Ahsanullah University of Science & Technology

Department of Computer Science & Engineering

Final Document On

U-DRIVE – A Car Rental System

Prepared For:

Dr. Taslim Taher	**Assistant Professor**
Nafis Fuad	**Lecturer**

Course No: CSE3224

Course Title: Information System Design and Software Engineering Lab

Submitted By: Group 5 (A1)

Zarin Tasnim	**20200104007**
Nusaiba Rafiq Surovi	**20200104011**
Munim Shahriar	**20200104020**
Azizah Mamun Abha	**20200104021**

Date: 24.08.2023

Table of Contents

Contents:

1. Project Overview:

A car rental system (CRS) is a website-based system for a company that rents out cars. This system will enable the company to make its services available to the public through the Internet and also keep records of its services. This is a company that rents automobiles for a short period of time for a few days or weeks. Car rental companies operate by purchasing or leasing a number of fees. To make this service more popular and accessible to the public it has been transformed into a web base system and connected to the Internet where everyone can be able to have access to it. To develop a web-based system that will help manage the business transactions of car renting. Rental fleets can be structured in several ways they can be owned outright, they can be leased, or they can be owned under a guaranteed buy-back program arranged directly through a manufacturer or manufacturer's financial arm. The primary Objective of our project is to design and implement an online car rental system through which customers can reserve a car from a car renting business and the secondary Objective is to provide an easy platform for people who are renting cars, where they can easily search and reserve their cars online.

1.1 Implementation Tool/Framework

- HTML, CSS, Javascript, PHP
- XAAMP
- Visual Studio Code
- MySql

1.2 User Story:

As a potential customer of the Car Rental System, I want to easily access and utilize the website to rent a car for my transportation needs, So that I can have a seamless and convenient experience in renting a vehicle.

- **Registration:**

As a new user, I want to create an account using my personal details, providing my name, email address, and a secure password. After entering valid information, I should receive a verification email to confirm my registration.

- **Logging In:**

As a registered user, I want to log in with my email and password securely. I should be able to reset my password if I forget it, receiving an email with instructions to do so.

- **Browsing and Searching:**

Upon logging in, I want to easily navigate through the website to find the "Search Cars" section. I should be able to search for available cars by specifying criteria such as location, date, time, and car type.

- **Selecting a Car:**

From the search results, I want to view detailed information about each car, including its specifications, images, and rental cost. I should be able to select a car by clicking on it and confirming my choice.

- **Renting a Car:**

After selecting a car, I want to specify the rental duration, pickup date, and return date. The system should calculate the total cost based on my selections and display it for confirmation.

- **Booking Confirmation:**

Once I confirm the rental, I want to receive a booking confirmation that includes all relevant details such as car information, rental dates, and total cost.

- **Car Pickup:**

On the designated pickup date and time, I want to receive clear instructions on where to collect the car. The system should allow me to mark the pickup as completed once I have the car in my possession.

- **Car Return:**

At the end of the rental period, I want to easily locate instructions for returning the car. After returning the car, I should be able to mark the return as completed and provide any necessary feedback.

- **Payment:**

Before confirming the rental, I should be able to securely enter my payment details, including credit card information. The system should process the payment and provide me with a receipt that includes a breakdown of the charges.

- **User Profile:**

As a registered user, I want to have a user profile where I can view my booking history, and current reservations, and update my personal information if needed.

- **Feedbacks:**

I want to easily find information about the services offered by the website and be able to contact the website administrators if needed,

- **Services Overview:**

Upon accessing the website, I want to be able to locate a clearly labeled section titled "About Us" in the main navigation menu. Clicking on the "About Us" link should take me to a page that provides an overview of the various services offered by the car rental platform.

- **Contact Information:**

I want to easily find a "Contact Us" link or button prominently displayed on the website. Clicking on the "Contact Us" link should lead me to a page where I can access contact information for the website administrators.

- **Communication Options:**

On the "Contact Us" page, I expect to see multiple communication options, including a contact form, an email address, and a phone number. The contact form should allow me to enter my name, email address, subject, and message, and submit my inquiry directly from the website.

Admin User Story:

As an admin of the car rental platform, I want to improve vehicle management, user interactions, feedback collection, and booking request processing to provide a seamless and efficient experience for both customers and staff.

Admin Login

The admin login page should have fields for entering a valid username and password. Upon successful login, the admin should be directed to the admin dashboard. In case of incorrect login credentials, an error message should be displayed. The login session should be secure and utilize appropriate encryption methods.

Admin Logout

A "Logout" button or option should be available within the admin dashboard. When the admin clicks on the "Logout" option, they should be immediately logged out and redirected to the login page. After logout, any access to the admin dashboard or privileged areas should require re-authentication. The logout process should clear the admin's session and any stored authentication tokens.

Add Car to Car List

As an admin, I want to be able to add new cars to the rental inventory, ensuring that customers have access to an up-to-date selection. I can access the admin dashboard and navigate to the "Add Car" section. I am prompted to provide details about the new car, such as make, model, year, mileage, and availability status. I can upload images of the car to give customers a visual representation. Upon submission, the new car information is added to the car list and becomes available for customers to rent.

Delete the Car from the Car List

As an admin, I want to have the ability to remove cars from the rental inventory when they are no longer available for rent. I can access the admin dashboard and go to the "Manage Cars" section. I am presented with a list of cars currently in the inventory, including their details and availability status. Next to each car, there is a "Delete" button or option. When I click the "Delete" button for a specific car, a confirmation prompt appears. After confirming the deletion, the car is removed from the car list and is no longer visible to customers.

Access User Information

As an admin, I need the ability to access user information for better management and support. In the admin dashboard, there should be a section labeled "User Management" or similar. When I navigate to this section, I can see a list of registered users along with their basic details, such as username, email, and contact information. Clicking on a user's name or profile should provide me with more detailed information about their rental history, ongoing bookings, and feedback (if applicable).

Delete User

As an admin, I need to be able to delete user accounts when necessary, such as in cases of policy violations or account closure requests. In the "User Management" section of the admin dashboard, each user's entry should have a "Delete" button or option. When I select the "Delete" option for a specific user, a confirmation prompt should appear. Upon confirming the deletion, the user's account is permanently removed from the system. Any associated data, such as booking history and feedback, should also be appropriately updated or deleted.

Access User Feedback :

As an admin, I want to access and review the feedback provided by users to gain insights into their experiences and identify areas for improvement. In the admin dashboard, there should be a dedicated "Feedback" or "User Feedback" section. When I navigate to this section, I can see a list of recent feedback entries from users. Each feedback entry should display the user's name, date of submission, and a brief summary of the feedback content. Clicking on a specific feedback entry should allow me to view the full feedback details, including ratings and comments provided by the user.

- **Access Booking Requests for Admin**

As an admin, I need the ability to access and manage booking requests made by users. In the admin dashboard, there should be a dedicated "Booking Requests" section. When I navigate to this section, I can view a list of pending booking requests along with relevant details, such as user information, requested vehicle, rental dates, and status. Each booking request should have options to approve, reject, or view more details.

- **Approve Booking Requests**

As an admin, I want to be able to review and approve booking requests from users. Within the "Booking Requests" section, there should be an "Approve" button or option for each pending request. When I select the "Approve" option for a specific request, a confirmation prompt should appear. Upon approval, the booking status should be updated to "Approved" and the user should receive a confirmation email with booking details.

- **Delete Booking Requests**

As an admin, I need the ability to delete or reject booking requests when necessary. Within the "Booking Requests" section, there should be a "Reject" or "Delete" button for each pending request Clicking the "Reject" or "Delete" option should prompt a confirmation message to ensure the admin's intention. Upon rejection, the booking status should be updated to "Rejected" and the user should be notified via email.

1.3 Module Breakdown:

- **User Authentication:**

Allow users to create accounts and log in securely. Manage user profiles and account information.

- **Car Catalog:**

Display a range of available cars for rent. Implement search and filters for users to find desired cars.

Enable users to select a car for rental.

- **Booking and Rental:**

Provide a process for users to reserve cars for specific dates. Incorporate a payment gateway for secure rental payments. Track rental history for both users and cars. Define a system for users to return rented cars.

- **Feedback and Information:**

Collect feedback from customers about their rental experience. Showcase services offered and offer contact details.

- **Admin Dashboard:**

Admin-specific login for secure access. Enable addition, editing, and deletion of car listings. View and manage customer details and rental records. Handle customer feedback and inquiries.

- **Rental Tracking:**

Keep records of rented cars and rental periods. Calculate rental costs and return dates.

- **Contact and Support:**

Provide users with contact information and support details. Implement a communication channel for user inquiries.

- **Payment Gateway:**

Integrate a secure payment system for rental payments.

- **Notifications:**

Send notifications to users about reservations, payments, etc.

1.3 Collaboration Matrix:

Task	Nusaiba (11)	Zarin (07)	Abha(21)	Munim (20)
Planning & Design Phase	R	R	R	C
Project Planning & Feature Selection	R	R	A	A
Create a Roadmap	C	A	R	R
Project Proposal	R	A	R	R
Development Start	R	R	R	R
Database Connection & Design	R	R	R	C
Implementation of Login Page	R	A	C	A
Homepage Design	R	C	A	C
User Profile Design	R	A	A	I
Rental Page Design	R	R	R	A
Transaction Gateway Design	R	I	R	R
Admin Dashboard Design	R	R	R	A
User Feedback Recording Implementation	A	R	R	C
Other UI Designs	R	R	R	I
Backend Implementation of Frontend	R	R	R	R
Testing & Quality Assurance	R	R	R	A
Debugging	R	R	R	R
Cross-Device Testing	R	R	R	R
Report Writing & Presentation	R	R	R	R
Final Launch	R	R	R	R

Fig: Collaboration Matrix

R= Responsible for the Task

A = Accountable for the overall success or failure of the Task

C= Consulted for the expertise

I=Informed about the progress, decision

2. Process Modeling:

Our Project is an online car rental system which is a website built with HTML, CSS, JavaScript, and XAMPP. In our opinion, for a car rental website development project that utilizes HTML, CSS, JavaScript, and XAMPP, an Agile development process model would likely be the most suitable choice. Here are a few reasons why we think Agile process modeling would be best for our project –

- **Flexibility and Adaptability:**
  Agile methodologies such as Scrum or Karban are known for their flexibility and adaptability to changing requirements. As a car rental website, we may encounter evolving business rules, customer feedback, and market demands. An Agile approach allows us to quickly respond to these changes and adjust the development priorities accordingly.

- **Iterative Development:**
  Agile models promote iterative development, where the delivery of working software increments in short cycles called sprints. This approach allows to release of a functional version of the website early on, enabling to gathering of feedback from users and stakeholders. We can then incorporate their input into subsequent iterations, ensuring that the website meets their needs and expectations.

- **Continuous Improvement:**
  Agile methodologies encourage continuous improvements through regular retrospectives, where the development team reflects on their work and identifies areas for enhancement. This feedback-driven approach helps to identify and address issues promptly, leading to a better end product. It also provides opportunities to refine the website's features and user experience throughout the development process.

- **Collaboration and Communication:**
  Agile methodologies emphasize collaboration and communication among team members, stakeholders, and users. As a car rental website is developed, effective communication is crucial to understanding user requirements, resolving queries, and ensuring that the website aligns with business objectives. Agile practices such as daily stand-up meetings and frequent interactions, foster a collaborative environment that facilitates timely decision-making and problem-solving.

- **Quality Assurance**:
  Agile models promote the integration of testing and quality assurance activities throughout the development process. By incorporating testing early and continuously, we can identify and fix defects promptly, ensuring the reliability and stability of our car rental website. This iterative testing approach helps in delivering a higher quality product as compared to traditional models, where testing often occurs toward the end of the development cycle.

Considering these factors, an Agile process model is well-suited for our car rental website development project. It provides the flexibility to accommodate changes, allows for continuous feedback and improvement, fosters collaboration, and ensures a focus on quality throughout the development lifecycle.

3. Prototype:

As a user, anyone can easily browse, select, and reserve rental vehicles according to their preferences and needs. They can access an extensive catalog of available cars, complete with detailed descriptions, images, and pricing information. The website typically shows the homepage.

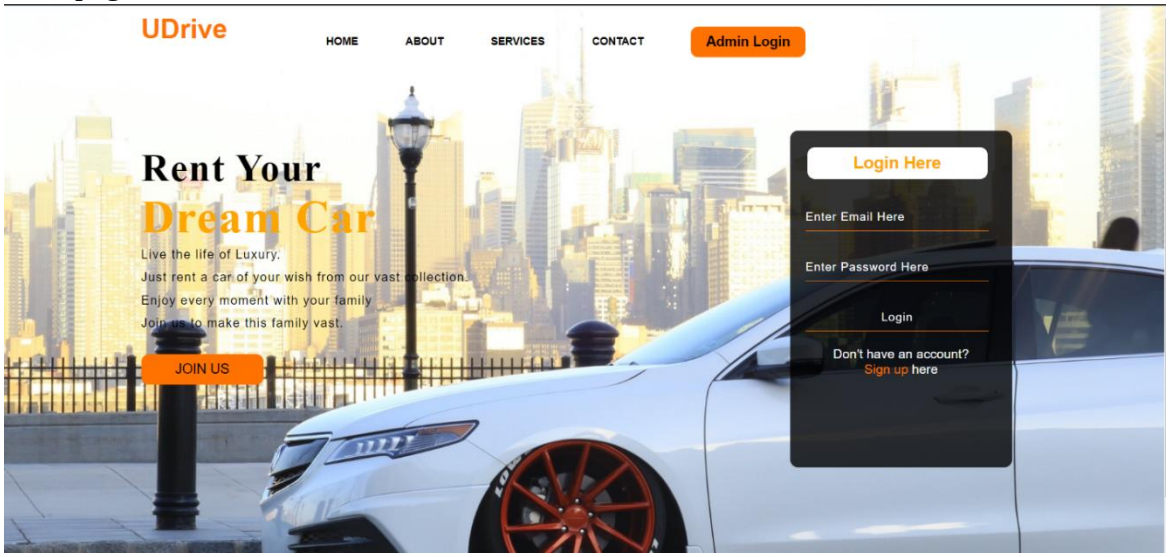

Fig (3.1): Homepage

Here, by implementing this Homepage we can accomplish this user story.

As a registered user, I want to log in with my email and password securely. I should be able to reset my password if I forget it, receiving an email with instructions to do so.

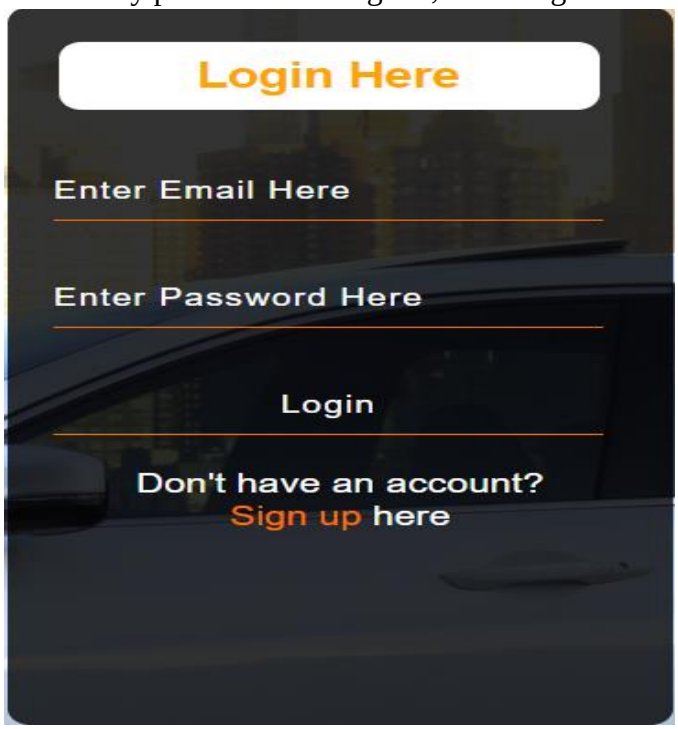

Fig (3.2): User Login

Here, by implementing this User Login we can accomplish this user story.

As a new user, I want to create an account using my personal details, providing my name, email address, and a secure password. After entering valid information, I should receive a verification email to confirm my registration.

Fig (3.3): Registration

Here, by implementing this Registration we can accomplish this user story.

Upon logging in as a User, from the search results, I want to view detailed information about each car, including its specifications, images, and rental cost. I should be able to select a car by clicking on it and confirming my choice.

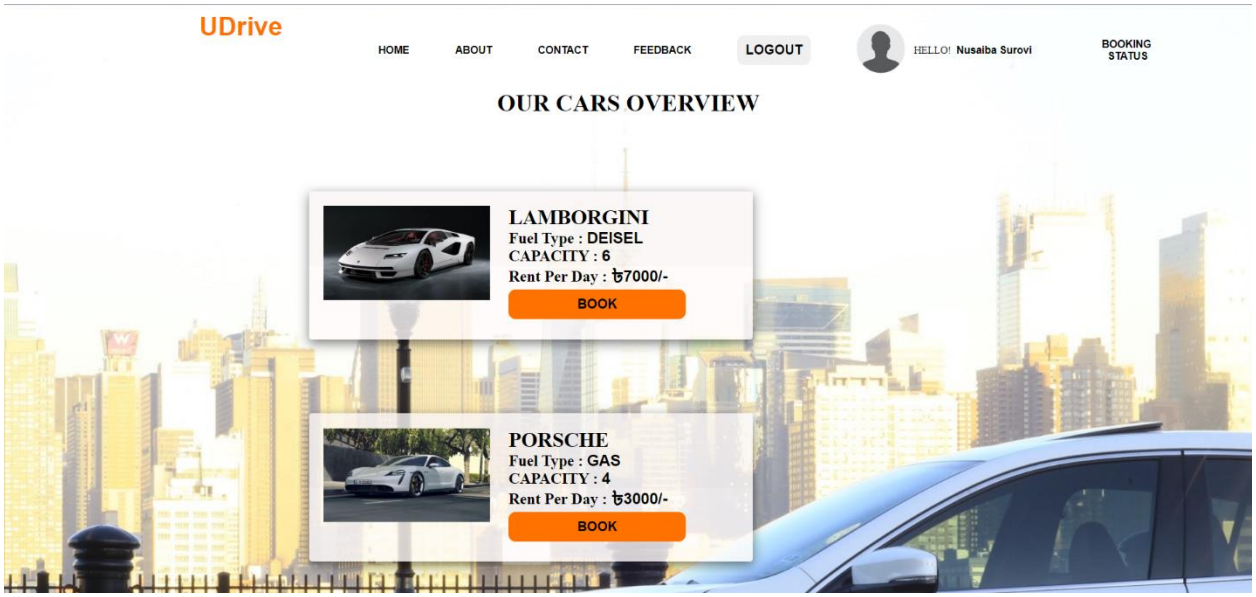

Fig (3.4): Car Selection

Here, by implementing this Car Selection we can accomplish this user story.

After selecting a car, I want to specify the place, phone number, rental duration, pickup date, and return date. The system should calculate the total cost based on my selections and display it for confirmation. Once I confirm the rental, I want to receive a booking confirmation that includes all relevant details such as car information, rental dates, and total cost.

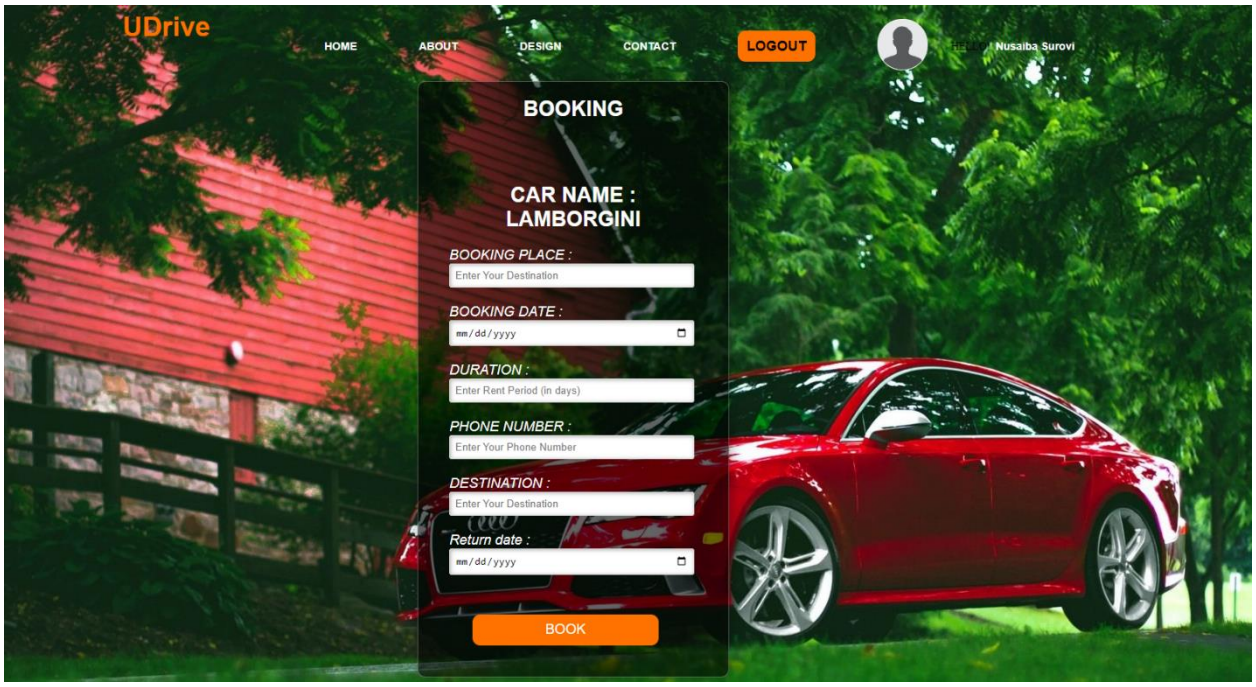

Fig (3.5): Car Booking

Here, by implementing this Car Booking we can accomplish this user story.

Before confirming the rental, I should be able to securely enter my payment details, including credit card information. The system should process the payment and provide me with a receipt that includes a breakdown of the charges.

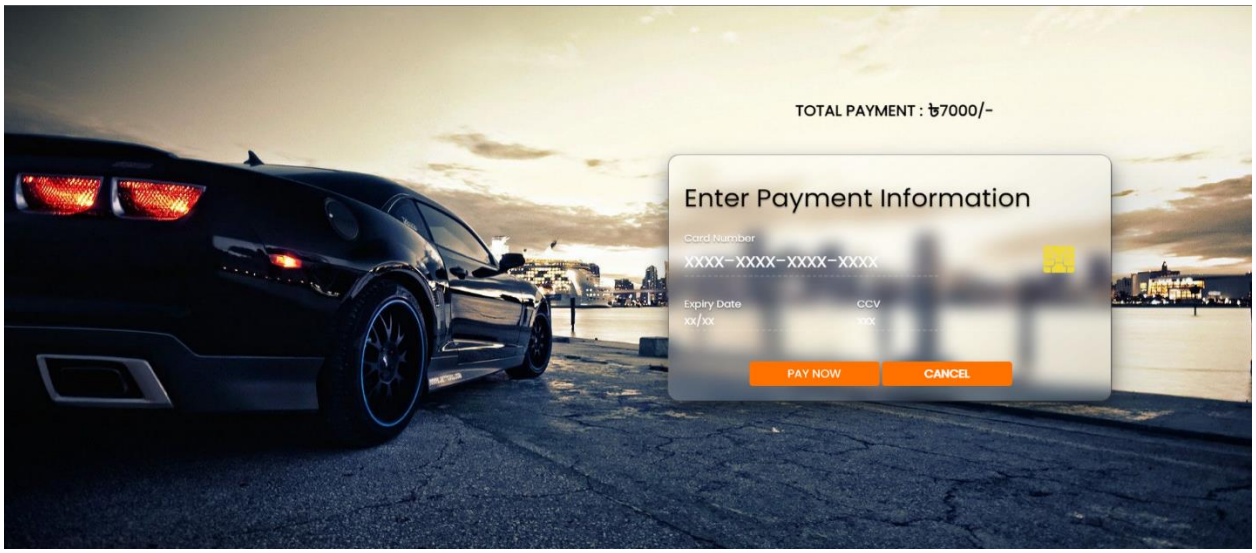

Fig (3.6): Payment

Here, by implementing this Payment we can accomplish this user story.

As a registered user, the system should process the payment and provide me with a receipt that includes a breakdown of the charges.

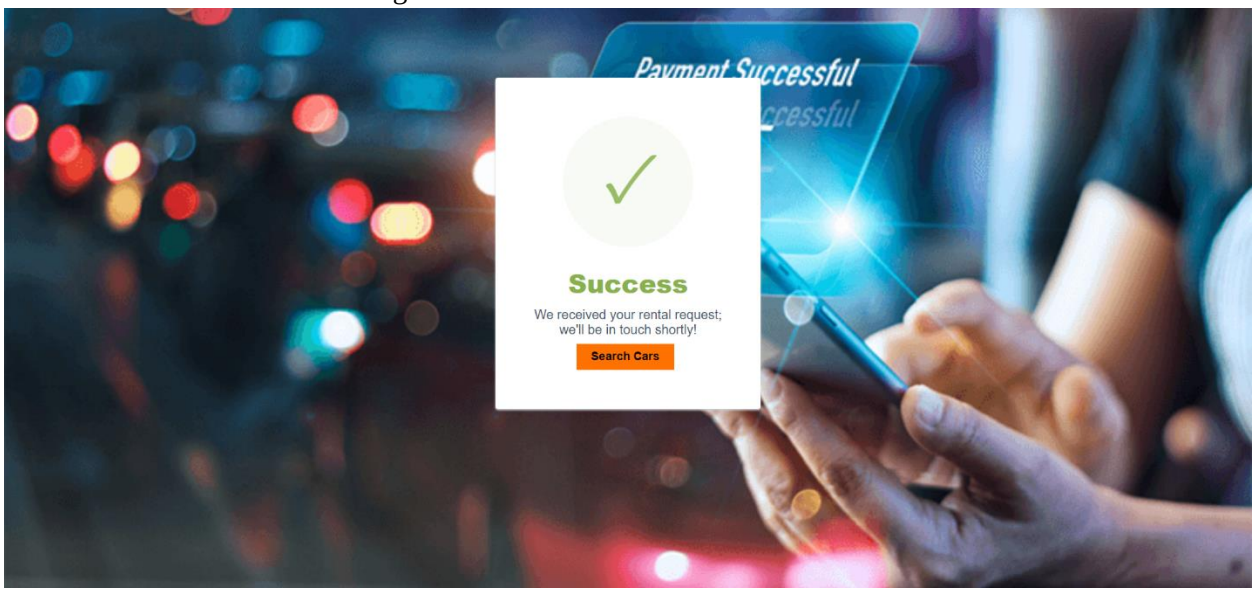

Fig (3.7): Payment Successful

Here, by implementing this Payment Successfully we can accomplish this user story.

As a registered user, I want to have a user profile where I can view my booking history, and current reservations, and update my personal information if needed.

Fig (3.8): Booking Status

Here, by implementing this Booking status we can accomplish this user story.

As a registered user, I want to easily find information about the services offered by the website and be able to contact the website administrators if needed.

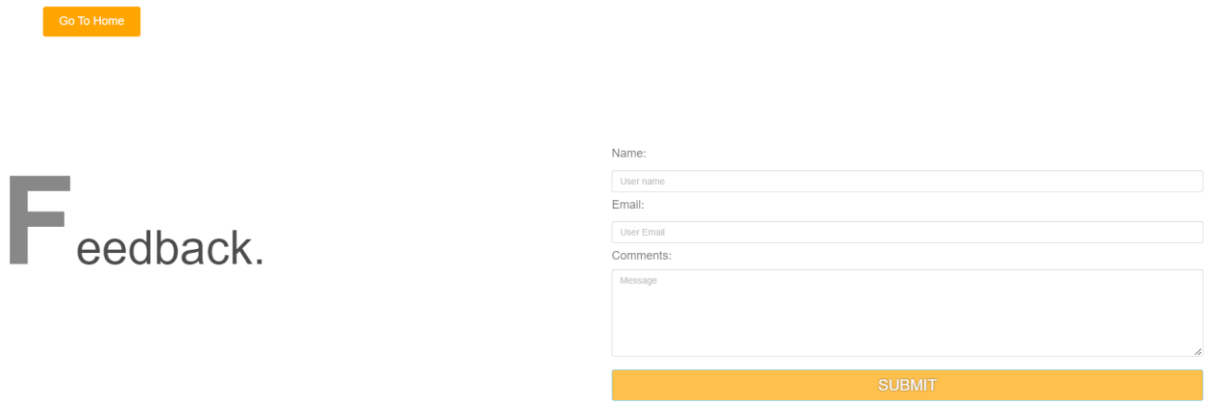

Fig (3.9): Feedback

Here, by implementing this feedback we can accomplish this user story.

Upon accessing the website, I want to be able to locate a clearly labeled section titled "About Us" in the main navigation menu. Clicking on the "About Us" link should take me to a page that provides an overview of the various services offered by the car rental platform.

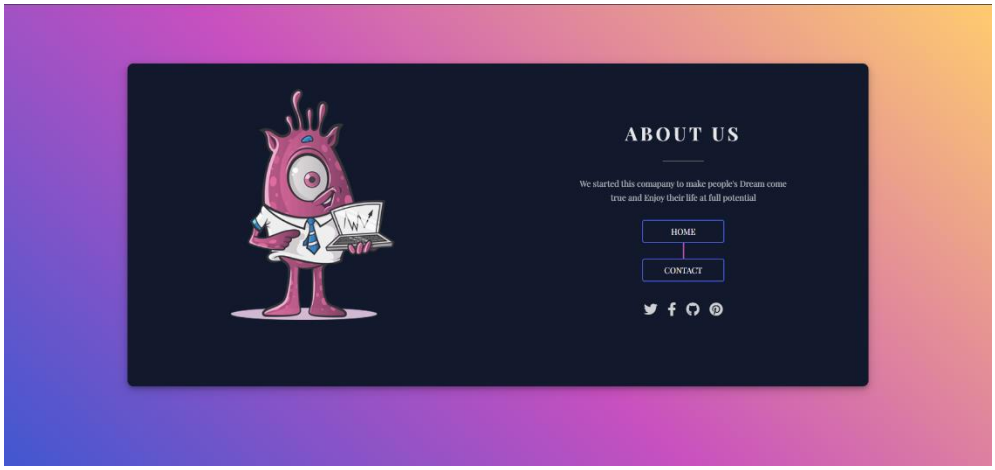

Fig (3.10): About us

Here, by implementing this About Us we can accomplish this user story.

As a registered user, I want to easily find a "Contact Us" link or button prominently displayed on the website. Clicking on the "Contact Us" link should lead me to a page where I can access contact information for the website administrators.

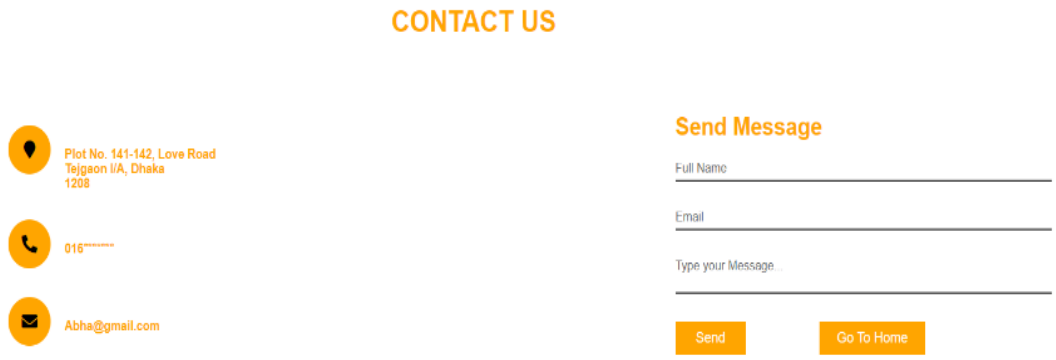

Fig (3.11): Contact Us

Here, by implementing this Contact Us we can accomplish this user story.

As an Admin, the login page should have fields for entering a valid username and password. Upon successful login, the admin should be directed to the admin dashboard. In case of incorrect login credentials, an error message should be displayed.

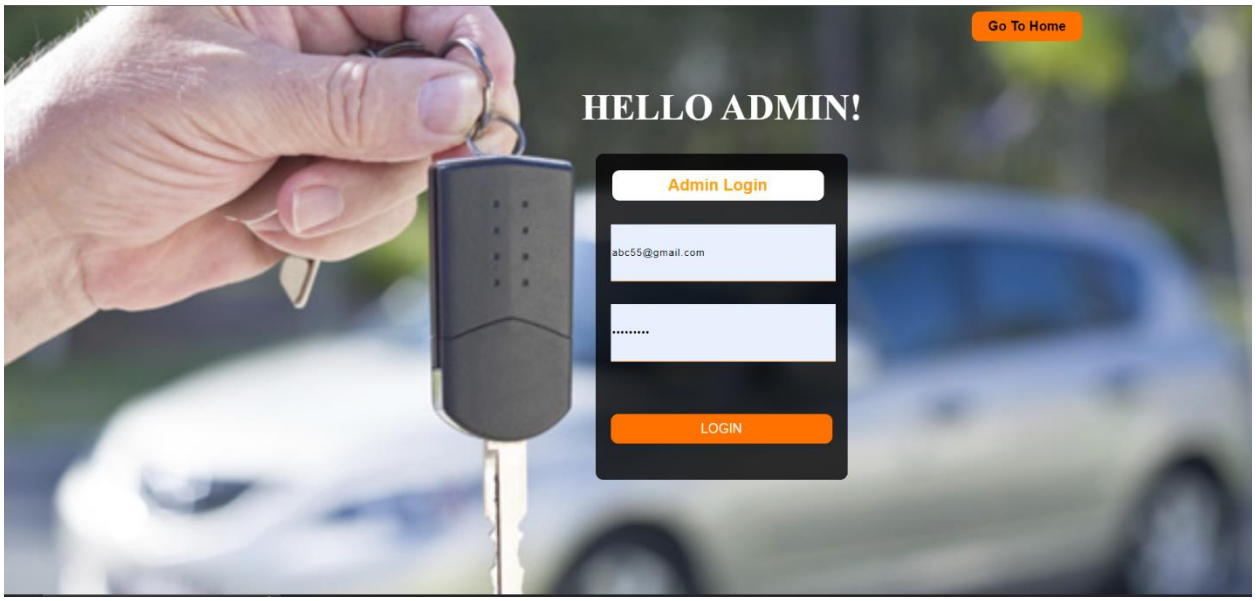

Fig (3.12): Admin Login

Here, by implementing this Admin Login we can accomplish this user story.

As an admin, I want to be able to add new cars to the rental inventory, ensuring that customers have access to an up-to-date selection. I can access the admin dashboard and navigate to the "Add Car" section. I can access the admin dashboard and go to the "Manage Cars" section. I am presented with a list of cars currently in the inventory, including their details and availability status. Next to each car, there is a "Delete" button or option.

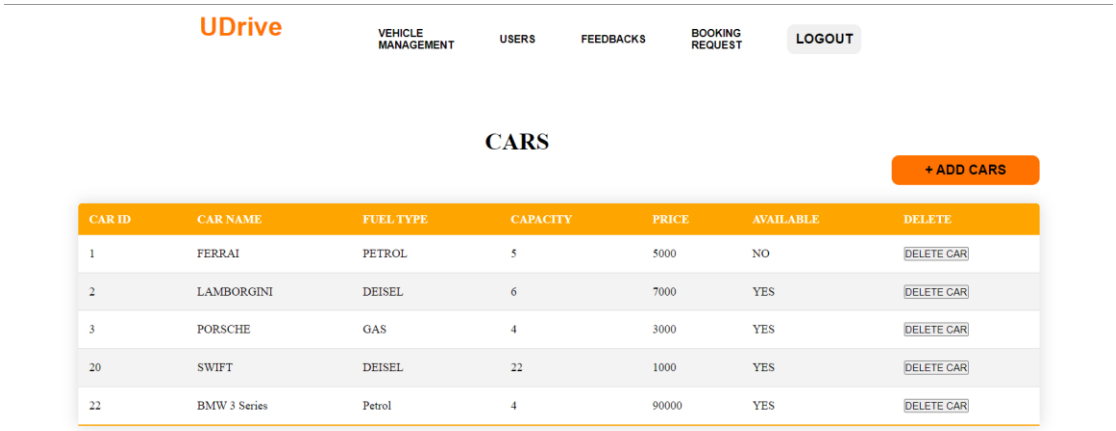

Fig (3.13): Vehicle management

Here, by implementing this Vehicle management we can accomplish this user story.

As an admin, I need the ability to access user information for better management and support. In the admin dashboard, there should be a section labeled "User Management" or similar. When I navigate to this section, I can see a list of registered users along with their basic details, such as username, email, and contact information.

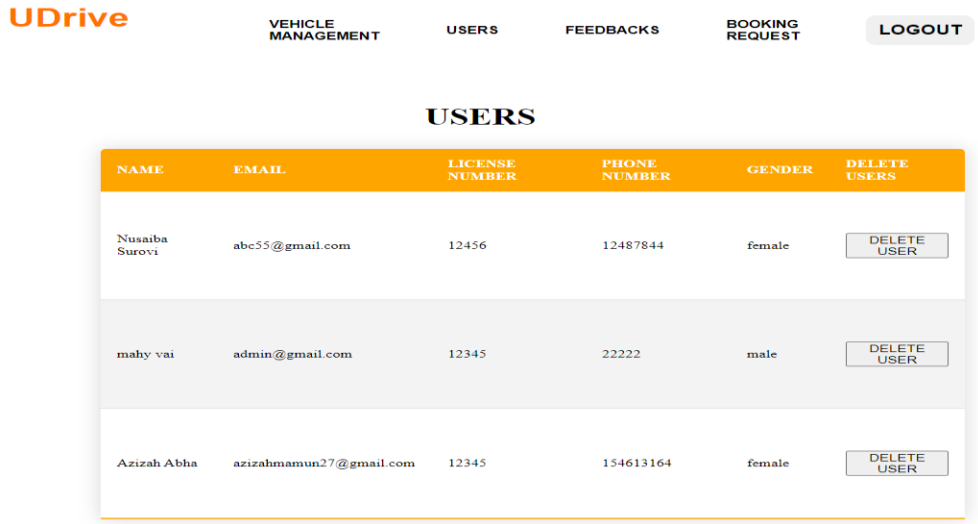

Fig (3.14): Users

Here, by implementing these Users we can accomplish this user story.

As an admin, I want to access and review the feedback provided by users to gain insights into their experiences and identify areas for improvement. In the admin dashboard, there should be a dedicated "Feedback" or "User Feedback" section. When I navigate to this section, I can see a list of recent feedback entries from users. Each feedback entry should display the user's name, date of submission, and a brief summary of the feedback content.

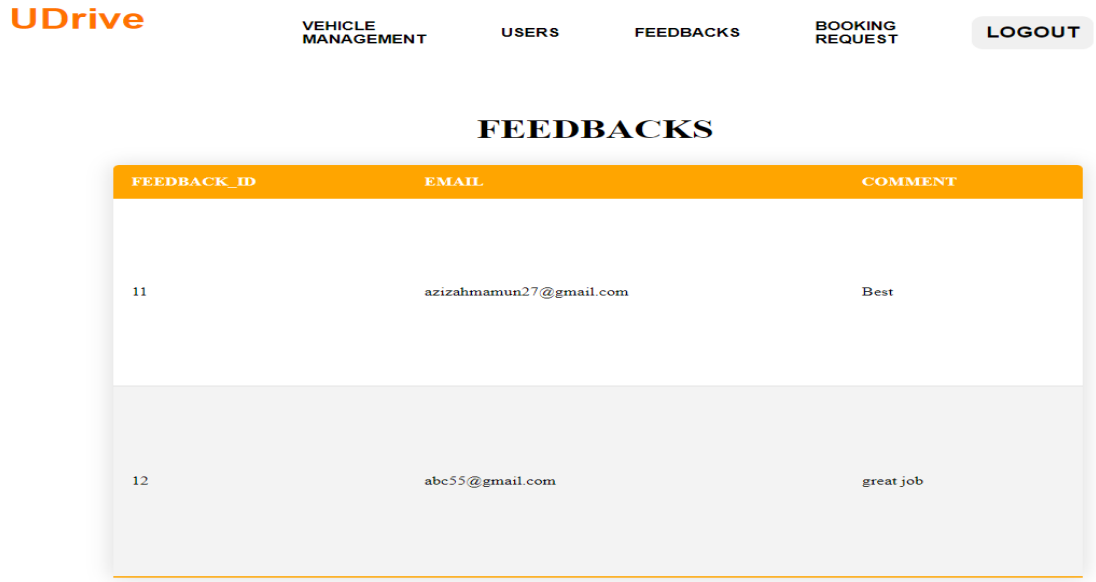

Fig (3.15): Feedbacks

Here, by implementing this Feedback we can accomplish this user story.

As an admin, I need the ability to access and manage booking requests made by users. In the admin dashboard, there should be a dedicated "Booking Requests" section. When I navigate to this section, I can view a list of pending booking requests along with relevant details, such as user information, requested vehicle, rental dates, and status. Each booking request should have options to approve, reject, or view more details.

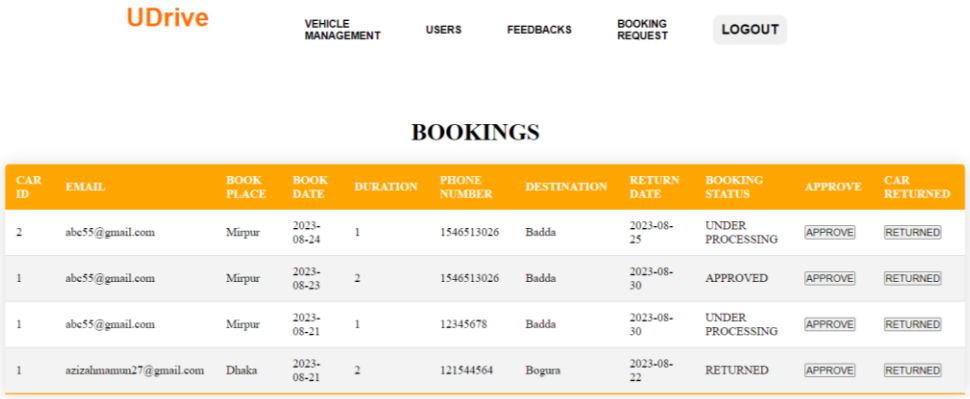

Fig (3.16): Booking Requests

Here, by implementing these Booking Requests we can accomplish this user story.

4. Function Point:

Function Point Ranking According to Complexity

Measurement Parameter	Low	Average	High
1. Number of External Inputs (EI)	7	10	15
2. Number of External Outputs (EO)	5	7	10
3. Number of External Inquiries (EQ)	3	4	6
4. Number of Internal Files (ILF)	4	5	7
5. Number of External Interfaces (EIF)	3	4	6

Table 1.1: Weights of 5-FP Attribute

4.1 Calculate the Unadjusted Function Point (UFP)

Estimate the complexity values for each Functional Point Component:

1. External Inputs (EI):

User Input:

- **User Registration: (Low)**
- **User Login: (Low)**
- **Car Listing: (Average)**
- **Car Search: (Average)**
- **Choose your car: (Low)**
- **Search Car: (Average)**
- **View Car Details: (High)**
- **Car Availability Check: (Low)**
- **Rent Car: (Average)**
- **Add Car Details: (Low)**
- **Update User Profile: (High)**
- **Update Car Details: (Average)**
- **Return Car: (Average)**

Total Weighting Factor: 7*5 + 10*6 + 15*2 = 125

2. External Outputs (EO):

User Output (EO):

- **Car Listing: (Average)**
- **Car Search: (Average)**
- **Car Details: (Average)**
- **Search Car: (Average)**
- **Rent your Car: (Average)**
- **Car Availability Check: (Average)**

- **User Profile: (High)**
- **Return Car: (Low)**
- **Update User Profile: (High)**
- **Update Car Details: (Average)**

Total Weighting Factor: 5*1 + 7*7 + 10*2 = 74

3. External Inquiries (EQ):

User Inquiry (EQ):

- **Search Car: (Average)**
- **Car Details: (High)**
- **Rent Car: (Average)**
- **Transaction: (Average)**
- **Return Car: (Average)**

Total Weighting Factor: 3*0 + 4*4 + 6*1 = 22

4. Internal Logical Files (ILF):

User Files (ILF):

- **User Profile: (Average)**
- **Search Car: (Average)**
- **Manage User: (Average)**
- **Rented car: (Average)**
- **Transaction: (Average)**

Total Weighting Factor: 4*0 + 5*5 + 7*0 = 25

5. External Interface Files (EIF):

- **Transaction Gateway: (High)**
- **Return Car: (High)**

Total Weighting Factor: 3*0 + 4*0 + 6*2 = 12

Measurement Parameter	Weighting Factor
1. Number of External Inputs (EI)	125
2. Number of External Outputs (EO)	74
3. Number of External Inquiries (EQ)	22
4. Number of Internal Files (ILF)	25
5. Number of External Interfaces (EIF)	12
	Total Count: 258

Table 1.2: Computing weighting factor

Therefore, the **UFP** = 258

4.2 Calculate the Complexity Adjustment Factor (CAF)

General System Characteristics (GSC)	Degree Of Influence (DI)
1. Data Communications	3
2. Distributed Data Processing	2
3. Performance	4
4. Heavily User Configuration	0
5. Transaction Rate	5
6. Online Data Entry	5
7. End-User Efficiency	5
8. Online Update	3

9. Complex Processing	2
10. Reusability	4
11. Installation Ease	0
12. Operational Ease	4
13. Multiple Sites	4
14. Facilitate Change	4
	TDI: 45

Table 2: 14-factors

CAF = [0.65 + (0.01 *** TDI**)] = [0.65 + (0.01 * **45**)] = **1.1**

4.3 Calculate Functional Points (FP)

FP = **UFP * CAF** = **258 * 1.1 = 283.8**

5. Gantt Chart:

Fig: Gantt Chart

6. Diagrams:

6.1 User Case Diagrams:

U Drive Car Rental System:

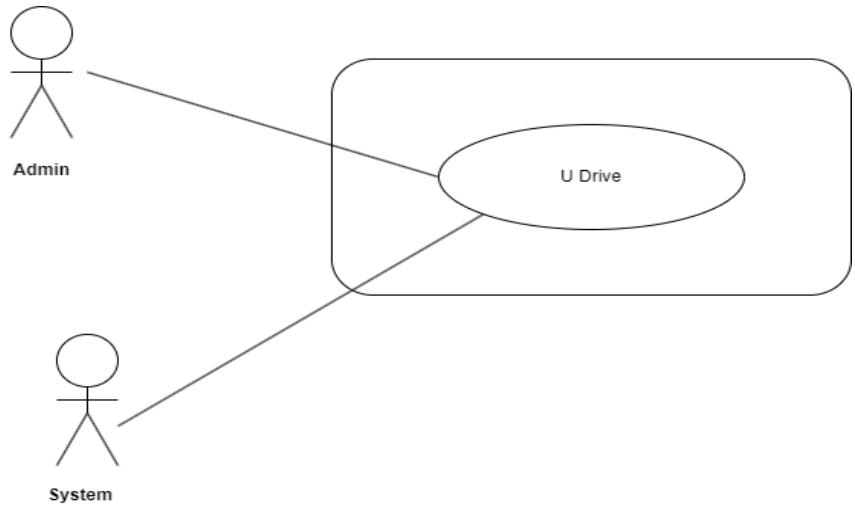

Fig1: Use case - UDrive Car Rental System (ADMIN)

Here is the admin who will play the actor role in this system. The primary actor will play action and will manage the whole system.

Use case Subsystem:

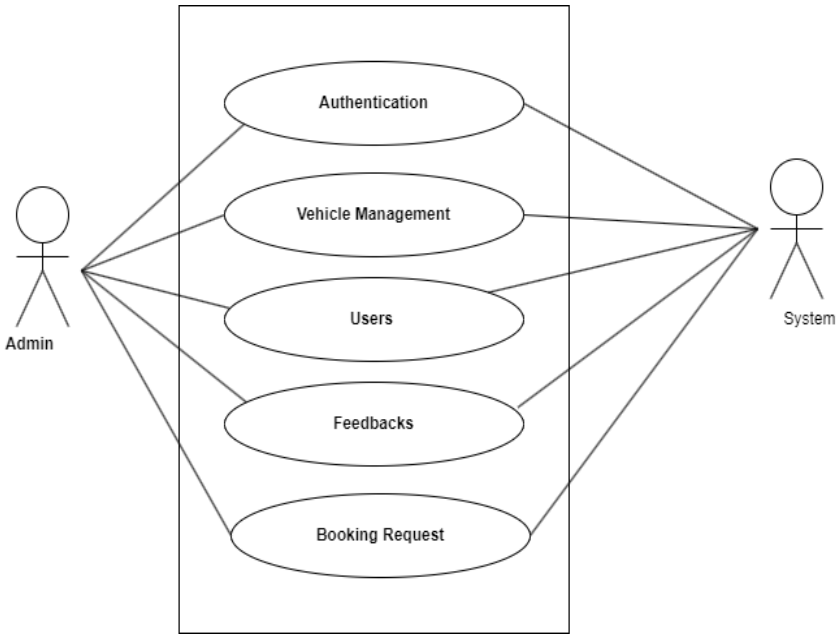

Fig2: Use case – Subsystem (ADMIN)

There are five subsystems in the U Drive Car Rental System. They are:

1. Authentication
2. Vehicle Managements
3. Users
4. Feedbacks
5. Booking Request

Use case Authentication:

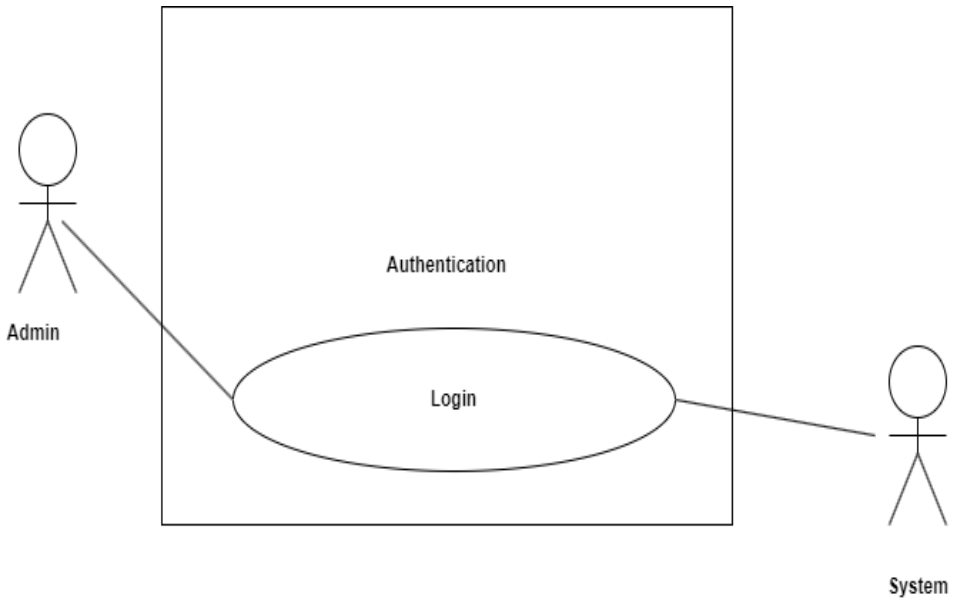

Fig3: Use case – Authentication (ADMIN)

When the admin wants to log in, he needs to enter username and password. If the username and password match then login is succuessful then 'Login succuessfully' message is shown. The system shows an error message if the username or password, or both are wrong and the user can try again to log into the system.

Use case Vehicle Management:

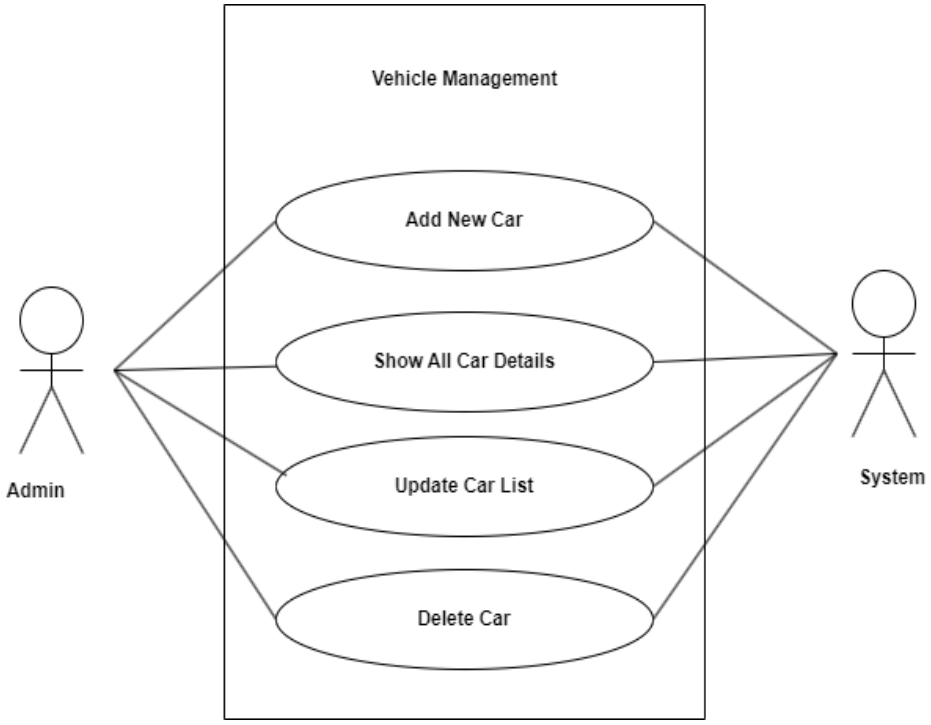

Fig4: Use case - Vehicle Management

Here admin can add a new car, show all car details, update the car list, and delete the car.

Use case - Users:

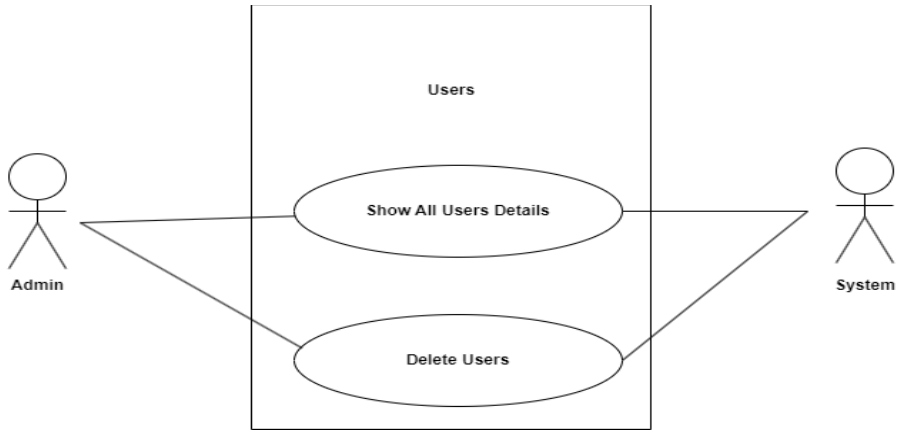

Fig5: Use case – User

Here admin can see all details of users and can delete users.

Use case Users Feedback:

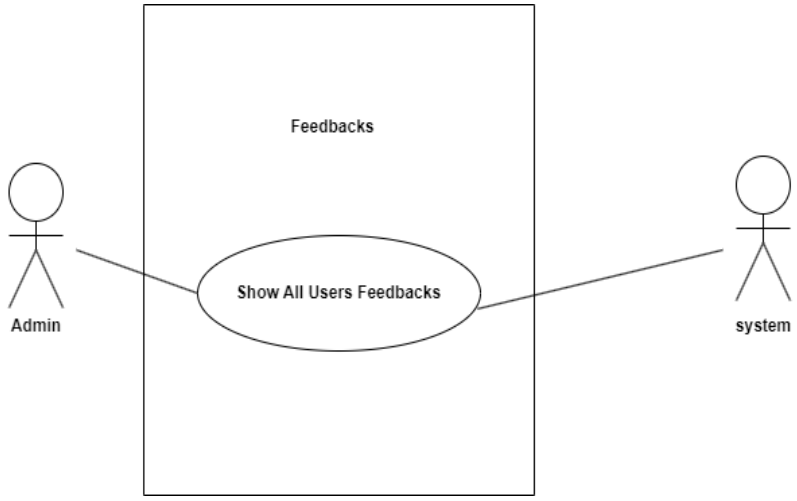

Fig6: Use case- Users Feedback

Here admin can see all users feedbacks.

Use case Booking Request:

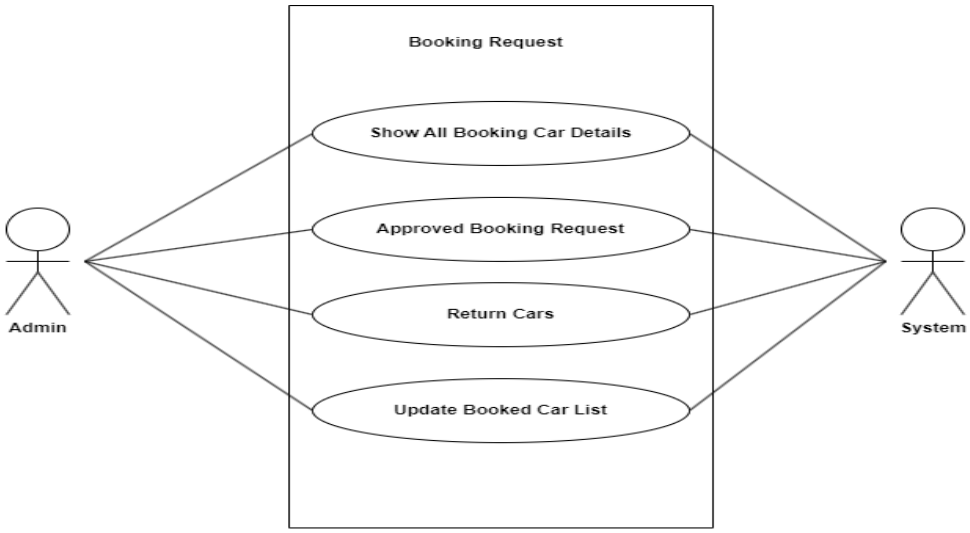

Fig7: Use case - Booking Request

Here admin can show all booking car details ,approve booking request, return cars and updated booked cars.

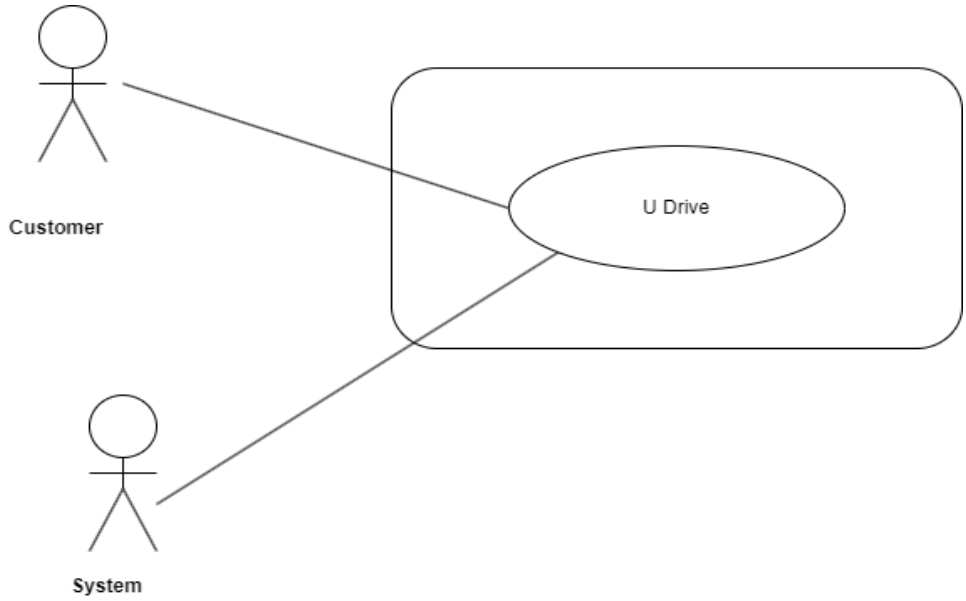

Fig8: Use case- U Drive Car Rental System for (CUSTOMER)

Here is the customer who will play the actor role in this system.

Use case Subsystem:

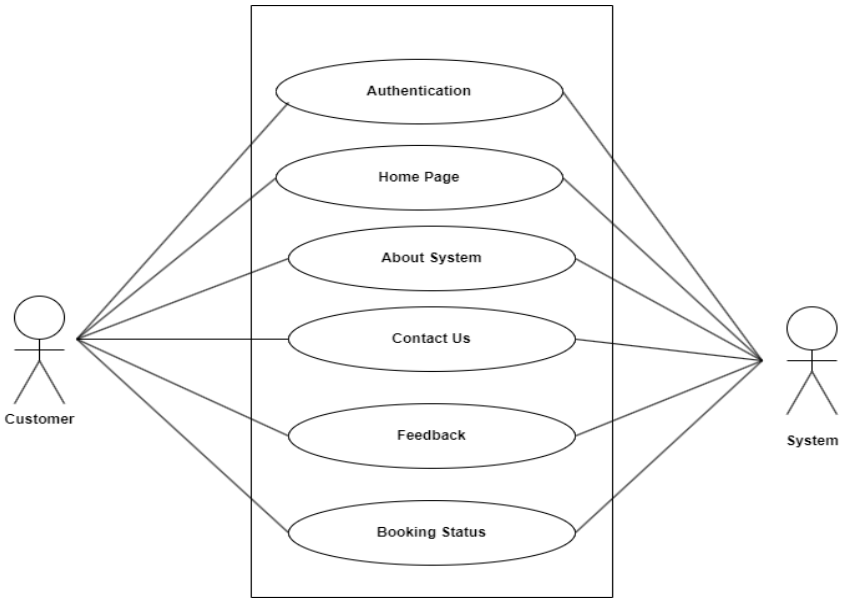

Fig9: Use case—Subsystem (CUSTOMER)

There are five subsystems in the U Drive Car Rental System. They are:

1. Authentication
2. Home Page
3. About System
4. Contact Us
5. Feedback
6. Booking Status

Use case Authentication:

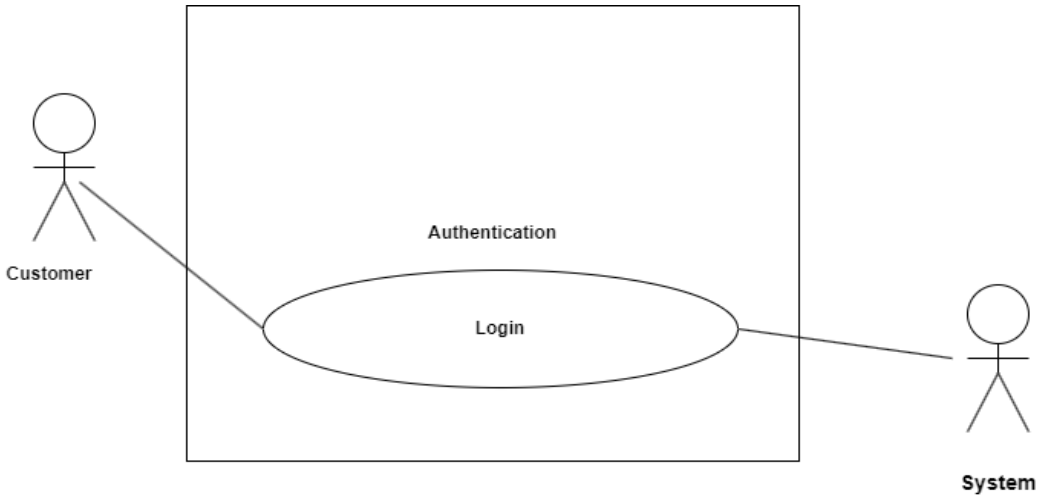

Fig10: Use case Authentication (CUSTOMER):

When the customer wants to log in, he needs to enter username and password. If the username and password match then login is succuessful then 'Login succuessful' message is shown. The system shows an error message if the username or password, or both are wrong and the user can try again to log into the system.

Use case Homepage:

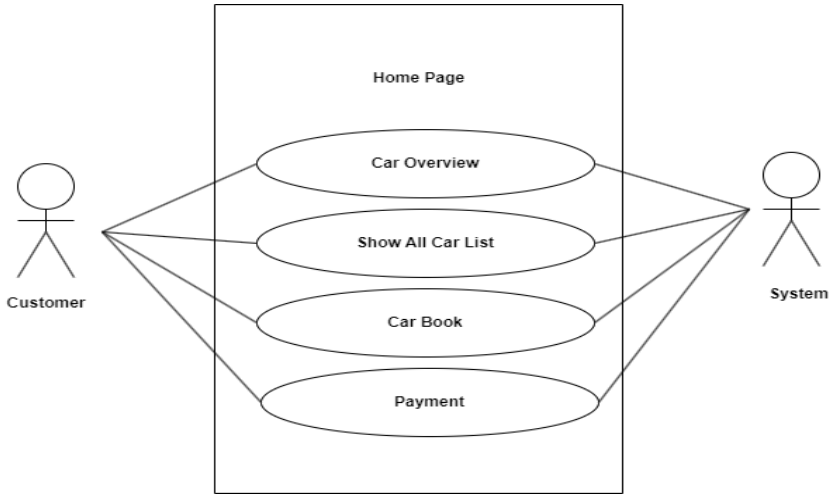

Fig11: Use case – Homepage

Customer can visit homepage .Then he can overview cars, show all car list, can book car and complete payment method.

Use case About Us:

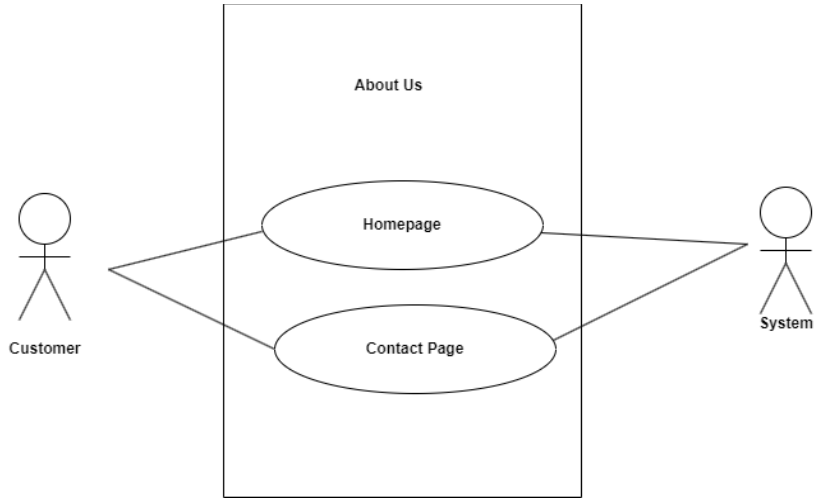

Fig12: Use case-About Us

Customer can visit home page ,can contact page to contact our system and can give feedbacks about our site.

6.2 Swimlane Diagrams:

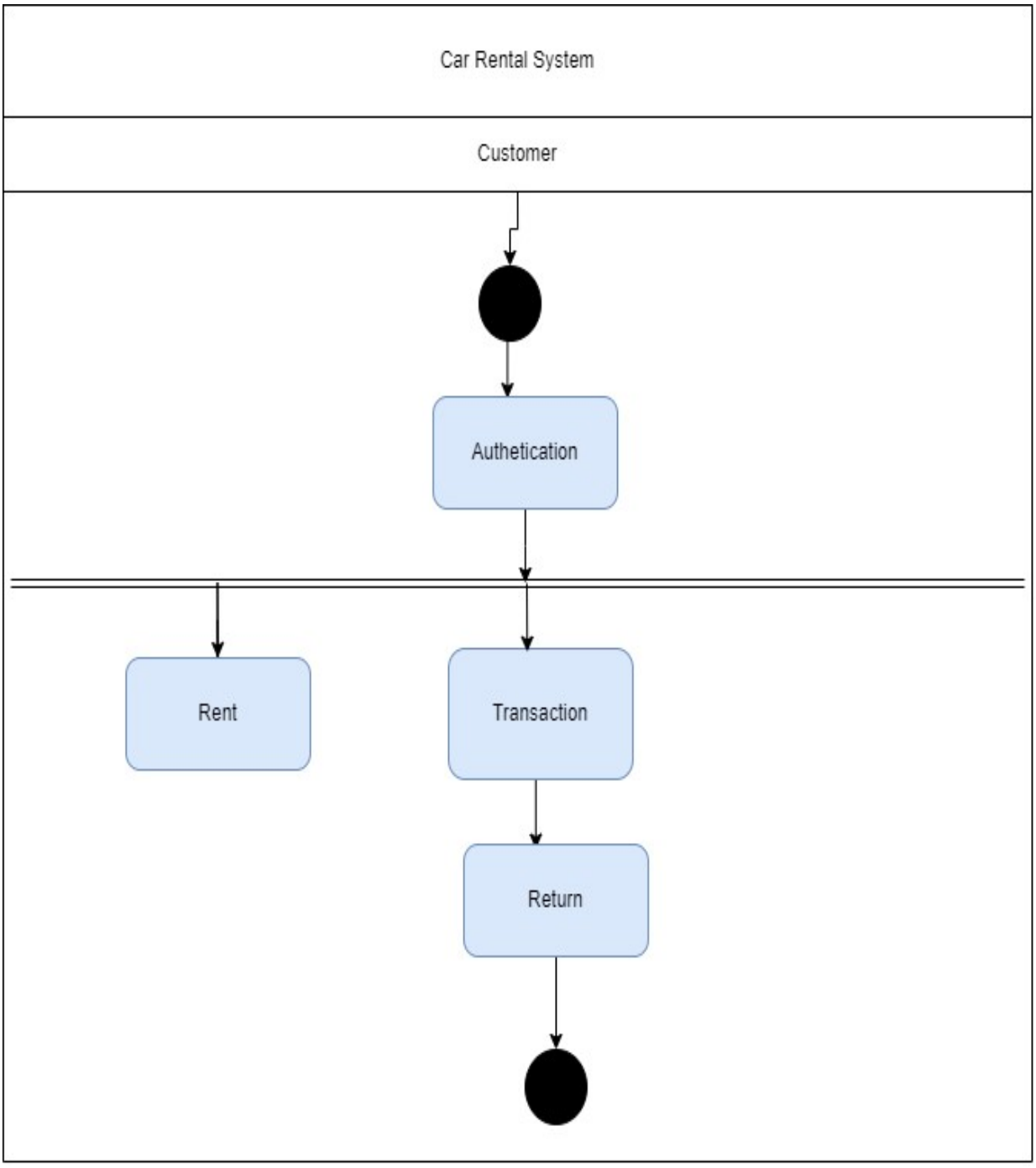

Fig (6.2.1)

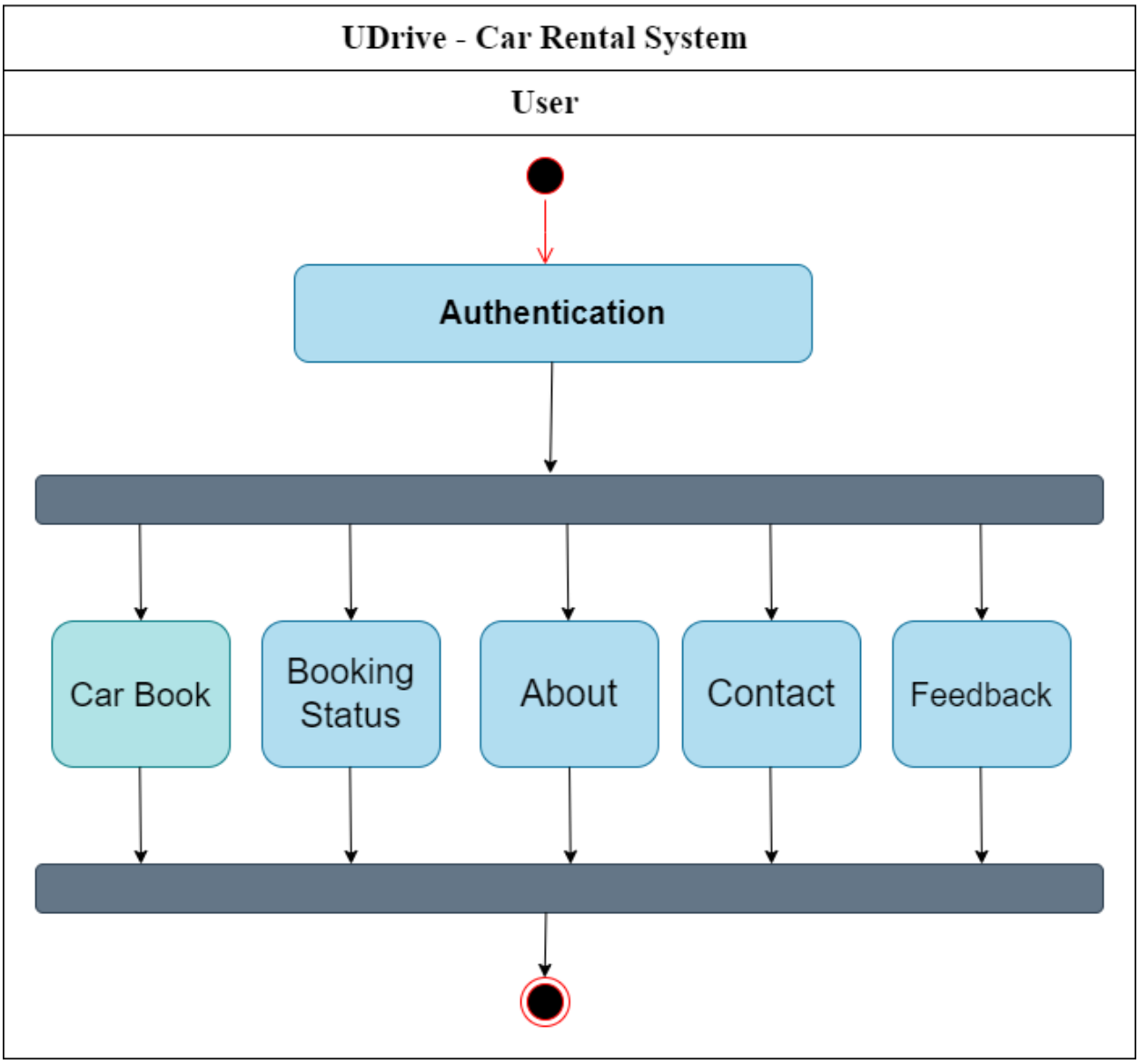
UDrive - Car Rental System
User
Authentication
Car Book
Booking Status
About
Contact
Feedback

Fig (6.2.2)

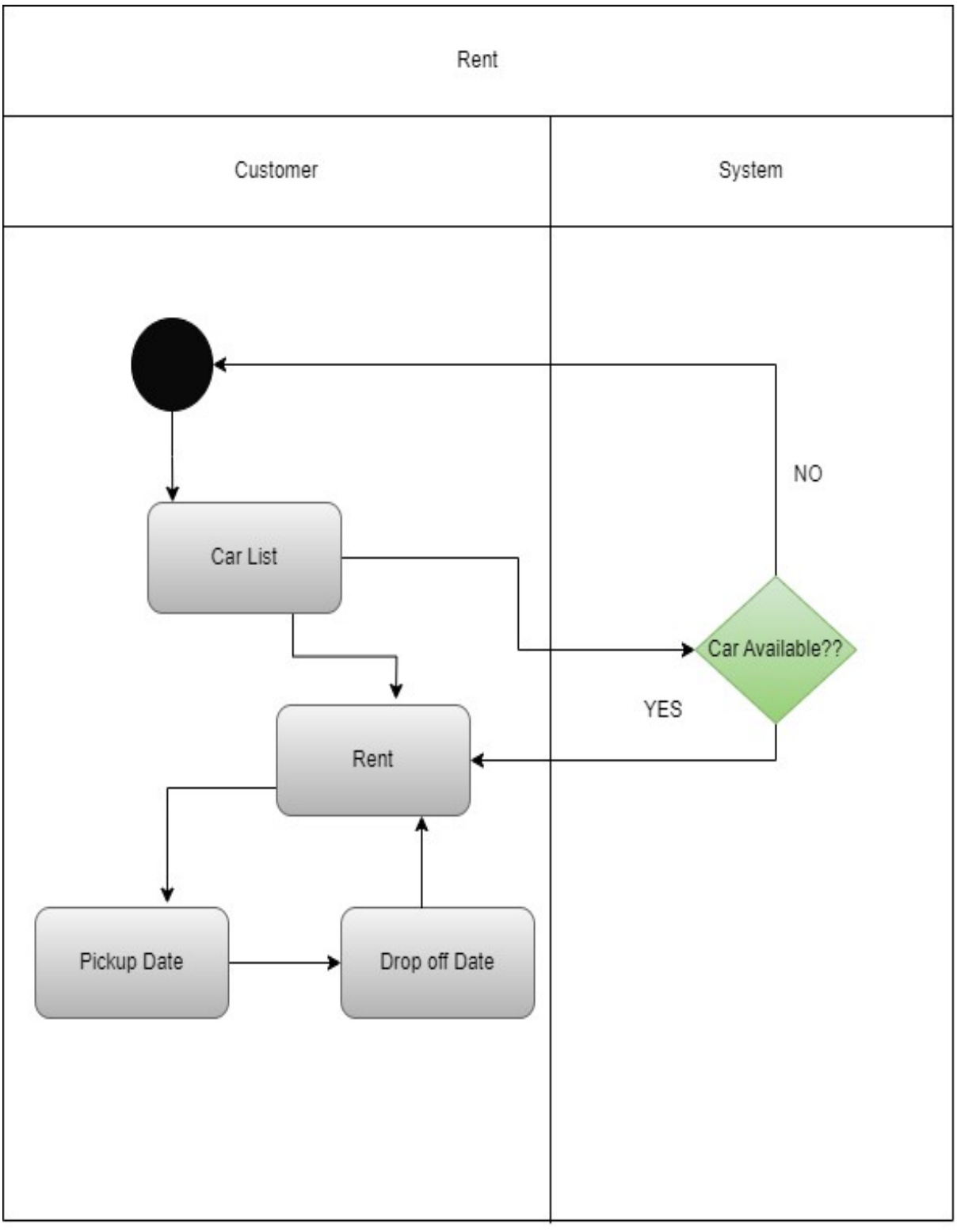
Rent
Customer
System
NO
Car List
Car Available??
YES
Rent
Pickup Date
Drop off Date

Fig (6.2.3)

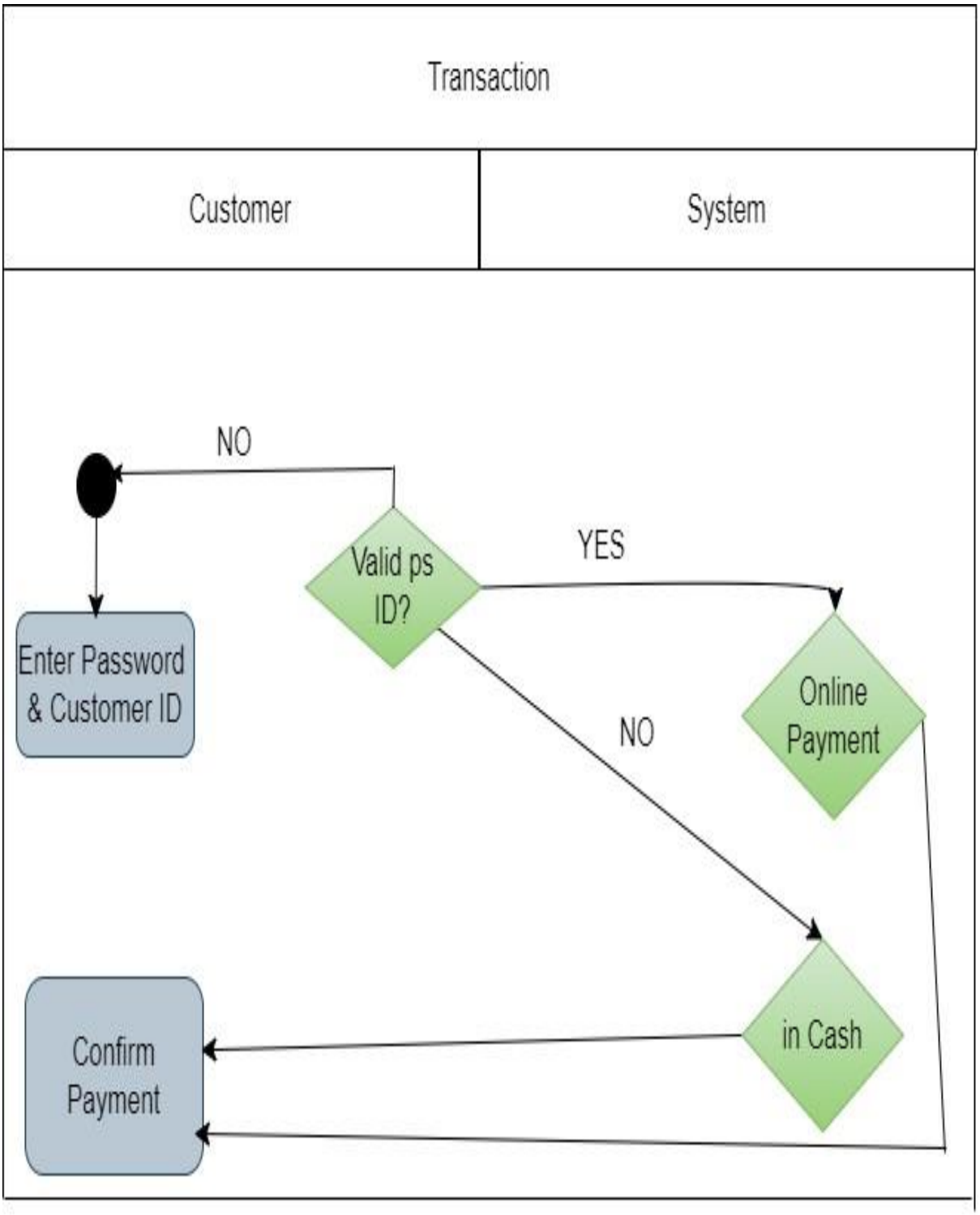
Transaction
Customer
System
NO
Valid ps ID?
YES
Enter Password & Customer ID
Online Payment
NO
in Cash
Confirm Payment

Fig (6.2.4)

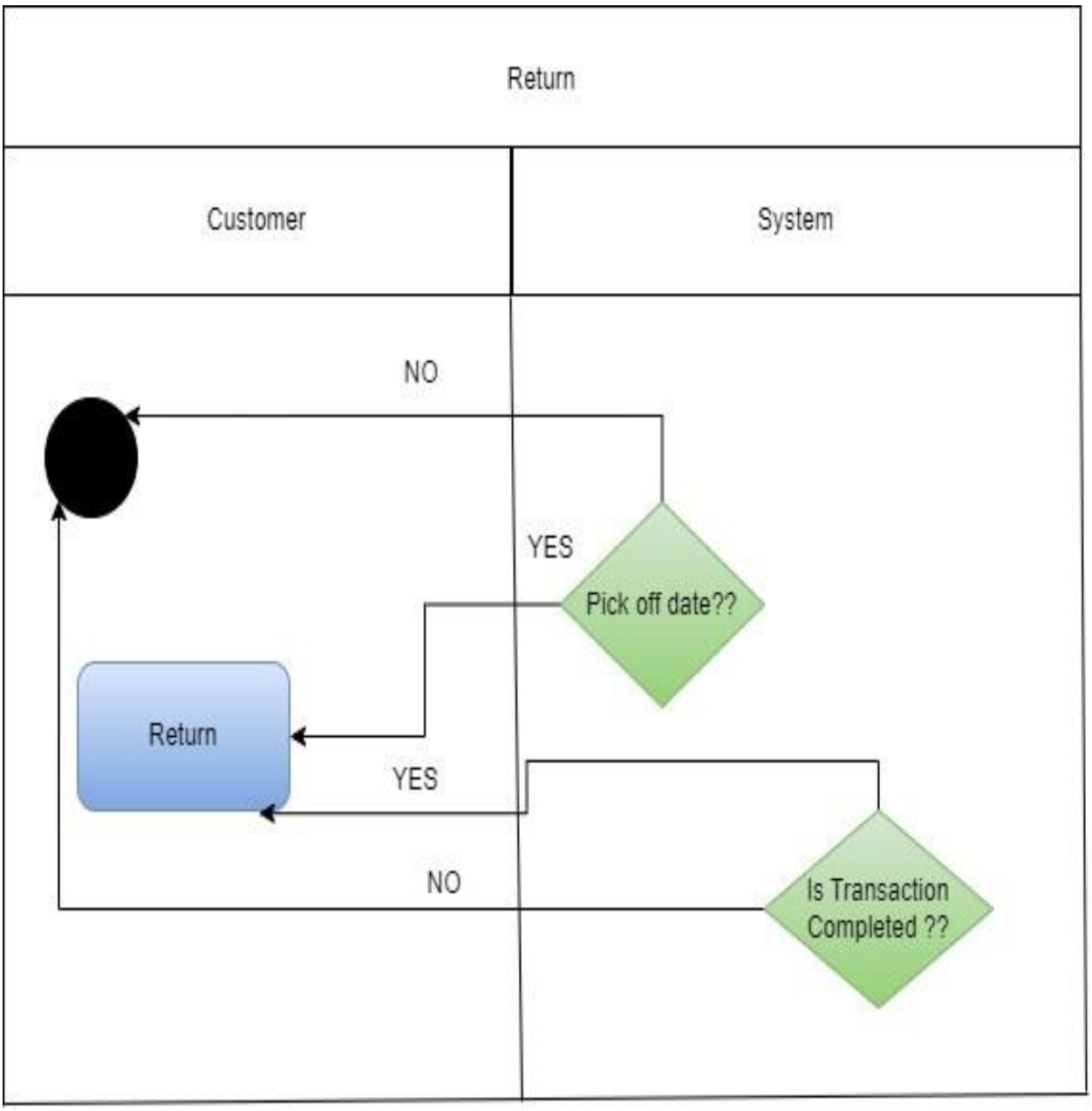
Return
Customer
System
NO
YES
Pick off date??
Return
YES
NO
Is Transaction Completed ??

Fig (6.2.5)

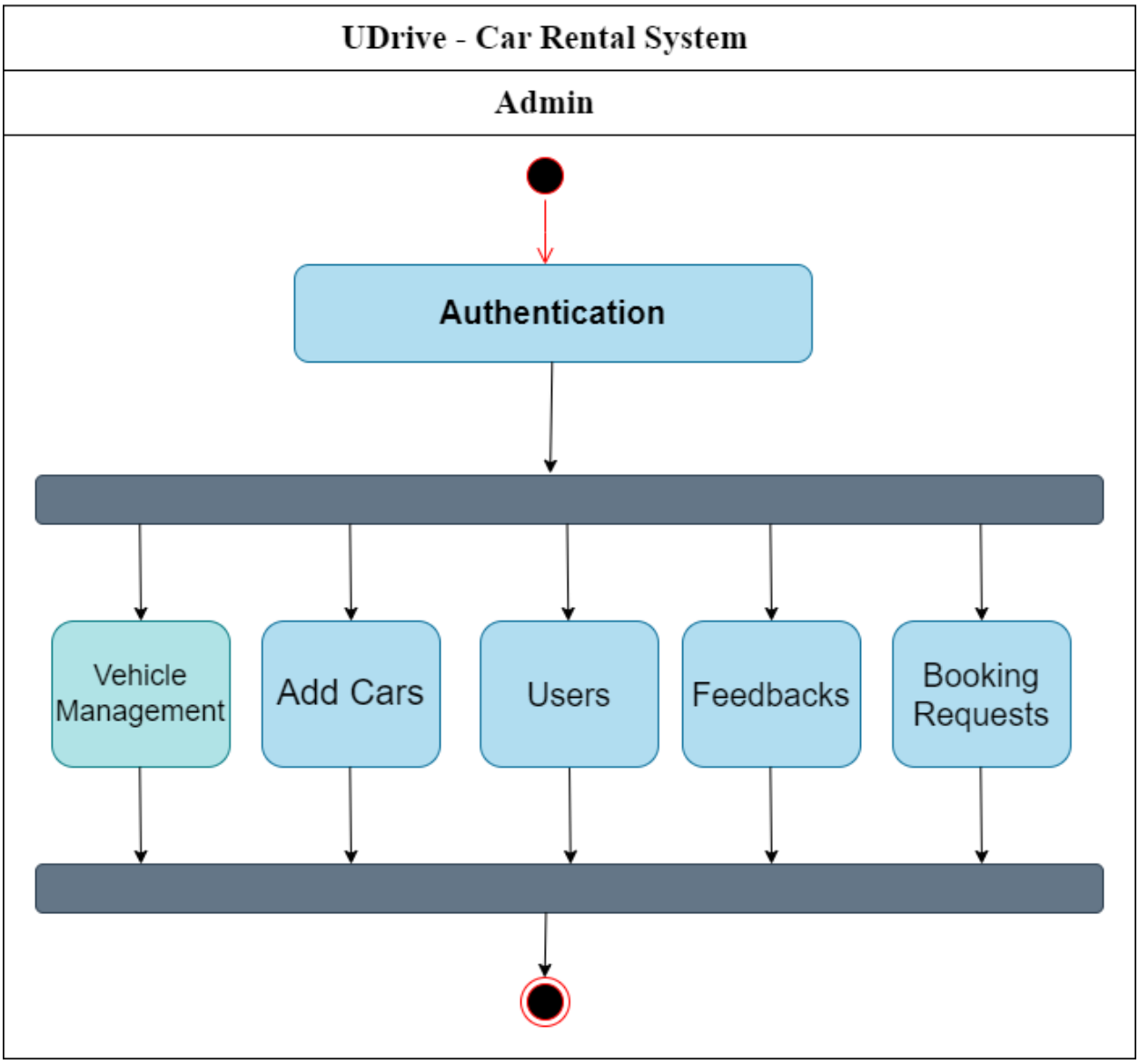
UDrive - Car Rental System
Admin
Authentication
Vehicle Management
Add Cars
Users
Feedbacks
Booking Requests

Fig (6.2.6)

6.3 Class Diagrams:

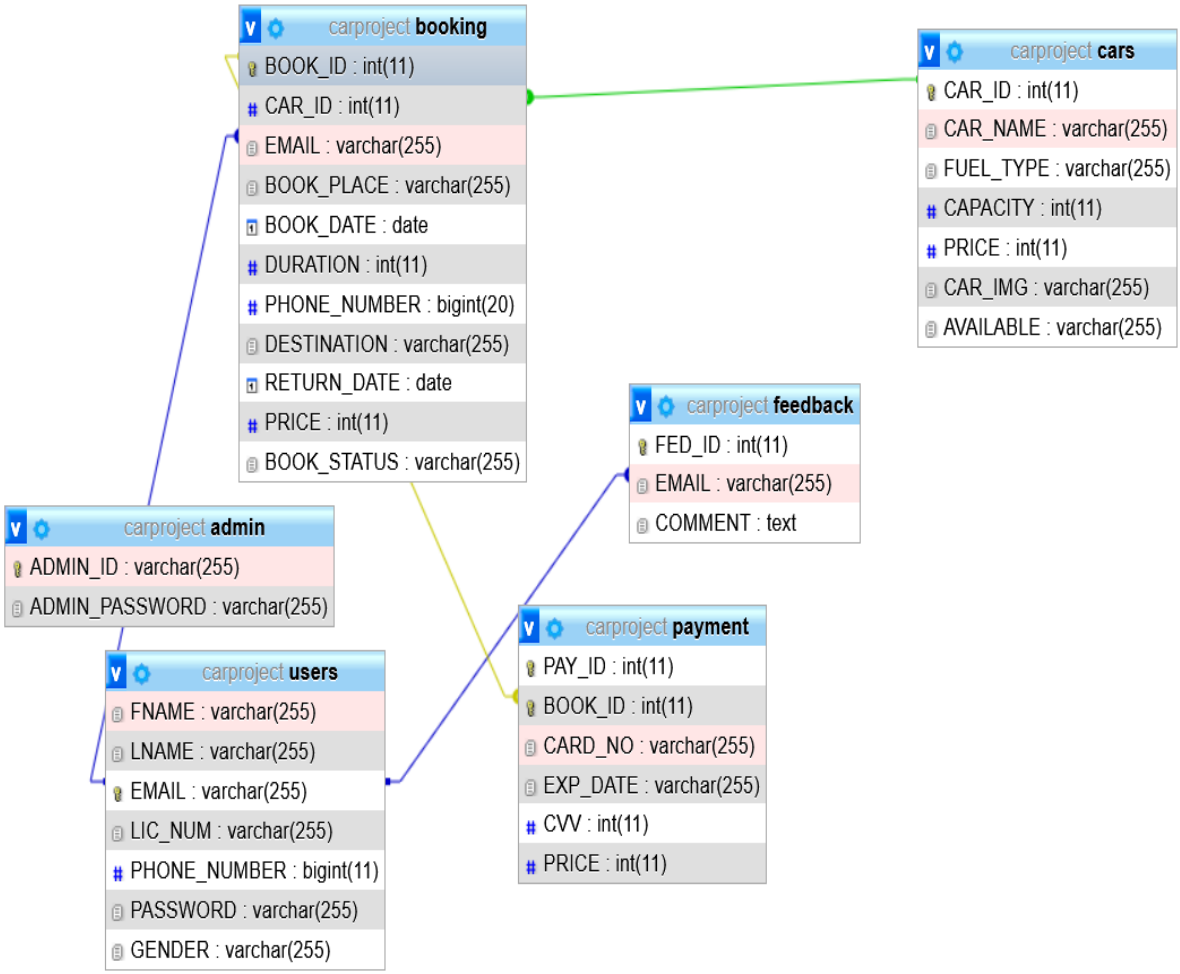

Fig (6.3): UML Design

6.4 State Diagrams:

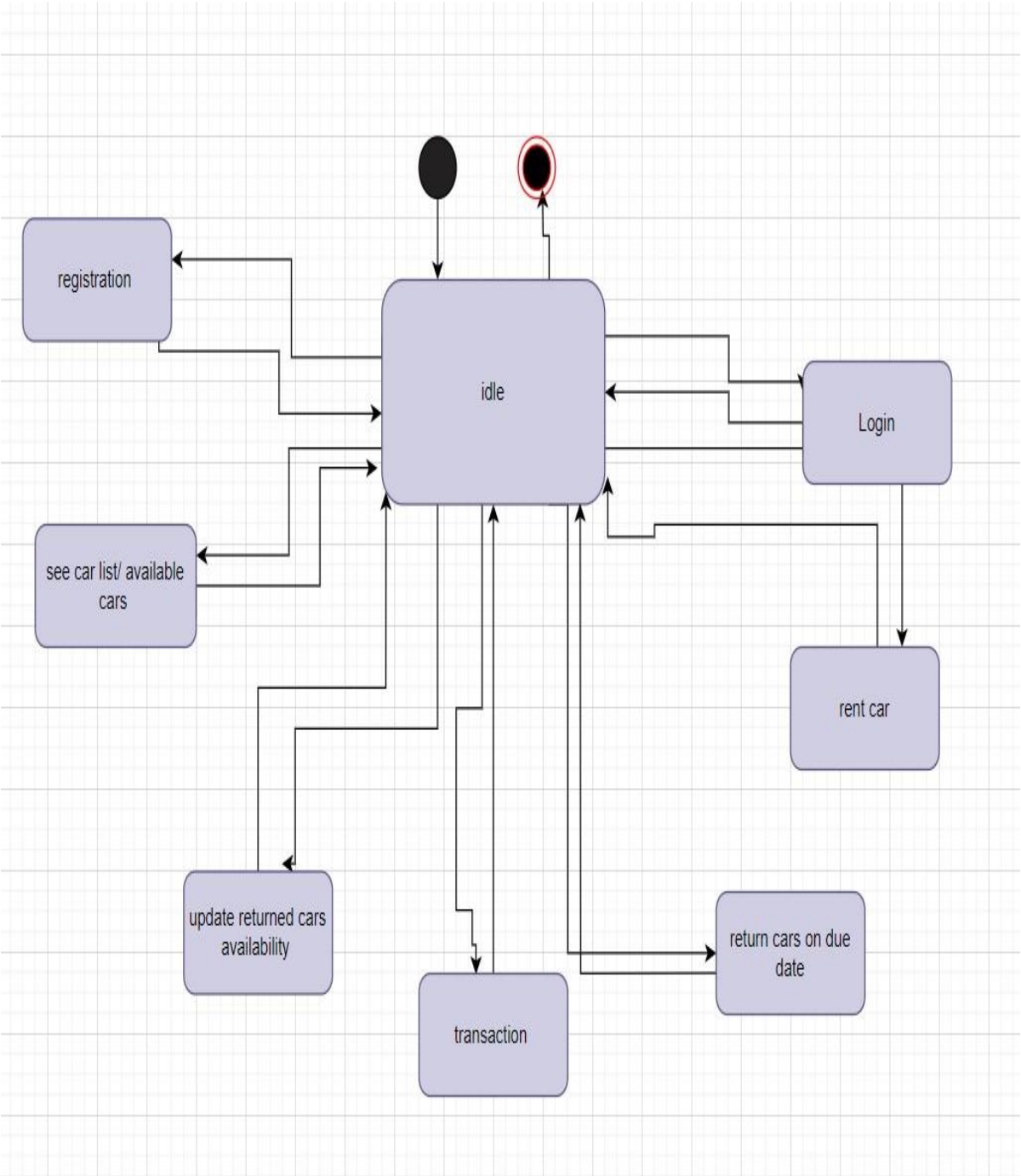

Fig (6.4.1)

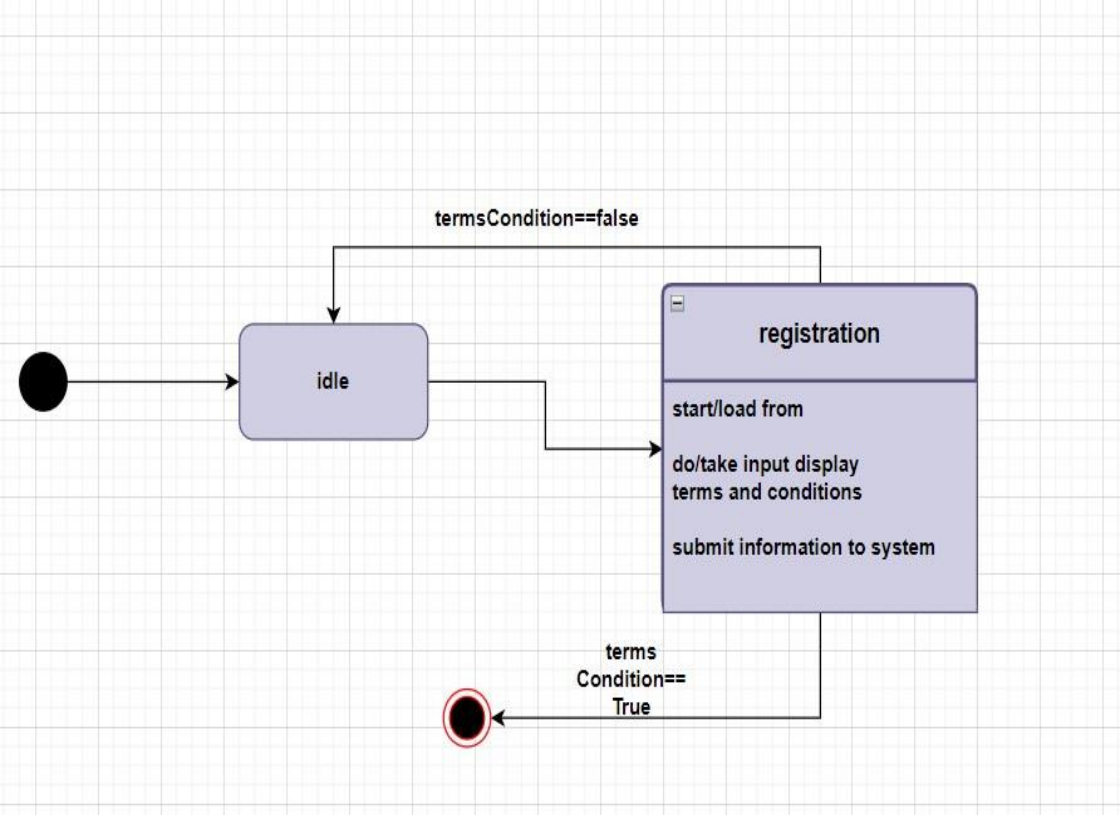

Fig (6.4.2)

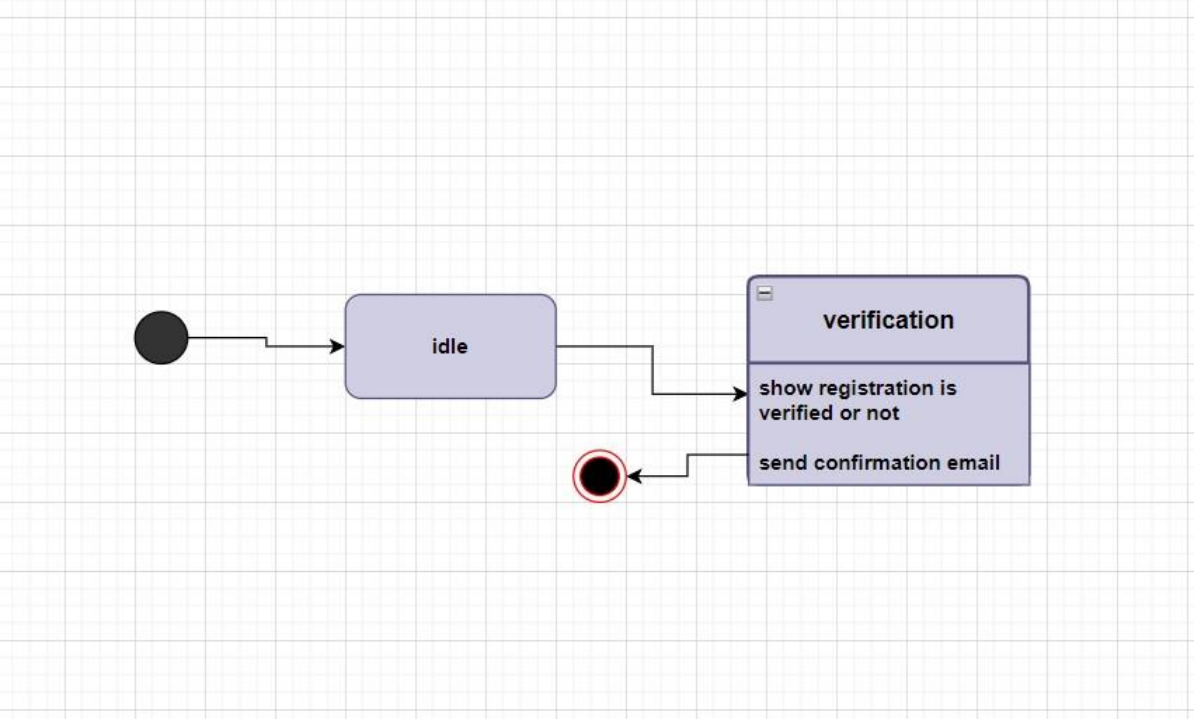

Fig (6.4.3)

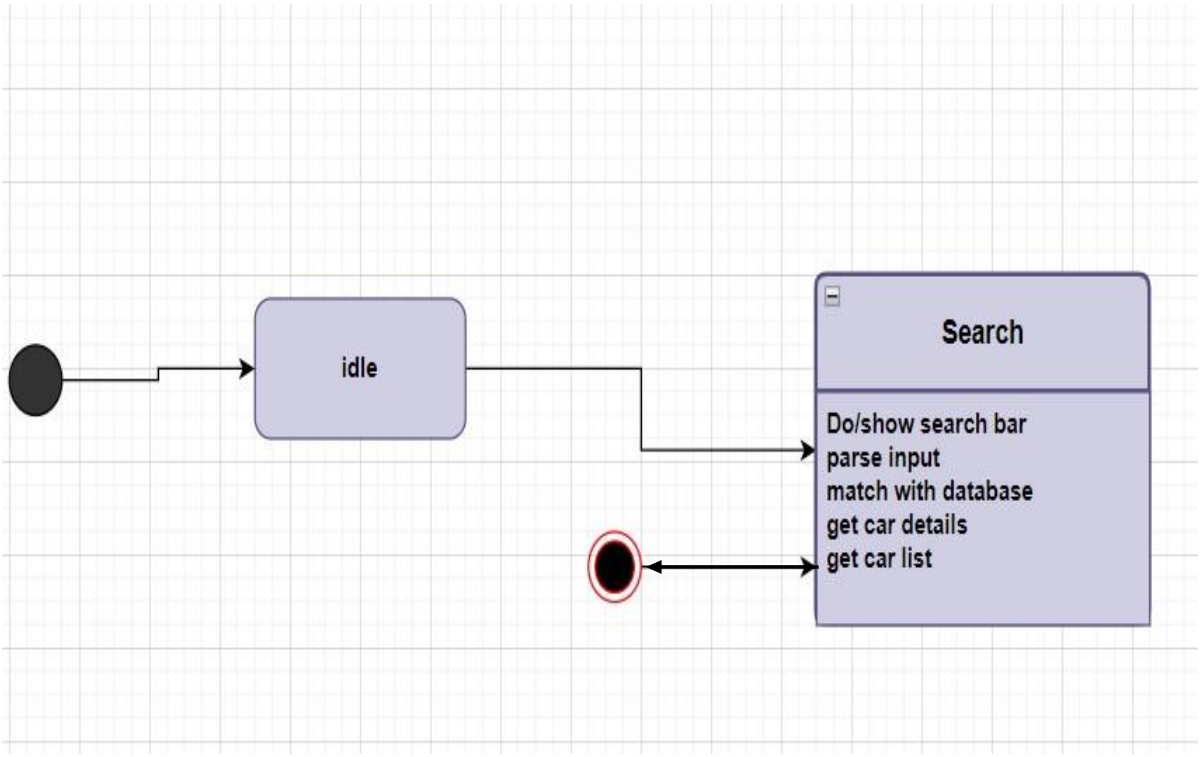

Fig (6.4.4)

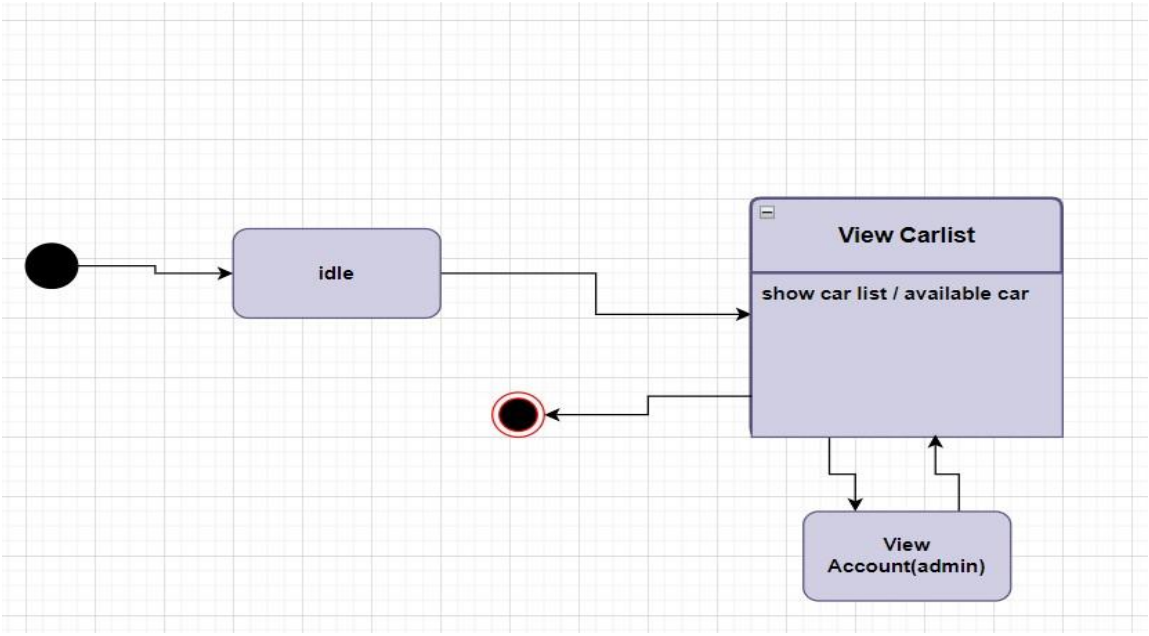

Fig (6.4.5)

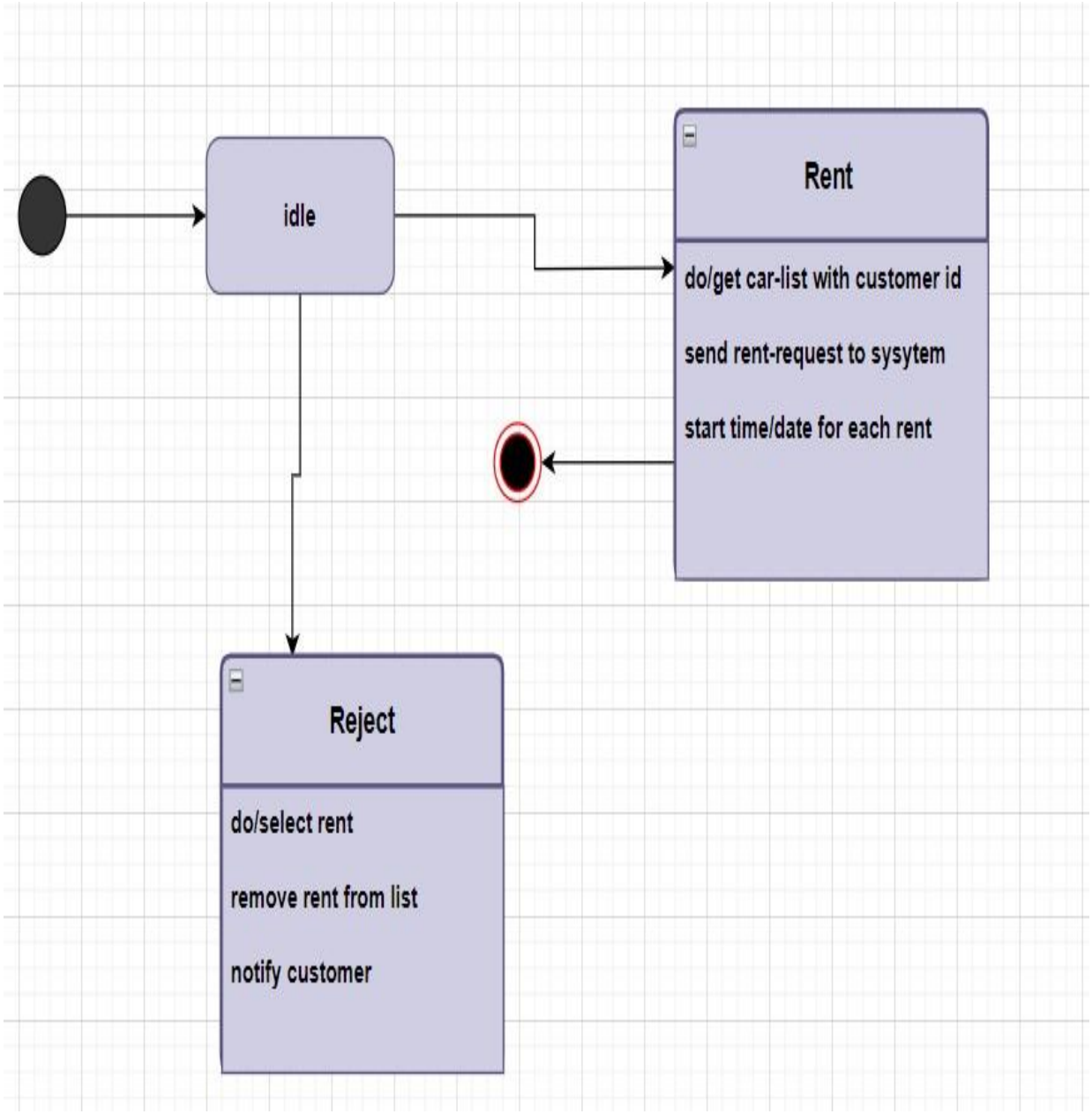
idle
Rent
do/get car-list with customer id
send rent-request to sysytem
start time/date for each rent
Reject
do/select rent
remove rent from list
notify customer

Fig (6.4.6)

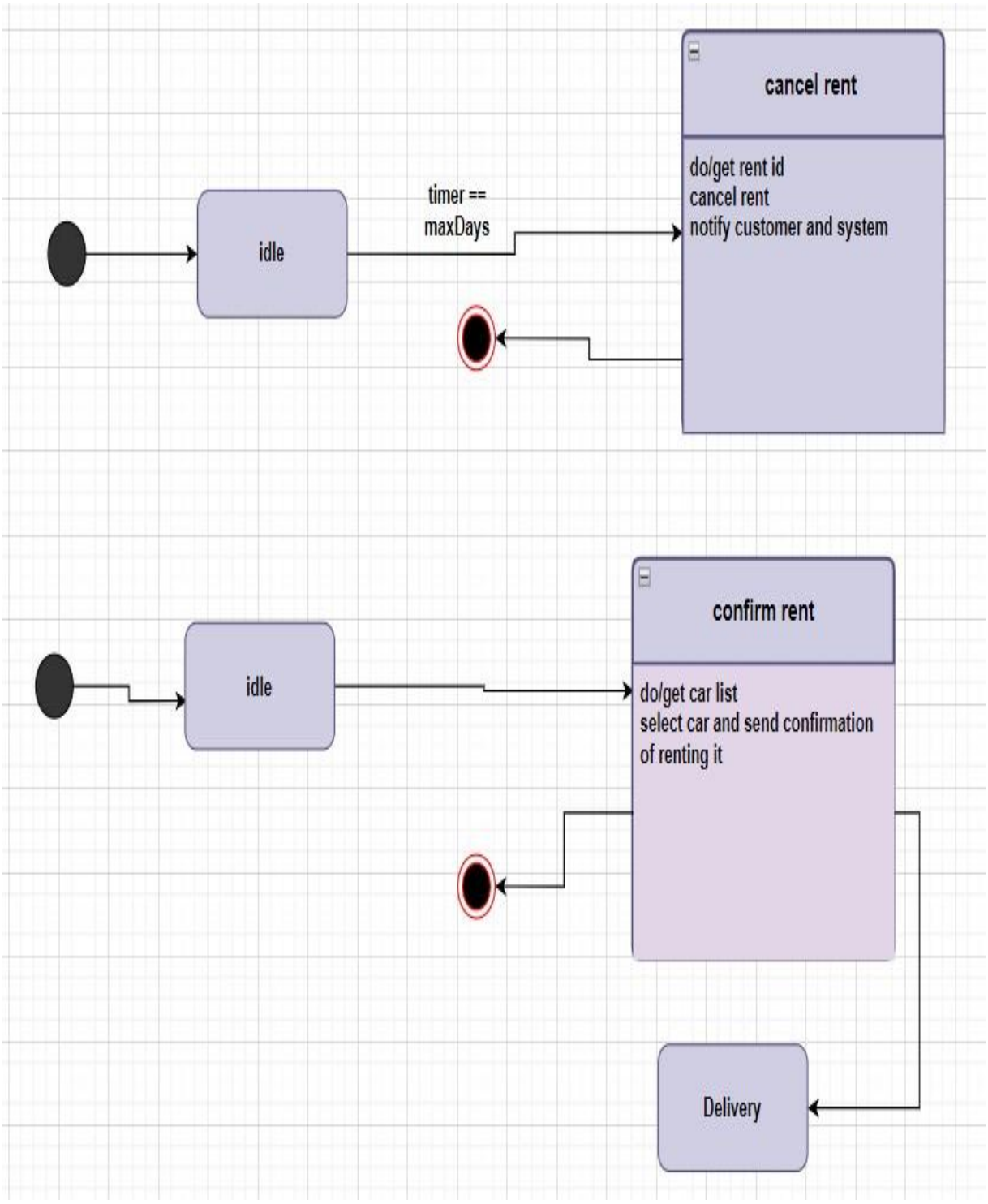
cancel rent
do/get rent id
cancel rent
notify customer and system
timer ==
maxDays
idle
confirm rent
do/get car list
select car and send confirmation
of renting it
idle
Delivery

Fig (6.4.7)

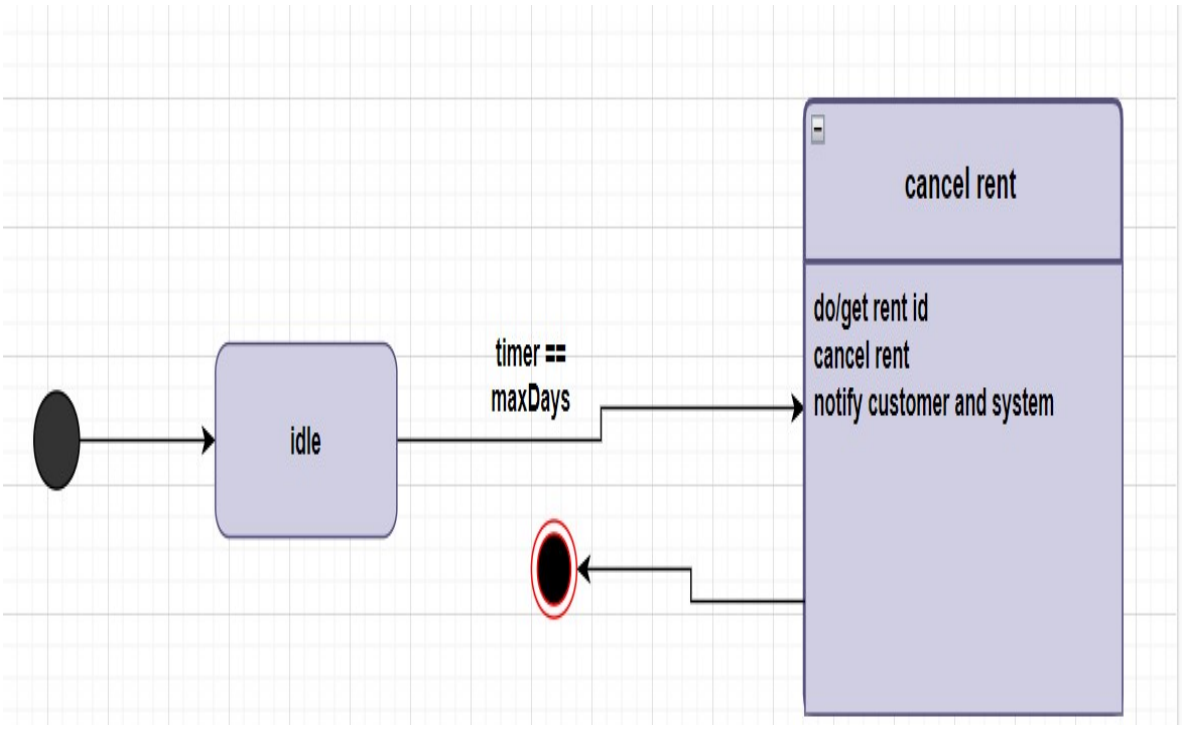

Fig (6.4.8)

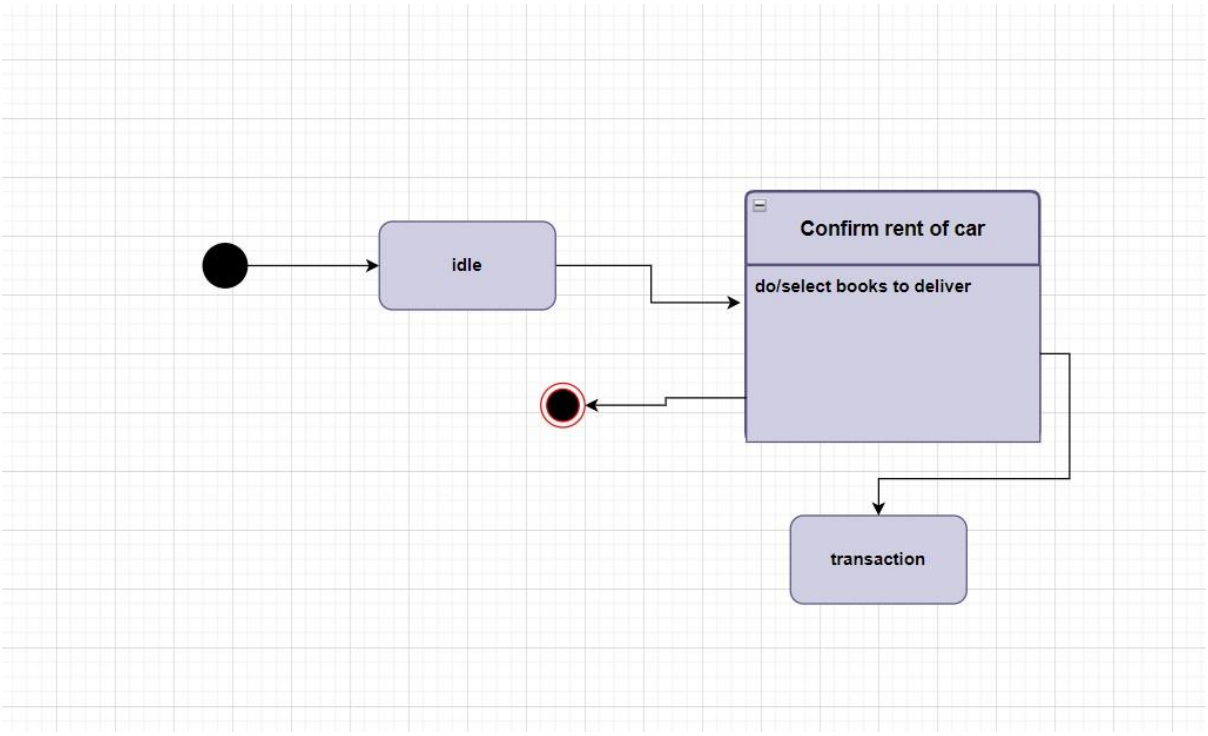

Fig (6.4.9)

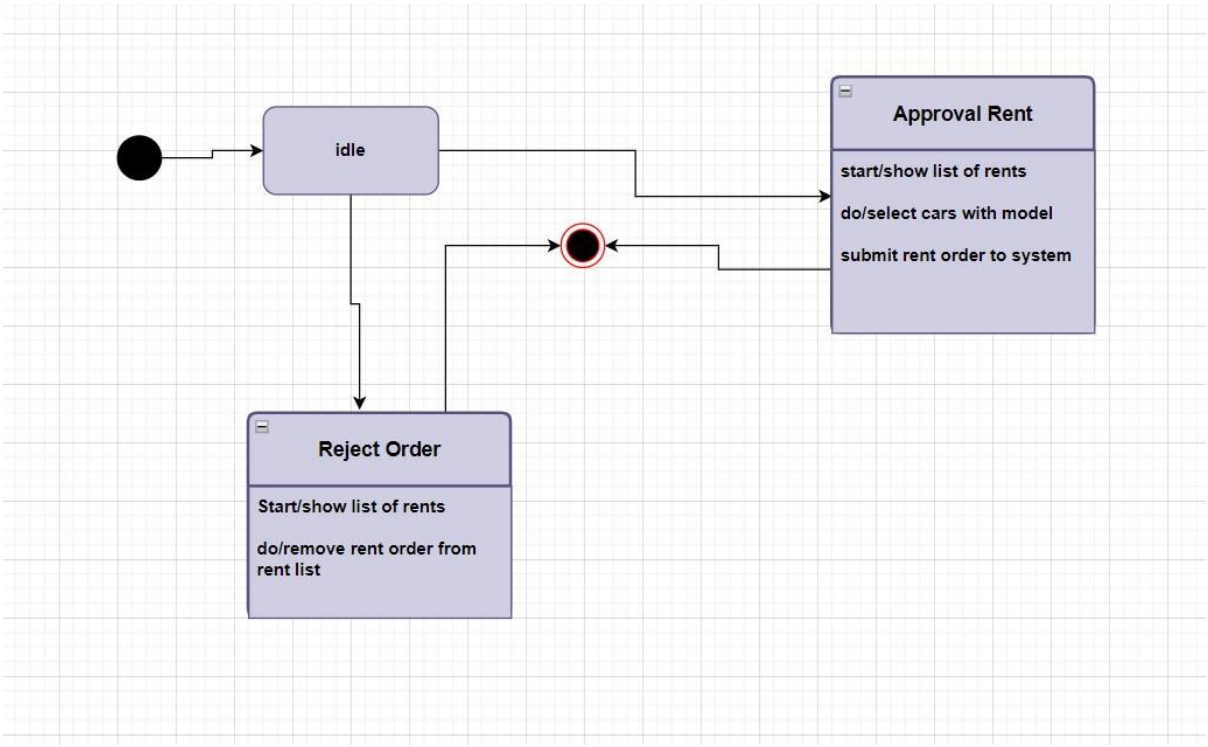

Fig (6.4.10)

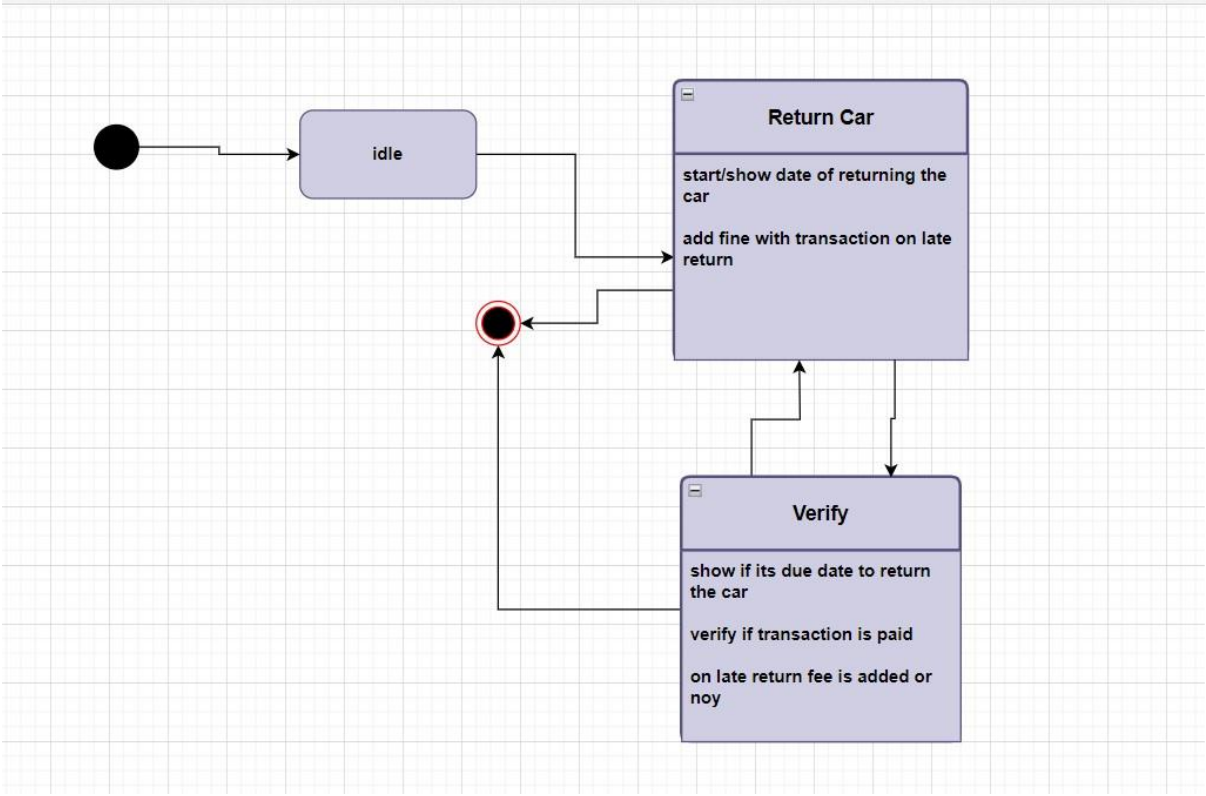

Fig (6.4.11)

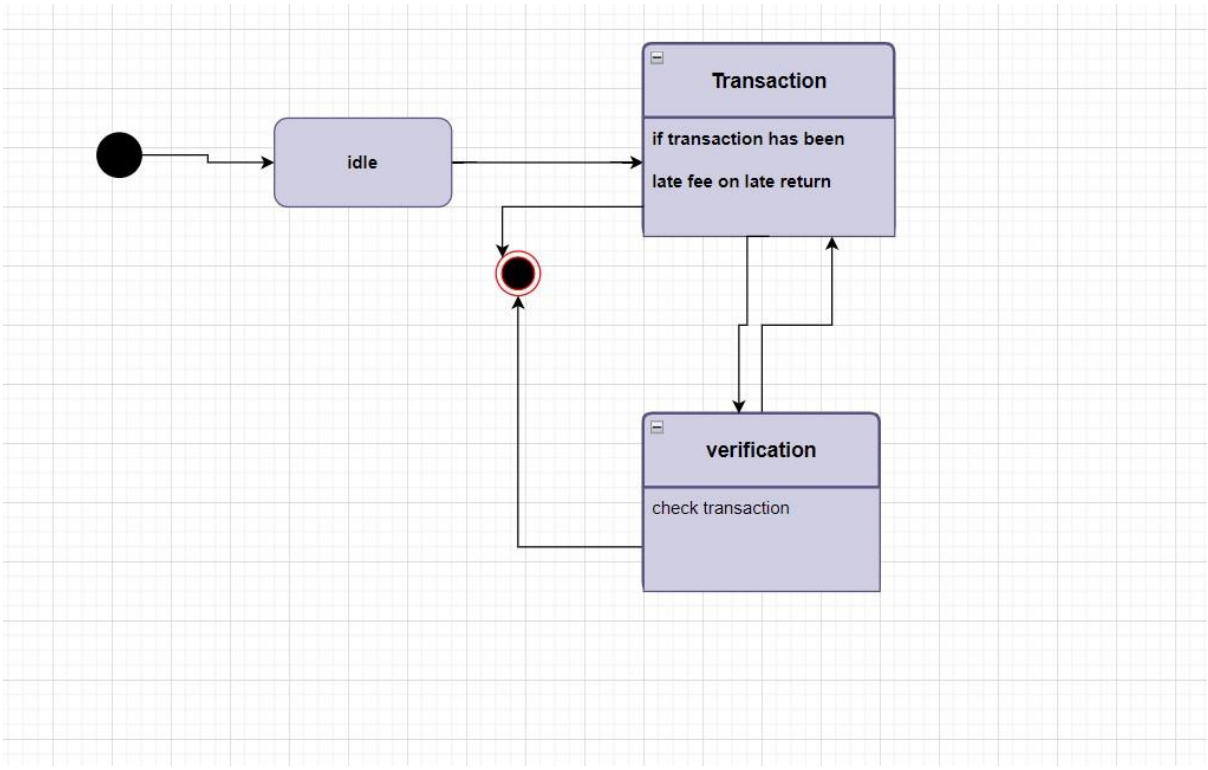

Fig (6.4.12)

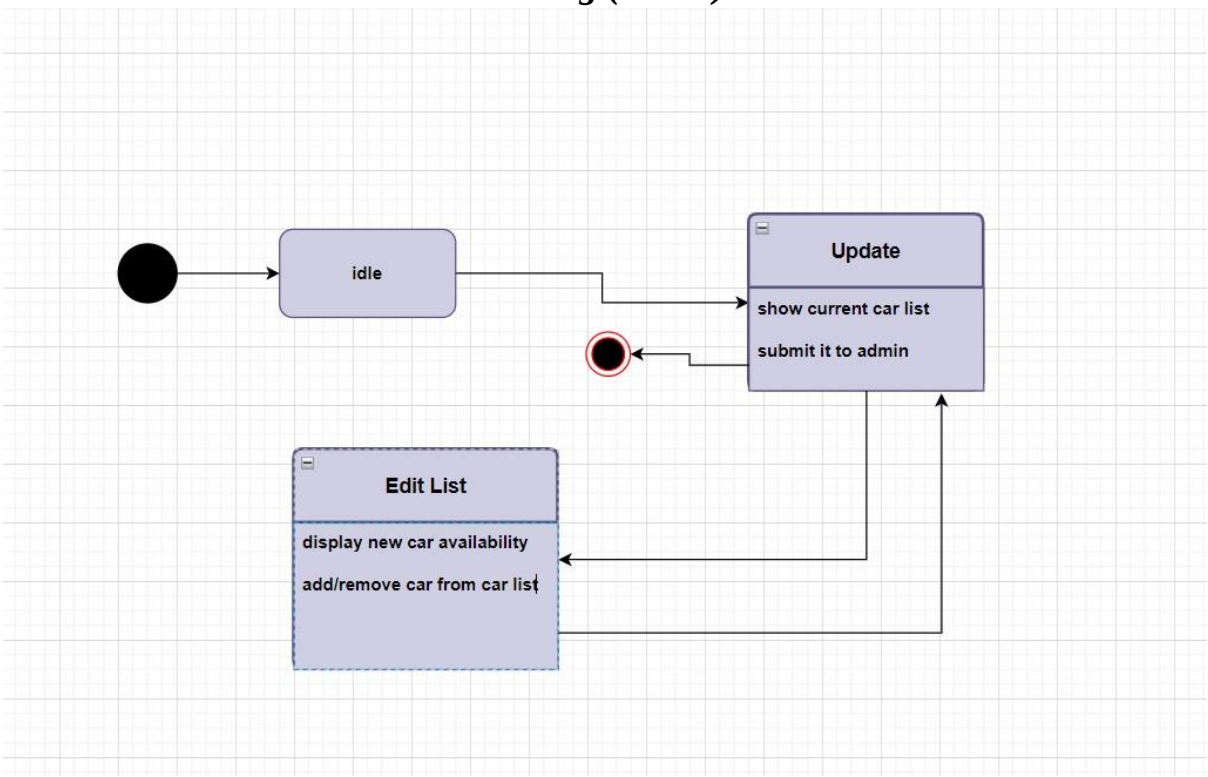

Fig (6.4.13)

6.5 Data Flow Diagrams:

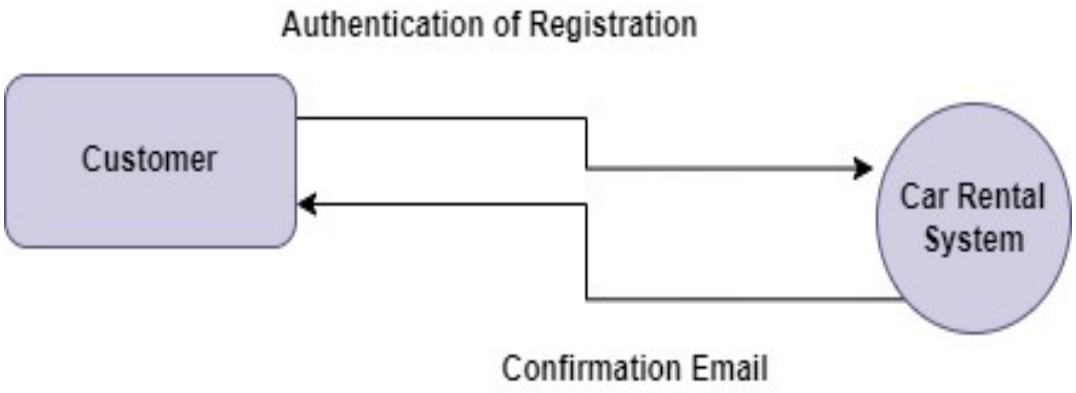

Fig (6.5.1)

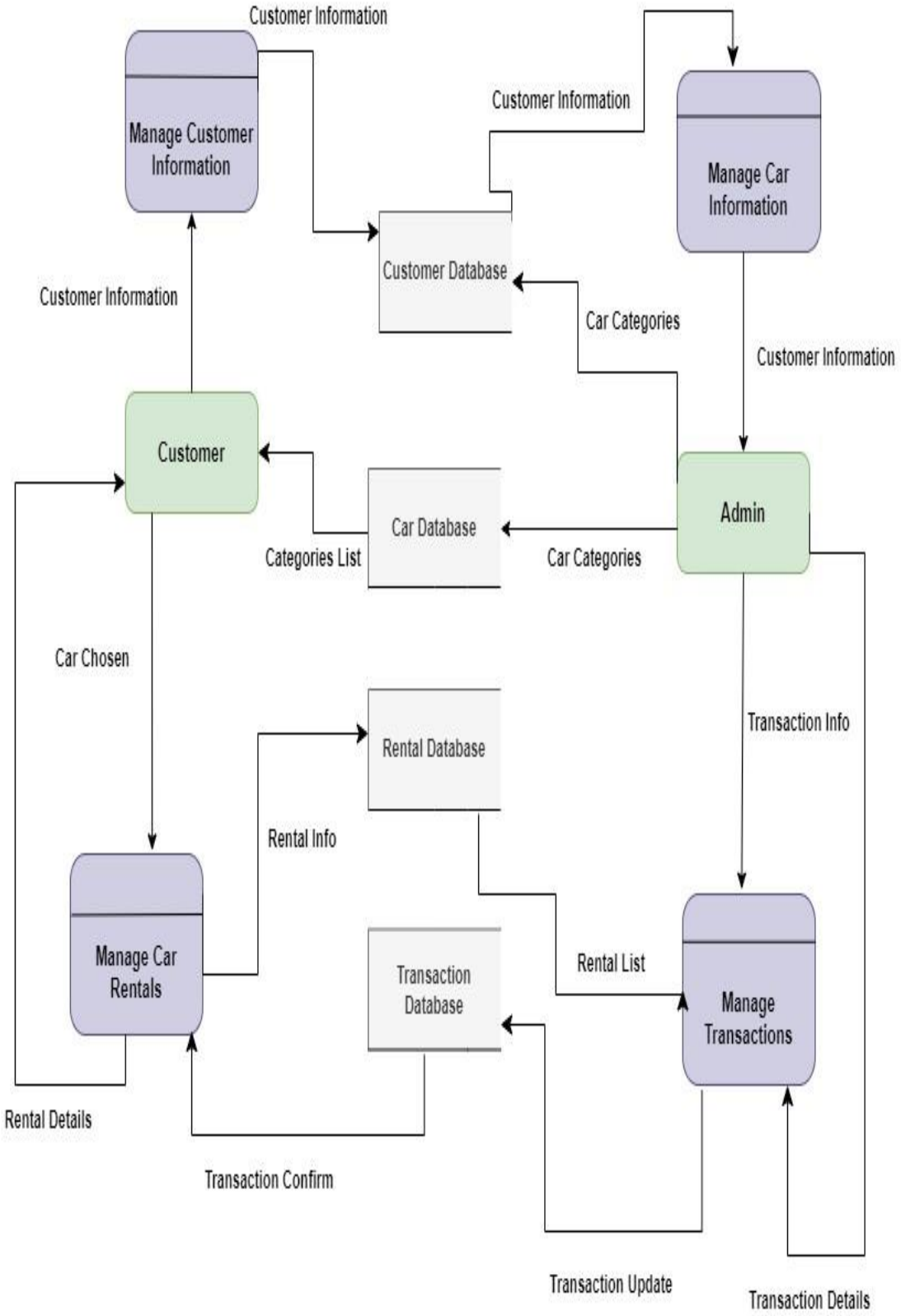
Customer Information
Manage Customer Information
Customer Information
Manage Car Information
Customer Database
Customer Information
Car Categories
Customer Information
Customer
Admin
Car Database
Categories List
Car Categories
Car Chosen
Transaction Info
Rental Database
Rental Info
Manage Car Rentals
Transaction Database
Rental List
Manage Transactions
Rental Details
Transaction Confirm
Transaction Update
Transaction Details

Fig (6.5.2)

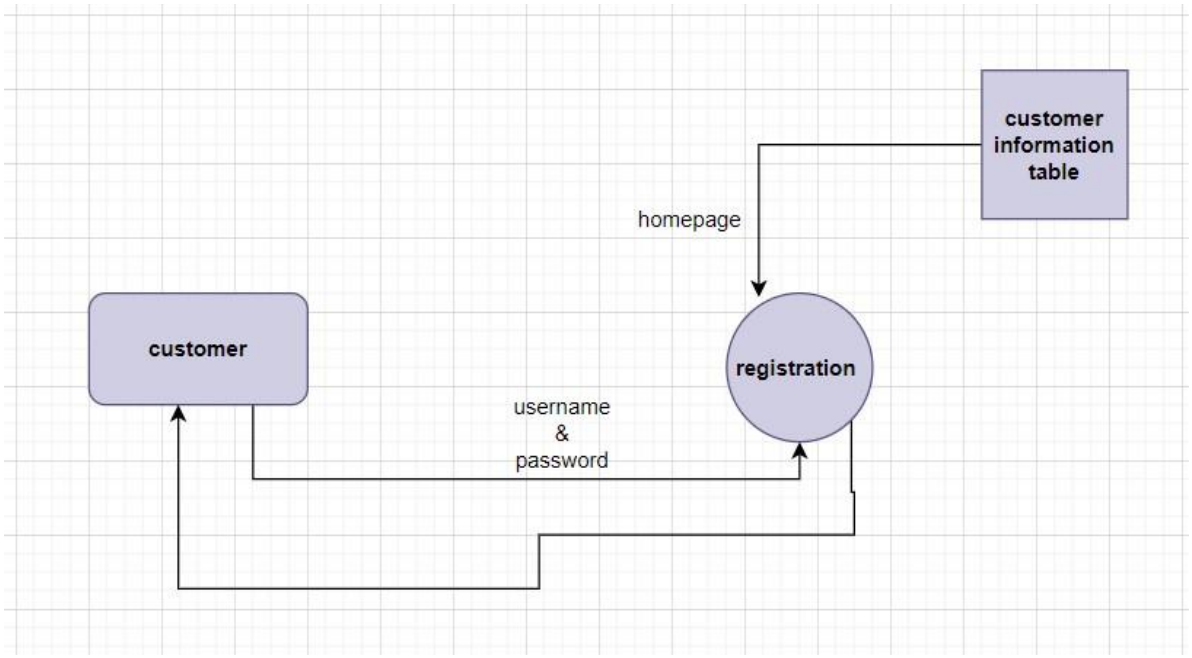

Fig (6.5.3)

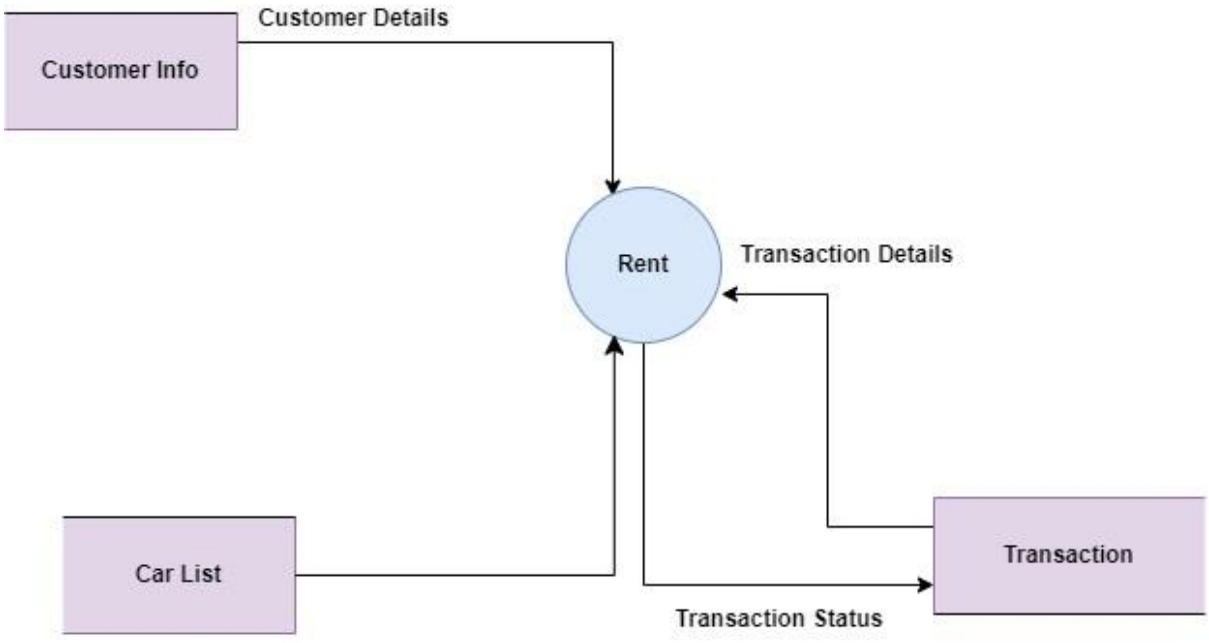

Fig (6.5.4)

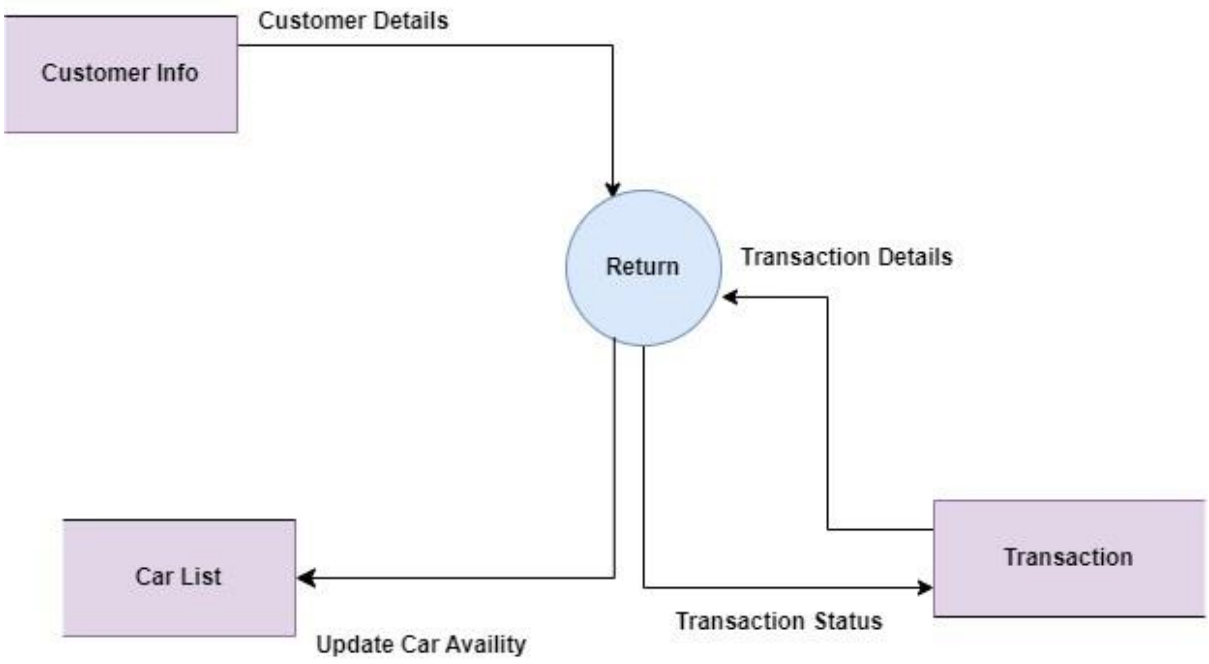
Customer Info
Customer Details
Return
Transaction Details
Car List
Update Car Availity
Transaction Status
Transaction

Fig (6.5.5)

6.6 Sequence Diagrams:

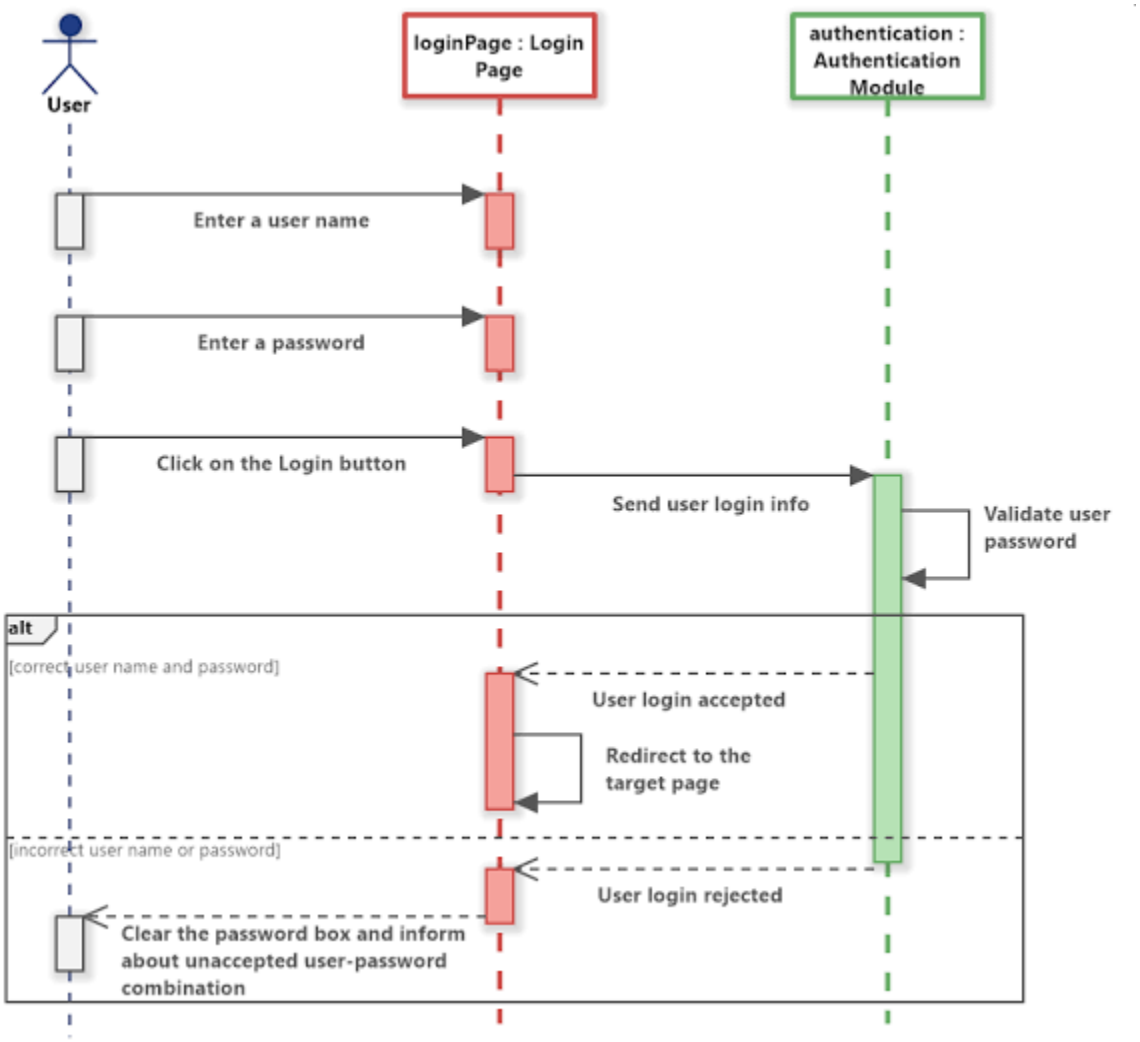

Fig (6.6.1)

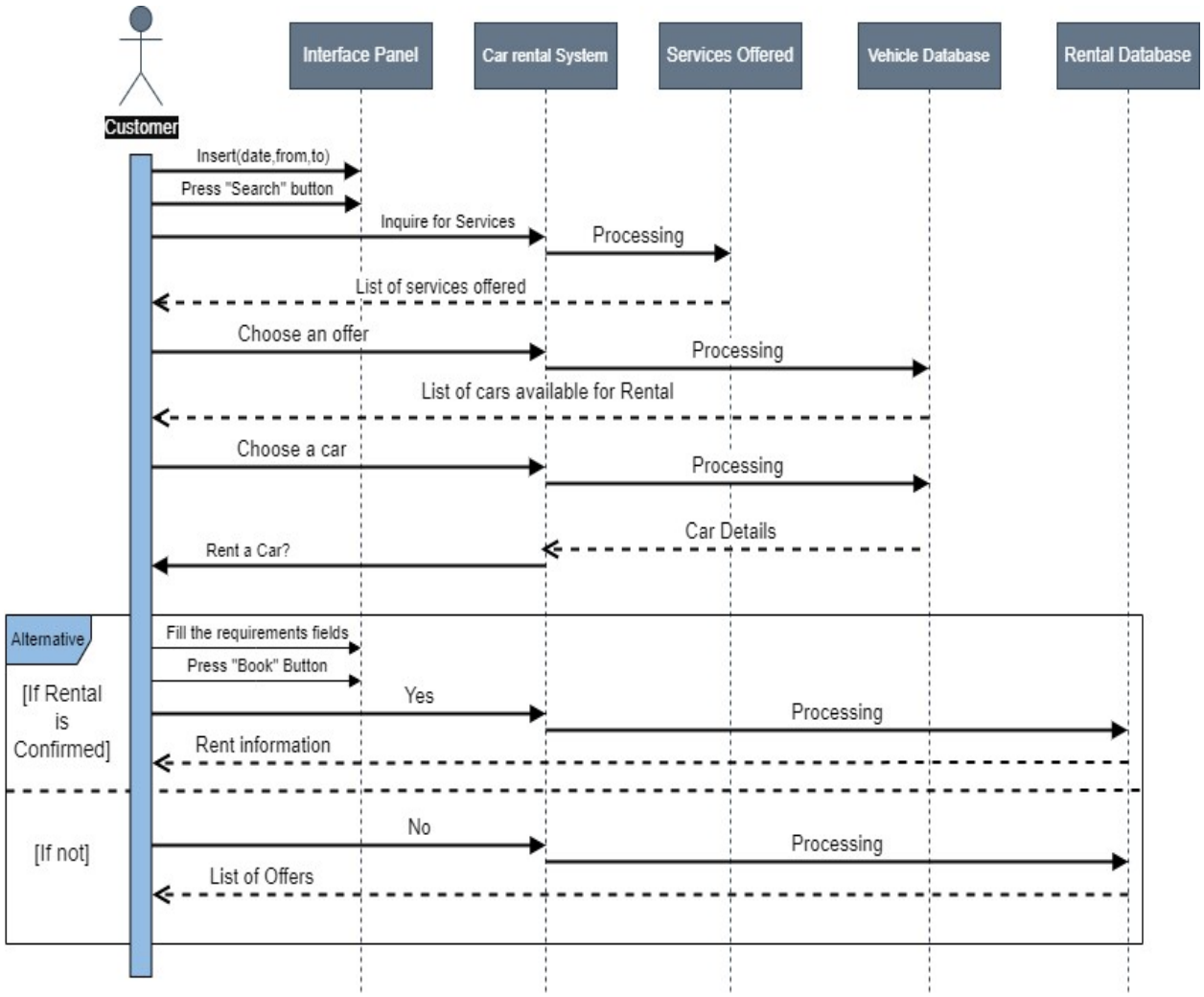
Customer
Interface Panel
Car rental System
Services Offered
Vehicle Database
Rental Database
Insert(date,from,to)
Press "Search" button
Inquire for Services
Processing
List of services offered
Choose an offer
Processing
List of cars available for Rental
Choose a car
Processing
Car Details
Rent a Car?
Alternative
[If Rental is Confirmed]
Fill the requirements fields
Press "Book" Button
Yes
Processing
Rent information
[If not]
No
Processing
List of Offers

Fig (6.6.2)

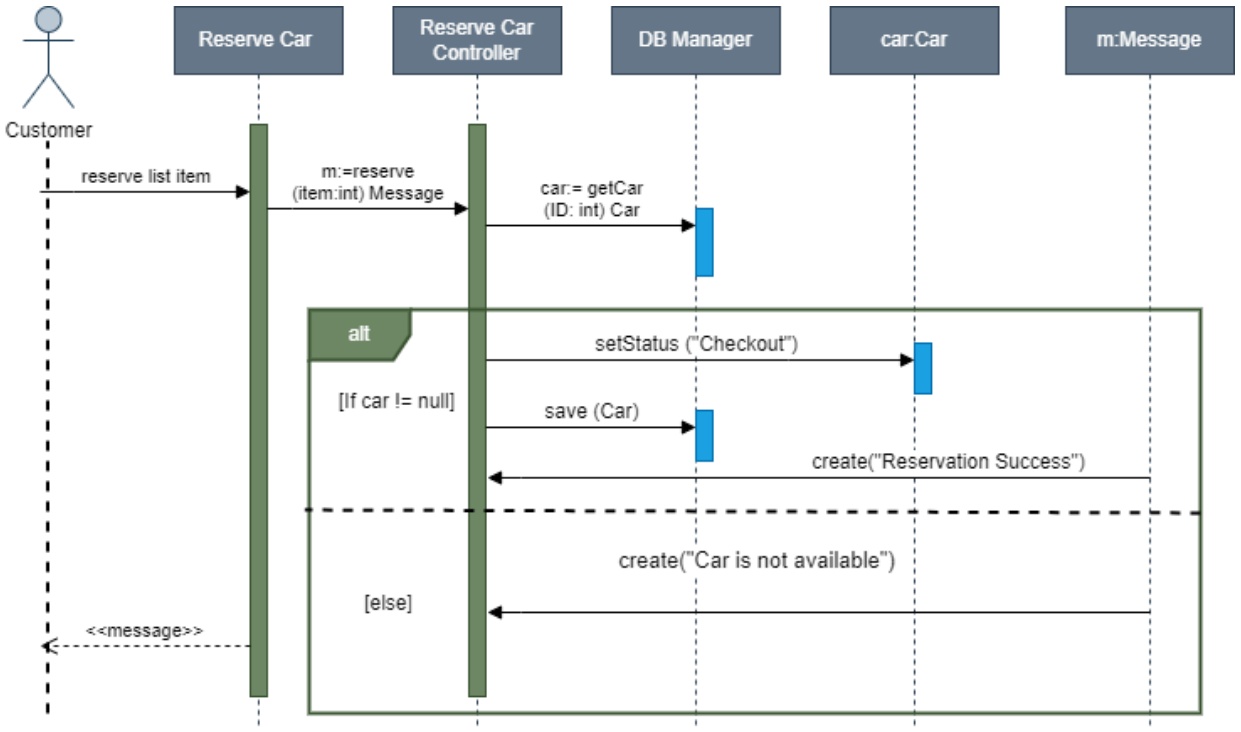

Fig (6.6.3)

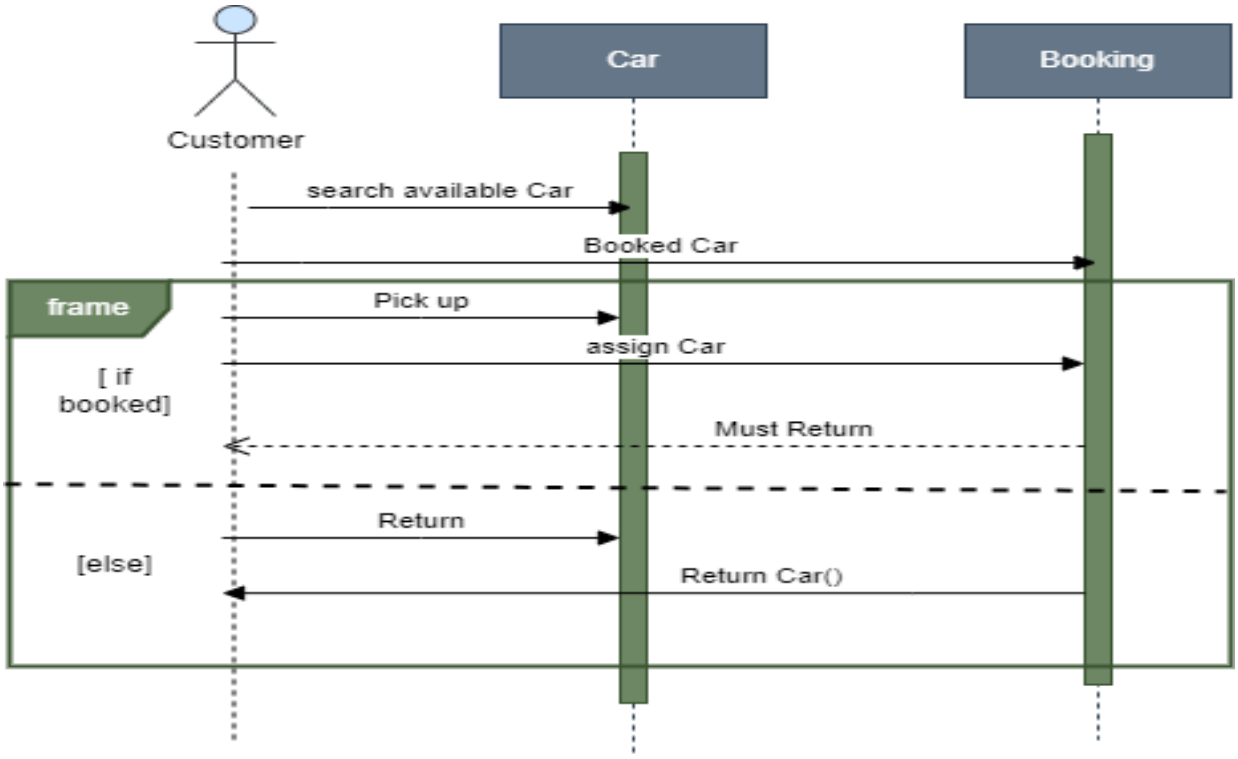

Fig (6.6.4)

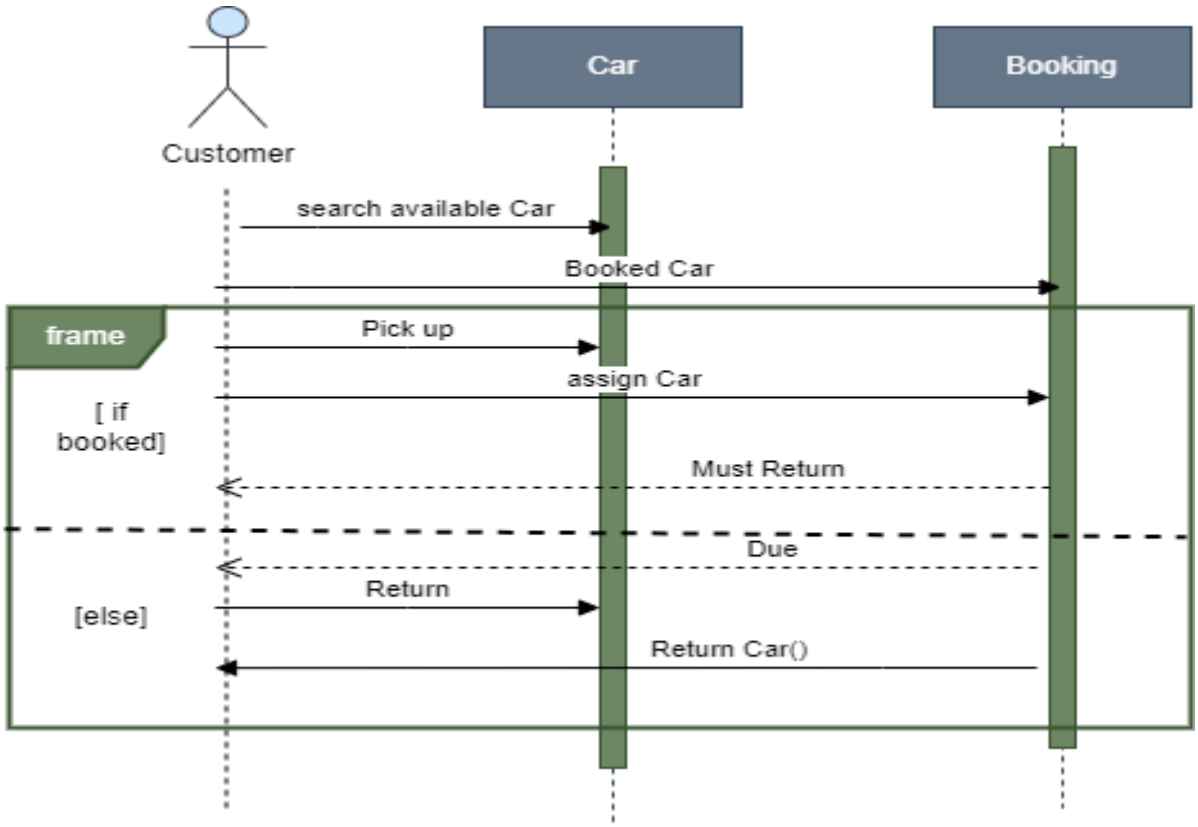
Customer
Car
Booking
search available Car
Booked Car
frame
Pick up
assign Car
[if
booked]
Must Return
Due
Return
[else]
Return Car()

Fig (6.6.5)