



Software Requirements Specification

Rental Vehicle Service

Prepared by

Abhishek Rawat (2101013)

Ganesh Kavthekar (2101099)

Pritesh Lohot (2101111)

Contents

1 Introduction

1.1 Purpose	3
1.2 Intended Audience	3
1.3 Intended Use	3
1.4 Product Scope	4
1.5 Risk Definition.....	4

2 Overall Description

2.1Product Perspective.	5
2.2Product Features.	5
2.3User Characteristics.....	6
2.4 Constraints	6
2.5 Operating Environment.	6

3 Requirements

3.1 Functional Requirements	7
3.2 Non-Functional Requirements	8

4.System Models

4.1 Use Case Diagram.....	9
4.2 Development Model.....	9-11
4.3 Data Flow Diagram (Level 0, Level 1 & Level 2)	12-13

5.Decision Tree and Decision Table

5.1 Decision Tree.....	14
5.2 Decision Table.....	15-18

6.Structure Chart

6.1 Structure Chart.....	19
--------------------------	----

Appendices

Update Section	11
-----------------------	-----------

1 Introduction

1.1 Purpose

The Efficient Vehicle Rental System aims to revolutionize the vehicle rental industry by leveraging cutting-edge technologies to enhance user experience, streamline operations, and optimize resource utilization.

1.2 Intended Audience

The intended audience for this SRS includes:

- **Development Team:** To understand the technical specifications and requirements needed for the implementation of the Renting Bikes project.
- **Design Team:** To grasp the design principles and user interface requirements for creating an aesthetically pleasing and user-friendly application.
- **Testing Team:** To develop test cases and scenarios based on the specified requirements and ensure the quality and reliability of the system.
- **Project Managers:** To oversee the project's progress, allocate resources effectively, and ensure that the final product aligns with the client's expectations.

1.3 Intended Use

The intended use of this comprehensive vehicle rental platform is to provide users with a seamless and technologically advanced solution for renting vehicles. The platform aims to leverage modern web technologies, deep fake image processing, and a neural network powered chatbot to enhance the user experience. By deploying the platform on AWS services, the goal is to achieve scalability, reliability, and efficient management.

Users, including vehicle renters, administrators, and those interacting with the chatbot, will benefit from a user-friendly interface that allows easy navigation, vehicle browsing, and initiation of rental processes. The integration of React JS for the frontend ensures an interactive and smooth user experience.

The platform's features, including deep fake image processing for user authentication and a chatbot powered by neural networks for natural language interaction, are designed to make the rental process efficient and user-friendly. The system's use of AWS services contributes to seamless scalability and effective management of the infrastructure.

1.4 Product Scope

The product scope encompasses the development of a cutting-edge vehicle rental platform. This includes a user-friendly frontend using React JS, a robust backend with JavaScript, MongoDB, and SQL databases, integration of advanced technologies like deep fake image processing, and a neural network-powered chatbot. The deployment will leverage AWS services for seamless scalability and efficient management, providing users with a secure, modern, and feature-rich vehicle rental experience.

1.5 Risk Definition

Technological Risks: Challenges in adopting advanced technologies like deep fake image processing and neural networks.

- **Security Risks:** Concerns regarding user data protection and unauthorized access.
- **Privacy Risks:** Addressing user-generated content and privacy compliance
- **Compliance Risks:** Ensuring adherence to data protection and privacy regulations.
- **Performance Risks:** Challenges in achieving low-latency content delivery and efficient image processing.
- **Scalability Risks:** Issues related to platform scalability during varying traffic loads.
- **User Adoption Risks:** Considerations for user acceptance and platform adoption.
- **Development Risks:** Challenges in meeting deadlines, addressing technical issues, and ensuring seamless integration.
- **Operational Risks:** Day-to-day operational challenges, including AWS infrastructure management.
- **Market Risks:** External factors impacting platform success, such as market changes and competition.

2 Overall Description

2.1 Product Perspective

Vehicle rental platform as a user-friendly interface. Incorporates advanced technologies: deep fake image processing and neural network-powered chatbot. Interaction with backend services, AWS infrastructure, and external machine learning components.

2.2 Product Features

2.2.1 User Interface (React JS):

Developed with React JS for an interactive user experience.

Users can browse, search, view vehicle details, and check availability.

2.2.2 Backend (JavaScript, MongoDB, SQL):

Backend in JavaScript, utilizing MongoDB for unstructured data and SQL for relational storage.

Requirements: Handle user authentication, authorization, and ensure efficient, secure data operations.

2.2.3 Image Processing:

The image processing technology should be seamlessly applied to authenticate users by matching the freshly clicked image with the information present on the driving license.

2.2.4 Chatbot (Neural Networks):

Features a chatbot powered by neural networks (TensorFlow or PyTorch).

Requirements: Understand and respond to user queries using NLP, with natural and intuitive conversations.

2.3 User Characteristics Primary

Users:

- **Vehicle Renters:** Individuals seeking to rent vehicles through the platform.
- **Administrators:** Responsible for managing and overseeing platform operations.
- **Chatbot Interactors:** Users engaging with the neural network-powered chatbot for assistance.

These user categories represent the diverse roles and interactions within the vehicle rental platform, catering to both those seeking services and those responsible for platform administration.

2.4 Constraints

- **Security:** Adherence to high-security standards for user data protection.
- **Privacy:** Address privacy concerns related to user-generated content and personal information.
- **Compliance:** Must comply with relevant data protection and privacy regulations.

2.5 Operating Environment

- **Description:** The vehicle rental platform operates in a dynamic online environment, accessible through web browsers.

Users can engage with the platform from various devices, including desktops, laptops, and mobile devices.

- **Key Points:** Online platform accessible via web browsers. Compatibility with various devices: desktops, laptops, and mobile devices.

The operating environment ensures accessibility and flexibility for users across different devices, fostering a seamless experience.

3 Requirements

3.1 Functional Requirements

3.1.1 Functionality

3.1.1.1 User Interface

- **Requirement:** Users should be able to easily navigate the platform, search for vehicles, and initiate rental processes.

3.1.1.2 Image Processing

- **Requirement:** The image processing technology should be seamlessly applied to authenticate users by matching the freshly clicked image with the information present on the driving license. The system should ensure a reliable and accurate verification process for user identification.

3.1.1.3 Chatbot

- **Requirement:** The chatbot should provide natural language responses to user queries, enhancing user interaction.

3.1.2 Performance Requirements

- The system should ensure low-latency content delivery, leveraging AWS CloudFront.
- Image processing should be efficient and not introduce significant delays in user interactions.

3.1.3 External Interface Requirements

3.1.3.1 AWS Services

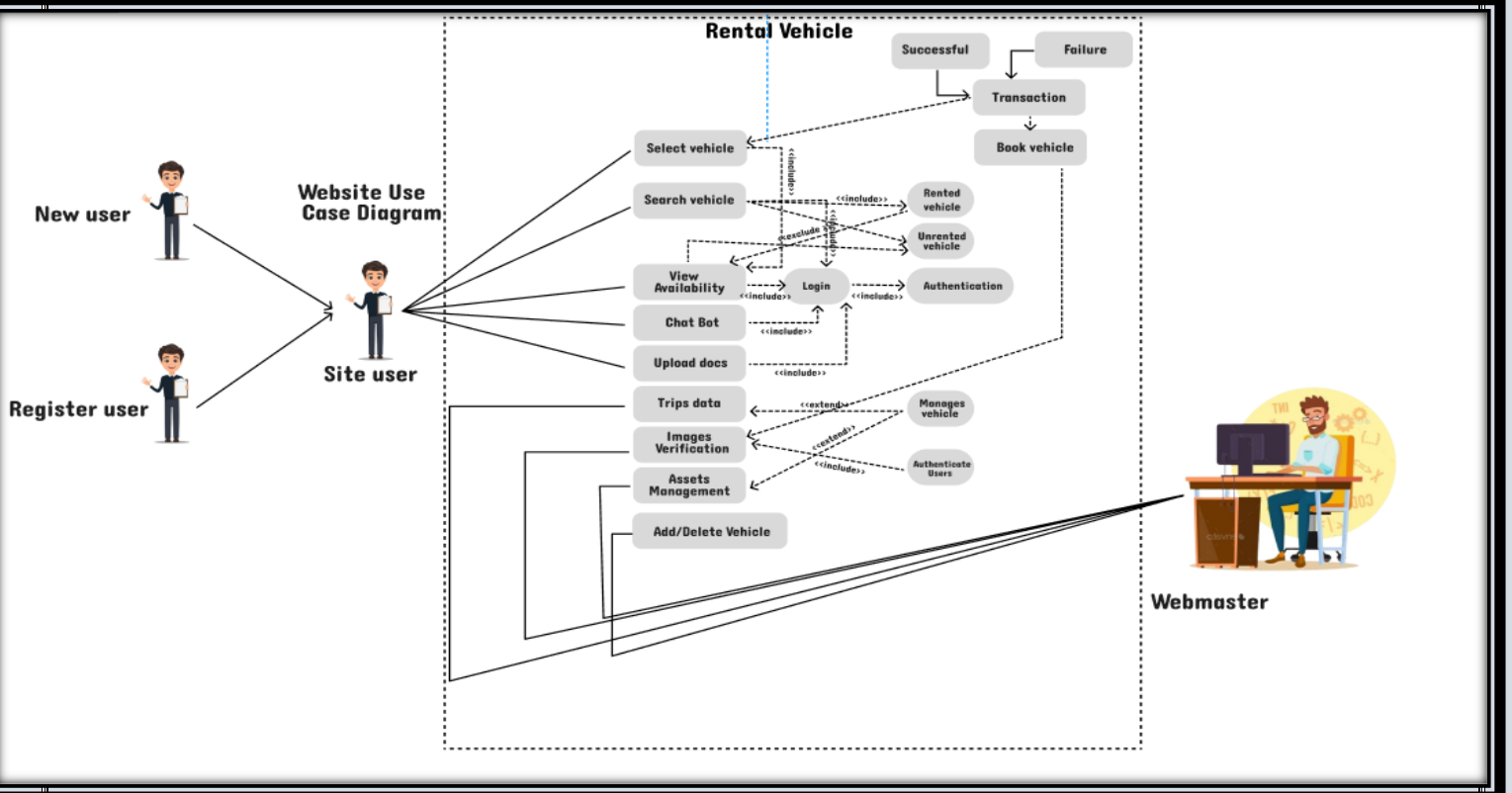
- **DynamoDB:** The system should efficiently store and retrieve data from DynamoDB.
- **CloudFront Distribution:** Content delivery should be optimized for low-latency.
- **API Gateway:** APIs should facilitate communication between frontend, backend, and machine learning components.
- **Elastic Beanstalk:** The deployment process should be smooth and easily manageable.
- **CloudWatch:** The system should be monitored for performance, and logs should be maintained. • **Autoscaling:** The infrastructure should automatically adjust to varying traffic loads.

3.2 Non-Functional Requirements

- **Security:** Implement robust security measures for the entire system. Include encryption of sensitive data and ensure secure API communications.
- **Privacy:** Handle user-generated content and personal information with strict privacy controls. Ensure compliance with privacy standards and regulations.
- **Compliance:** Ensure the platform complies with relevant data protection and privacy regulations. Adhere to legal and industry standards for data handling and user privacy.

This section emphasizes the critical non-functional aspects of the vehicle rental platform, focusing on security, privacy, and compliance to ensure a trustworthy and lawful application.

4 System Models



Link: -

<https://www.figma.com/file/0t06GMFB7vhDtCXIoDY46B/use-case--diagram?type=design&node-id=12%3A3&mode=design&t=VprH4UOah4MkycTN-1>

4.2 Development Model: Iterative Waterfall Model

Development Model: Iterative Waterfall Model ○ Description:

The Iterative Waterfall Model is a modified version of the traditional Waterfall Model, incorporating iterative cycles to enhance flexibility and adaptability.

○ **Key Points:**

- **Iterative Approach:**

Integrates iterative cycles within each phase, allowing for feedback and adjustments.

- **Document-Driven:**

Similar to the Waterfall Model, emphasizes documentation at each stage of development.

- **Enhanced Flexibility:**

Incorporates flexibility by allowing revisits to previous phases based on feedback and changing requirements.

○ **Phases in the Iterative Waterfall Model:**

- **Requirements:**

Initial gathering and documentation of project requirements.

- **Design:**

Creation of a detailed design with the option to revisit based on feedback.

- **Implementation:**

Coding and implementation with the ability to iterate for improvements.

- **Testing:**

Iterative testing cycles to identify and address issues as they arise.

- **Deployment:**

Deployment of the system in increments or with the flexibility to revisit if necessary.

- **Maintenance:**

Ongoing maintenance and support with the potential for iterative improvements.

○ **Advantages:**

- **Flexibility with Iterations:**

Allows revisiting and iterating on previous phases, enhancing adaptability to changing requirements.

- **Document-Driven Approach:**

Emphasizes documentation, aiding in understanding and maintaining the system.

Suitable for Evolving Projects:

Well-suited for projects where requirements may evolve or change during the development process.

○ **Disadvantages:**

- **Complexity of Iterations:**

The iterative nature can introduce complexity and may require effective management.

- **Potential for Extended Timelines:**

Iterations may lead to longer development timelines if not managed efficiently. •

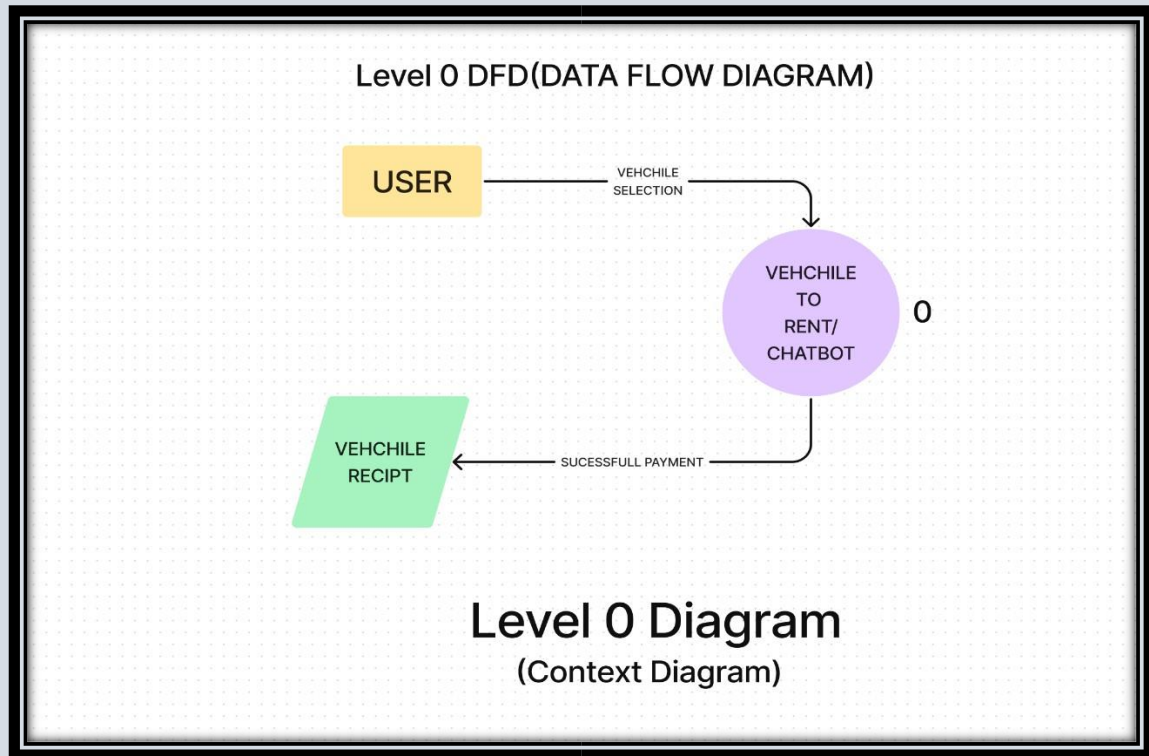
Suitability:

The Iterative Waterfall Model is suitable for projects with evolving or unclear requirements, providing flexibility through iterative cycles.

It strikes a balance between the structure of the Waterfall Model and the adaptability of iterative development.

○ **Conclusion:**

The Iterative Waterfall Model offers a compromise between the structured approach of the Waterfall Model and the flexibility of iterative development. It is well-suited for projects where requirements may change or evolve, allowing for adjustments throughout the development process.



4.3 Data Flow Diagram

Level 0:

<https://www.figma.com/file/72M1uD0ZqdnXIBASdWwxqy/DFD?type=whiteboard&node-id=01&t=3fiErxQoouvmrP1Q-0>

Level 1:

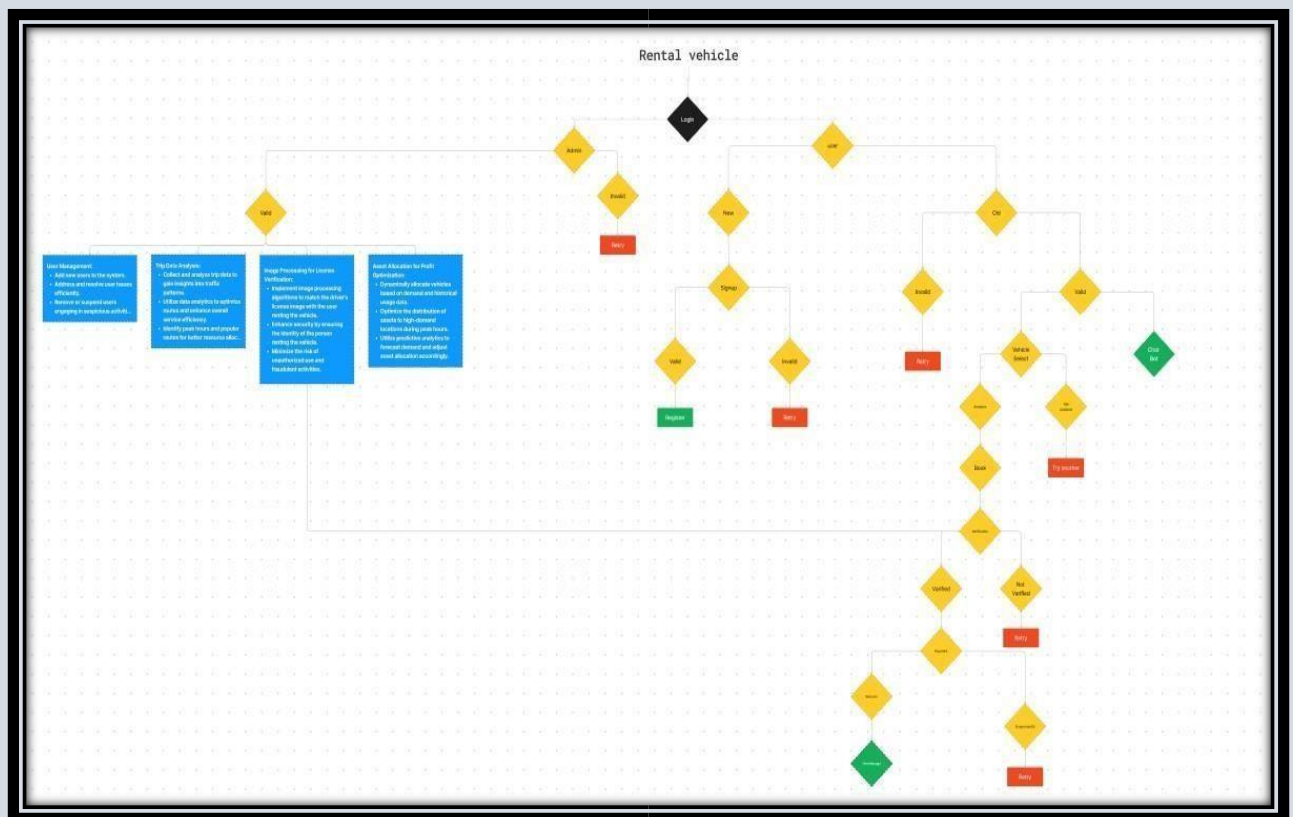
<https://www.figma.com/file/72M1uD0ZqdnXIBASdWwxqy/DFD?type=whiteboard&node-id=01&t=3fiErxQoouvmrP1Q-0>

Level 2: [https://www.figma.com/file/XSbpUFJIIGoPa9qSHVtOru/Data-Flow-](https://www.figma.com/file/XSbpUFJIIGoPa9qSHVtOru/Data-Flow-Diagram(Rentalbhai)?type=whiteboard&node-id=0%3A1&t=gEhprTrhgmdsewQV-1)

[Diagram\(Rentalbhai\)?type=whiteboard&node-id=0%3A1&t=gEhprTrhgmdsewQV-1](https://www.figma.com/file/XSbpUFJIIGoPa9qSHVtOru/Data-Flow-Diagram(Rentalbhai)?type=whiteboard&node-id=0%3A1&t=gEhprTrhgmdsewQV-1)

5 Decision Tree and Decision Table

5.1 Decision Tree: -



[https://www.figma.com/file/xaX7Nxuzb3bc8etUYy9ApC/Decision-Tree-\(FigJam\)--uxchunks\(Community\)-\(Community\)?type=whiteboard&node-id=0-1&t=MJhW58vi1VM5bldZ-0](https://www.figma.com/file/xaX7Nxuzb3bc8etUYy9ApC/Decision-Tree-(FigJam)--uxchunks(Community)-(Community)?type=whiteboard&node-id=0-1&t=MJhW58vi1VM5bldZ-0)

5.2 Decision Table: -

❖ Login Page

Rules				
Conditions	1	2	3	4
Admin Login	T	-	T	-
User Login	-	T	-	T
Valid Password/Id	T	-	-	T
Invalid Password/ ID	-	T	T	-
Actions				
Successful Login	T	-	-	T
Retry	-	T	T	-

❖ User Panel

Rules							
Conditions	1	2	3	4	5	6	7
Validation	F	T	T	T	T	T	T
Already Register User	-	F	T	T	T	T	T
Vehicle Selection	-	-	F	T	T	T	T
Surfing	-	-	T	F	F	F	F
Vehicle Availability	-	-	-	F	T	T	T
Verification	-	-	-	-	F	T	T
Transaction	-	-	-	-	-	F	T
Actions							
Registration	-	T	-	-	-	-	-

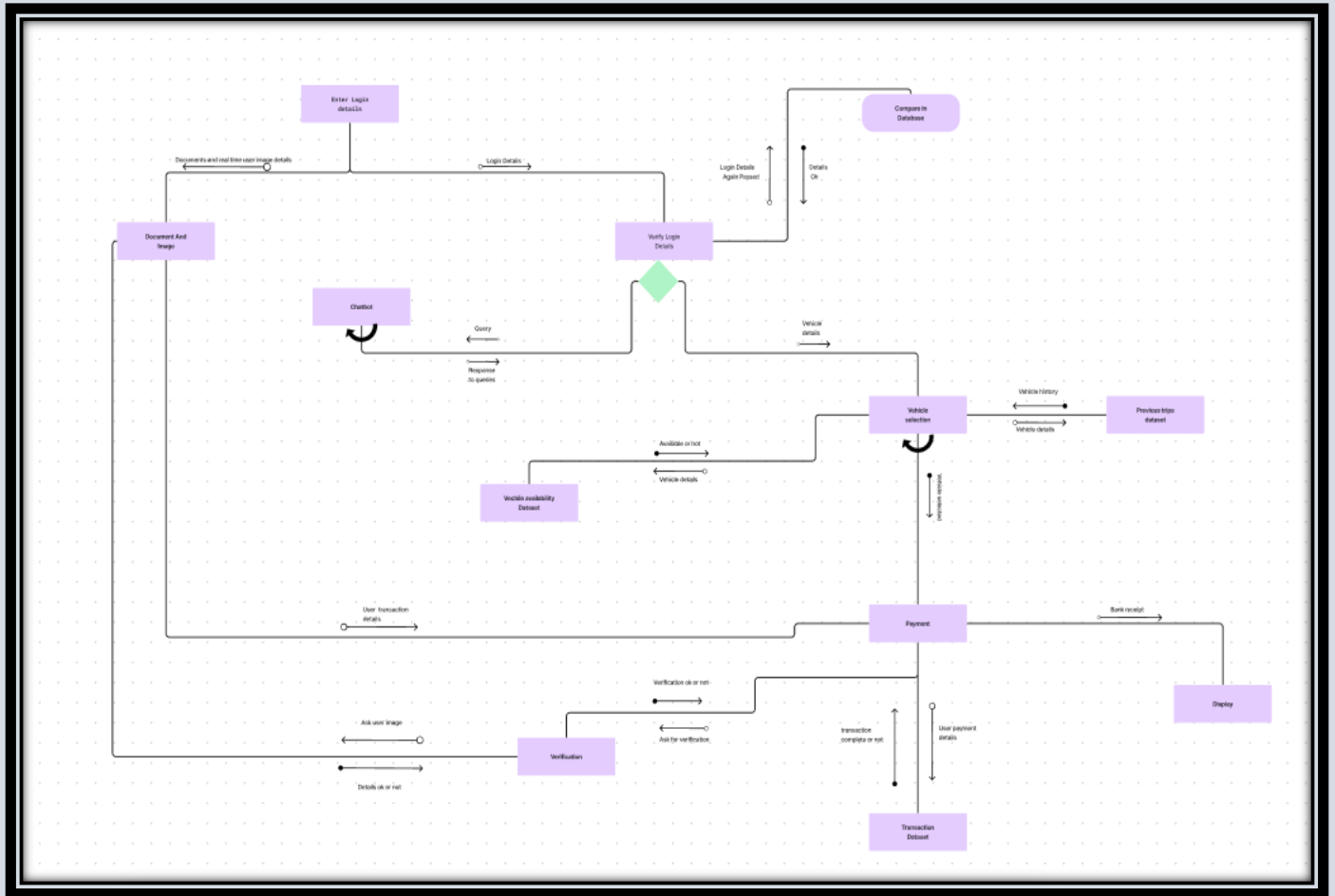
Chatbot Interaction	-	-	T	-	-	-	-
Retry Verification	-	-	-	-	T	-	-
Try Another Vehicle	-	-	-	T	-	-	-
Vehicle Booked	-	-	-	-	-	-	T
Transaction failed (Retry)	-	-	-	-	-	T	-
Retry Logging	T	-	-	-	-	-	-
Print Payment Receipt	-	-	-	-	-	-	T

❖ Admin Panel

Rules					
Conditions	1	2	3	4	5
User Management	T	T	-	-	-
Add user	T	F	-	-	-

Remove user	F	T	-	-	-
Vehicle management	-	-	T	T	-
Quality of Vehicle	-	-	T	F	-
Add Vehicle	-	-	T	F	-
Remove Vehicle	-	-	F	T	-
Trip Completion Status	-	-	-	-	T
Actions					
Add User Docs.	T	-	-	-	-
Remove User Docs.	-	T	-	-	-
Add Vehicle Docs.	-	-	T	-	-
Remove Vehicle Docs.	-	-	-	T	-
Validating/Calculating Trip data	-	-	-	-	T

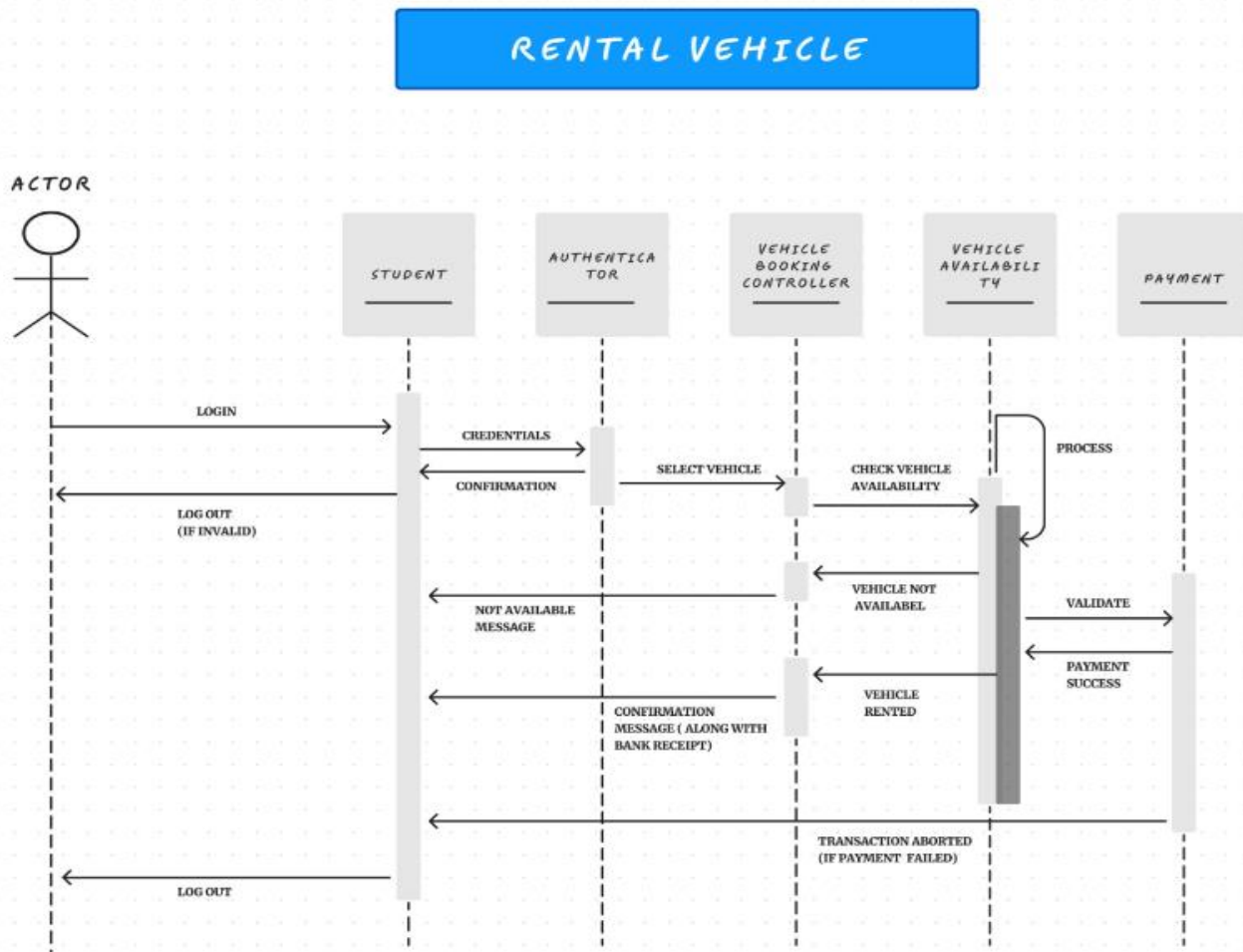
6 Structure Chart: -



Link: -

<https://www.figma.com/file/72M1uD0ZqdnXIBASdWwxqy/DFD?type=whiteboard&node->

Sequence Diagram: -



SEQUENCE DIAGRAM

Link: -

<https://www.figma.com/file/qjnHN00N4U74d2wTPsLGFE/Untitled?type=whiteboard&node-id=0-1&t=zitMtlHt0xkFcph5-0>

Update Section:

† Incorporated the development lifecycle model.