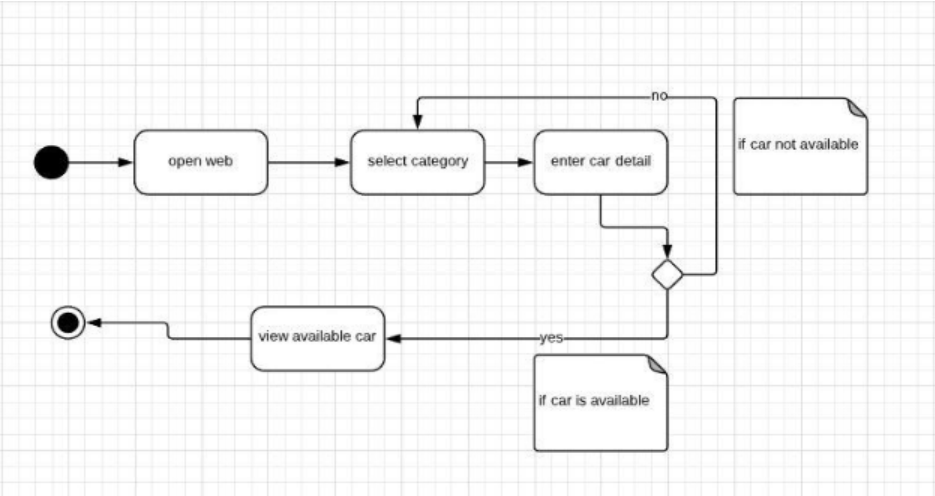
Car Reservation



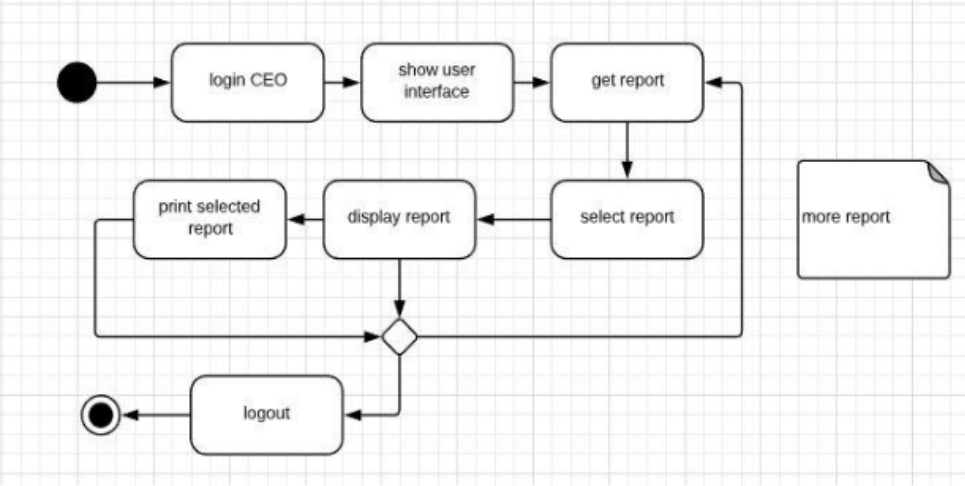
Add Car/Driver



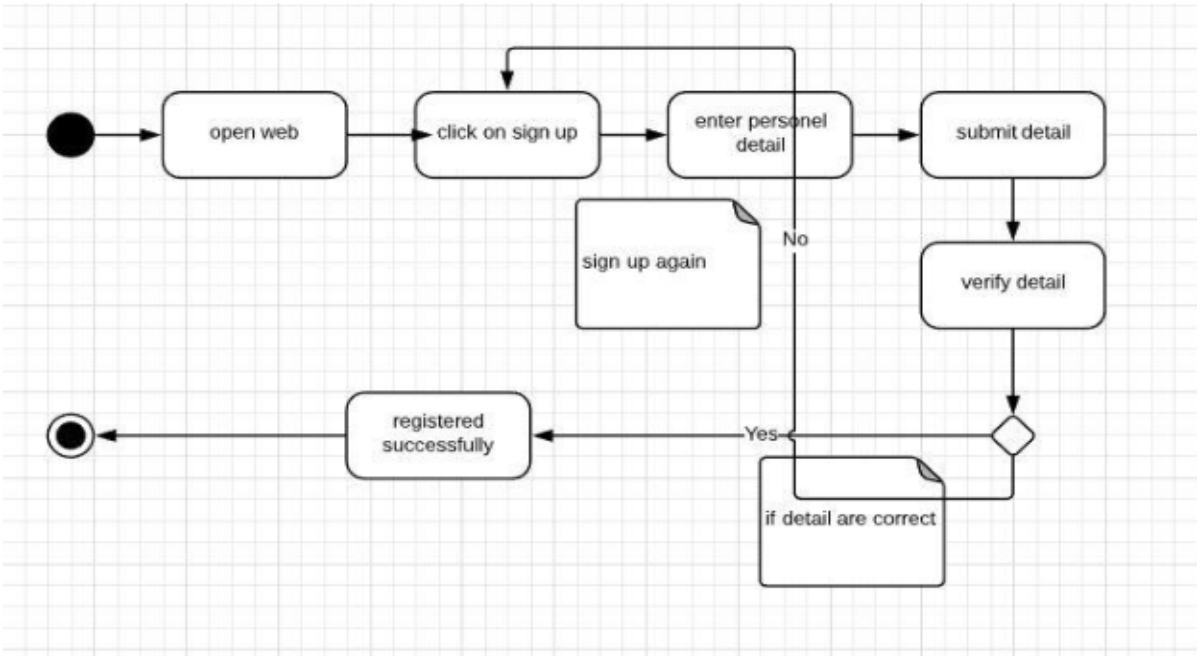
View Available Car/Driver



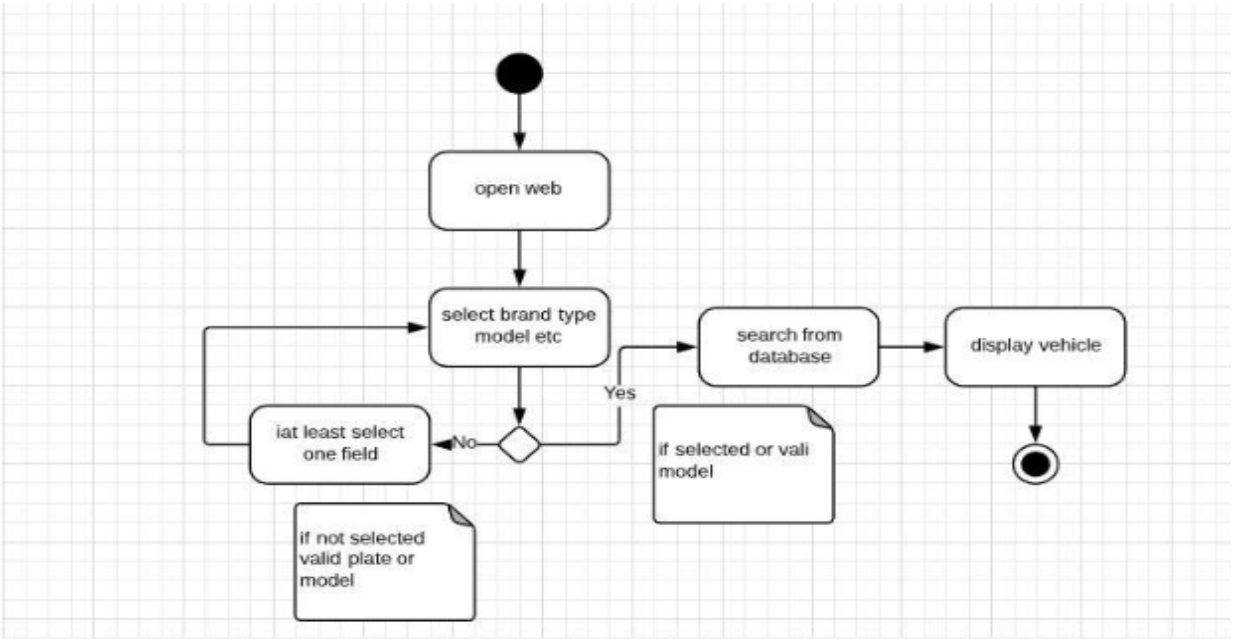
Generate Reports



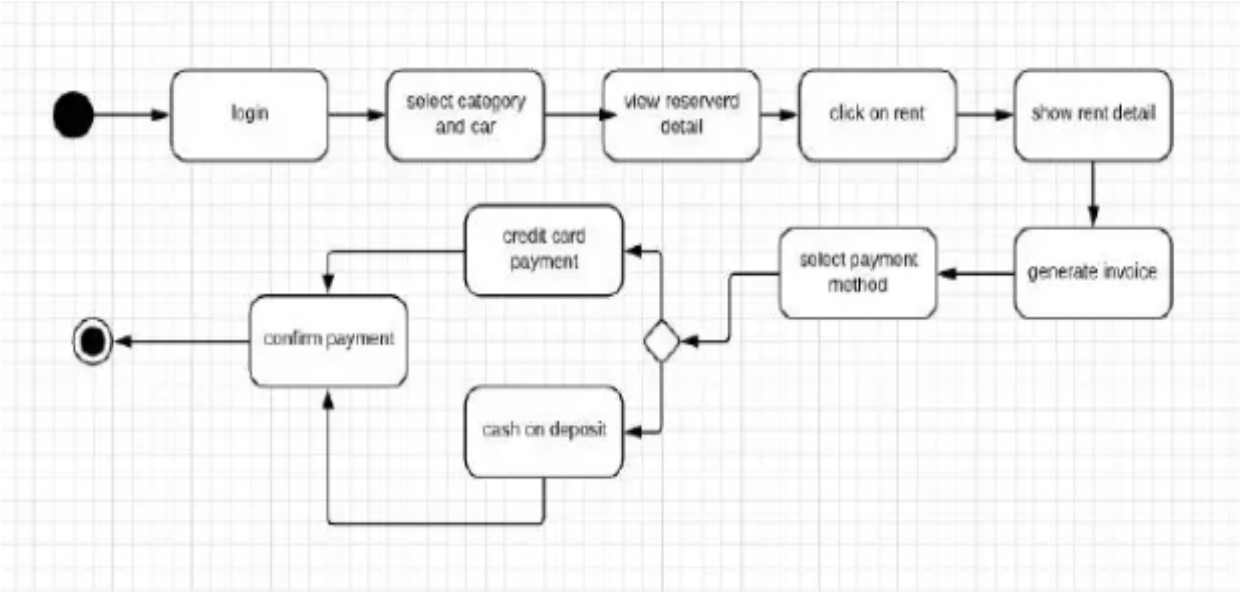
Registration



Search Car/Driver



Payment Gateway



References

- *The system, Online. 'Online Car Rental System'. Academia.edu. , April 20, 2023.*
- *Scribd.com, Online. 'Car Rental System Documentation', April 20, 2023.*
- *https://itsourcecode.com/uml/online-car-rental-system-class-diagram-uml/?utm_content=expand_article*
- *https://www.researchgate.net/figure/Car-Rental-System-Architecture_fig1_325253983*