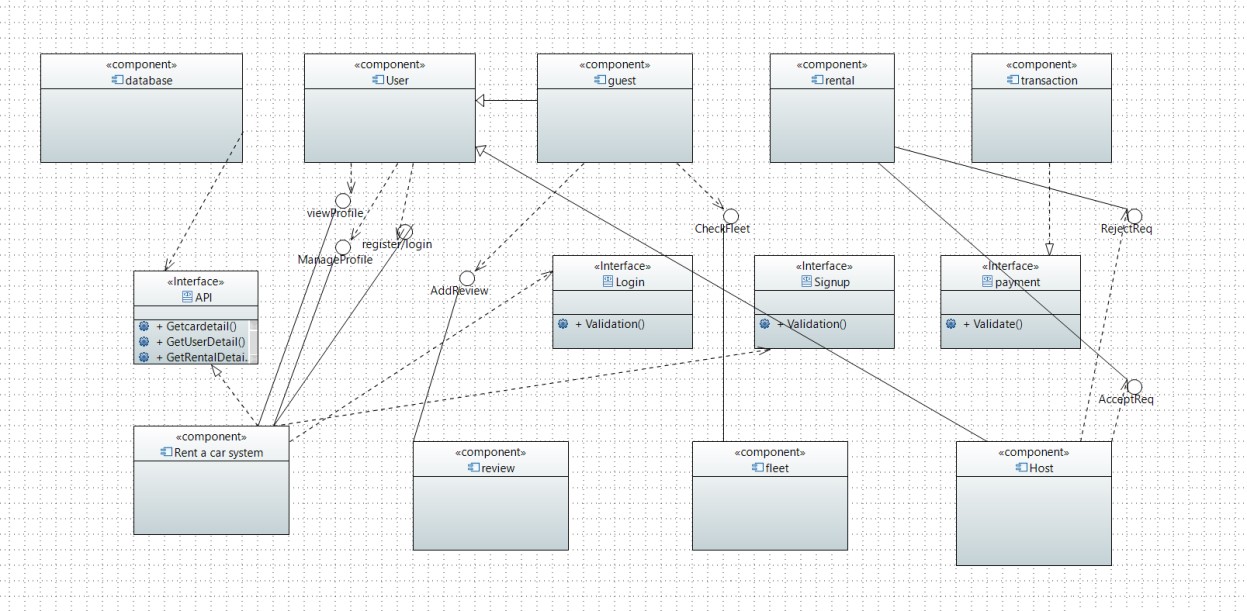
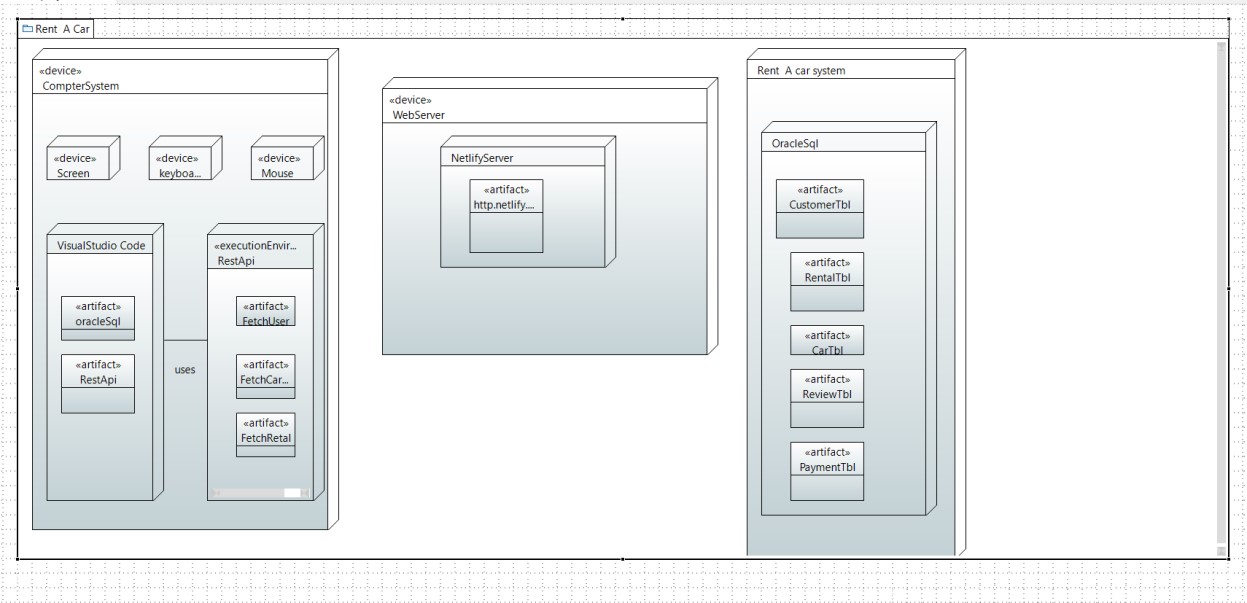
# Software Design Specifications (SDS)

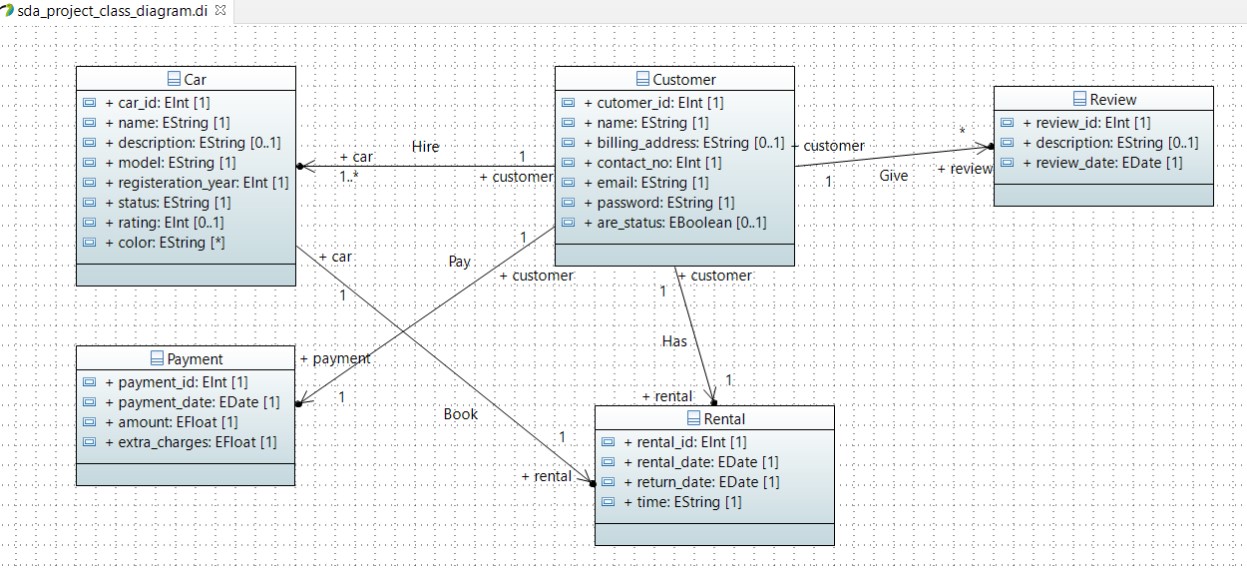
**System Architecture:**

**System Level Architecture:**

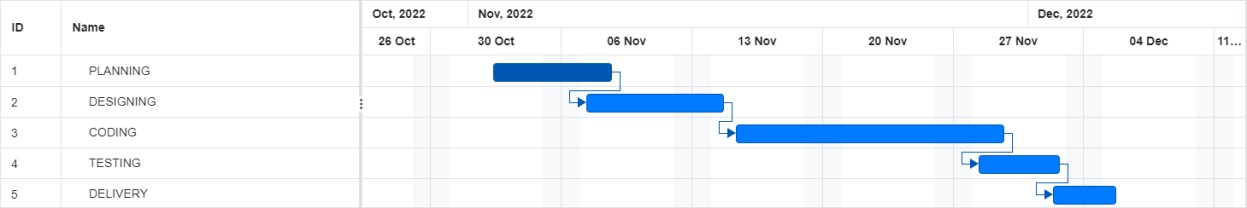


*The system used to execute the software is computer system consisting of keyboard, screen and track-pad to control or enter the data. The software is run using visual studio code containing the libraries which includes Fast API a python framework, which are used to create APIs to manipulate database which is connected to our React frontend which user uses to interact. This software uses HTTP protocols and is executed on web browser using local host.*

**Software Architecture:**



**Design Strategy:**

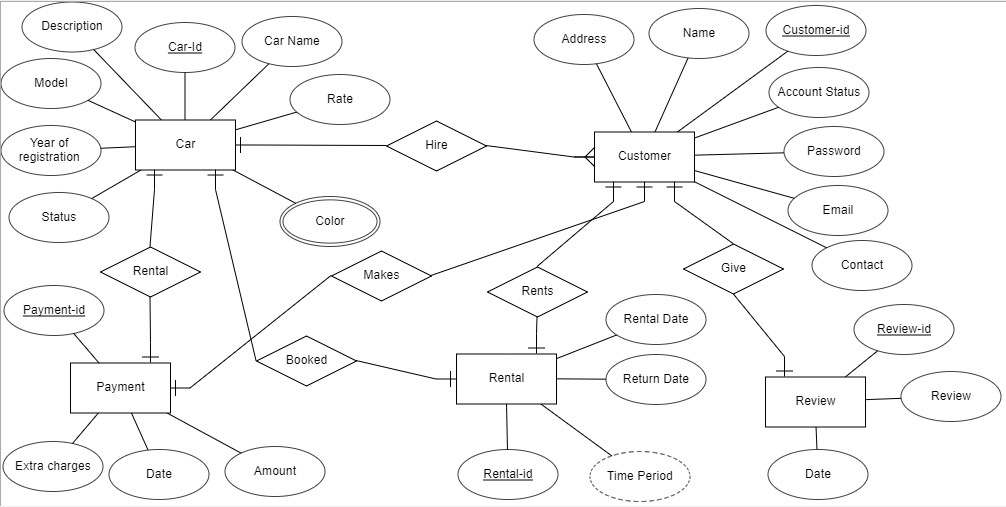


*During first 6 days we collected the information related to our project and problem statement and design a proposal which work as a ground work of what our project will do and what technology will be used. Then UML diagrams were generated to get an idea of how our system will interact with different hardware, software and its user and what features it will provide. Next was the largest phase of coding where we setup work environments and database were built and connected using API to the frontend and then testing was done with dummy data and next project was delivered.*

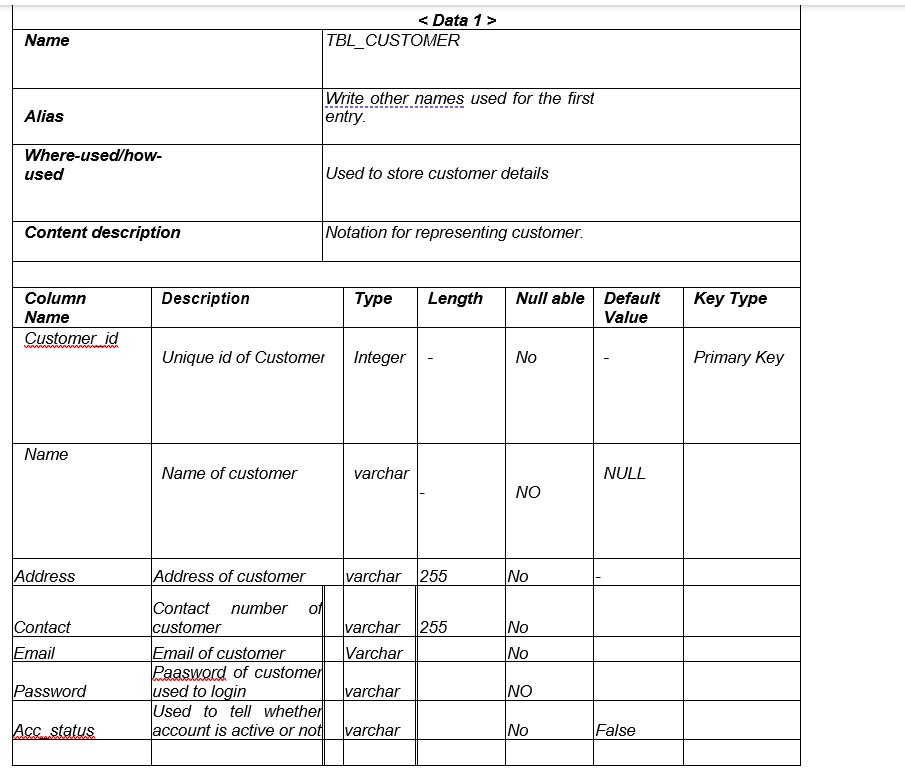
**Detailed System Design:**

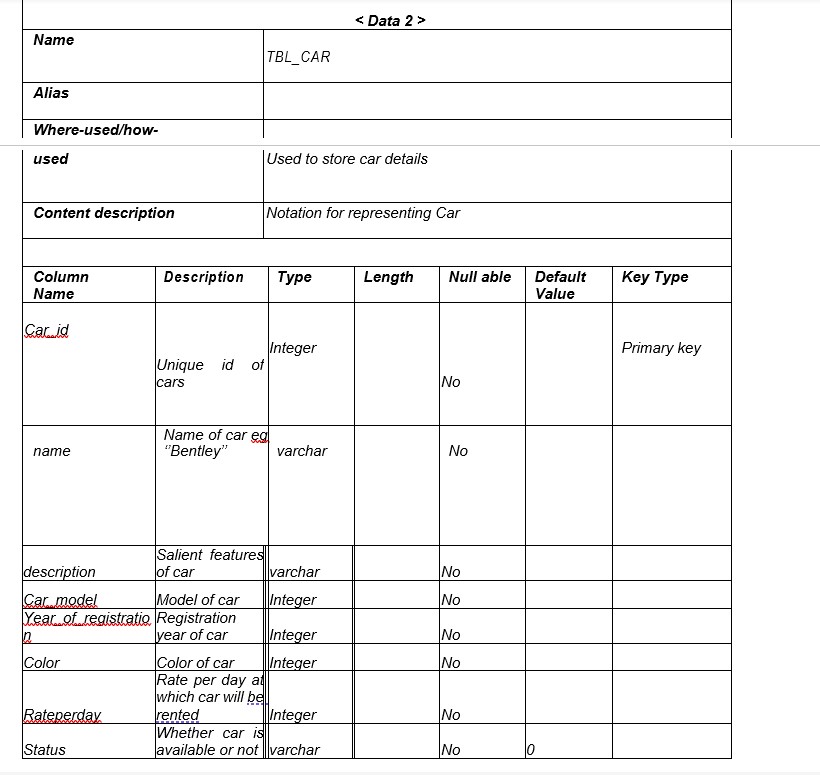
**Database Design:**

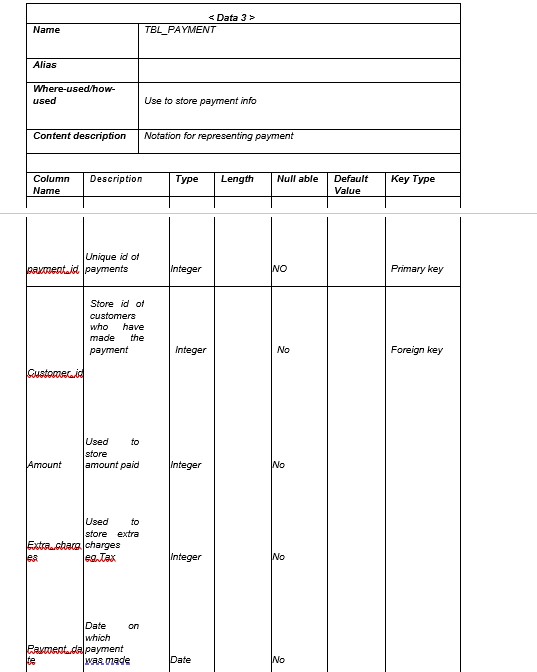
**ER Diagram:**

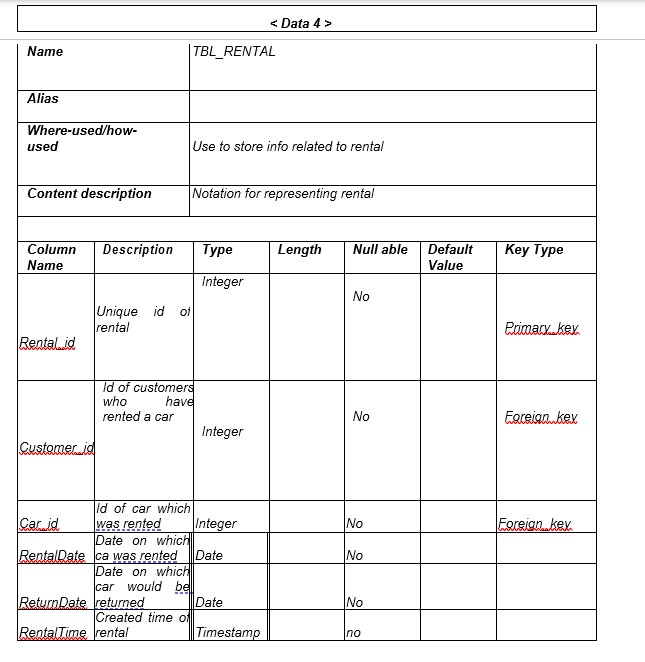


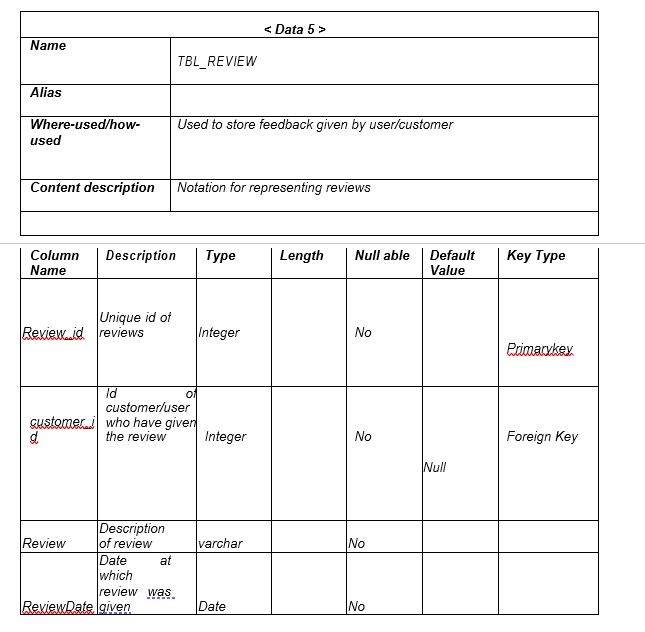
**Data Dictionary:**











**Application Design:**

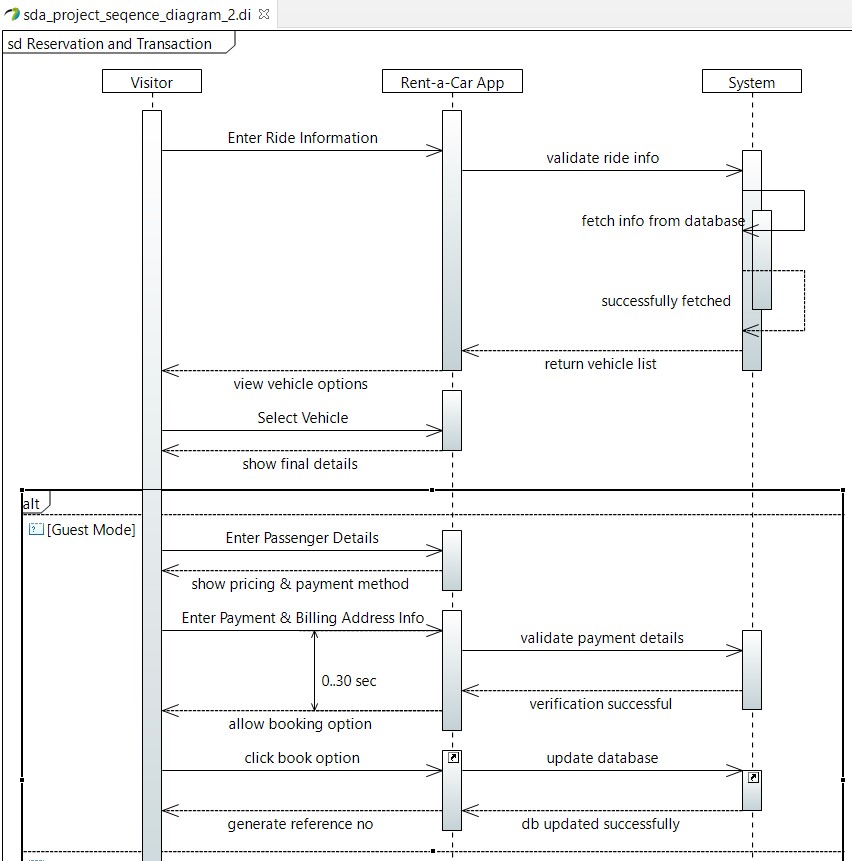
*Guests can browse the available cars, view rental prices, and check basic information about the rental services without the need to log in. They can explore the website's features and get an overview of the services offered by the Hash Limo Company.*

*Hosts need to create an account on the platform to list their cars for rent. They can upload information about their vehicles, set rental prices, specify availability, and manage bookings. Hosts may also have the ability to update details about their cars, such as maintenance records and photos.*

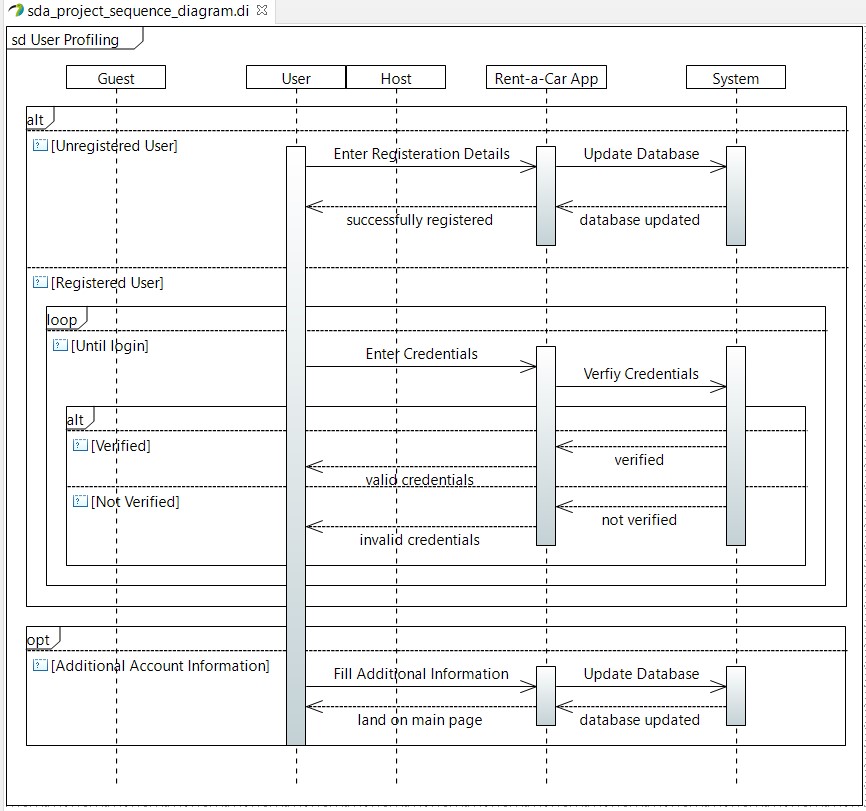
*The system manages user authentication and authorization to distinguish between guests and hosts. The system also handles the storage and retrieval of car-related information, including availability, rental rates, and host details. Moreover, it facilitates the booking process by coordinating communication between guests and hosts. Implements security measures to protect user data and financial transactions, and generates reports and analytics for both guests and hosts, providing insights into usage patterns, revenue, and other relevant metrics.*

*Guests can become hosts by creating an account, providing necessary information, and listing their cars for rent. Guests can browse the available cars, select one for rental, and initiate the booking process. The system verifies the availability of the selected car, calculates the rental cost, and facilitates the booking transaction. Hosts receive notifications about bookings, confirm reservations, and coordinate the pickup/drop-off details with guests. The system ensures the secure handling of payments, maintains a booking history, and provides support for dispute resolution if necessary.*

**Sequence Diagram (Behavior Model):**



*A visitor arrives at the webpage and feeds the ride information to the web application which is then in return verified by the system by fetching the information from the database and the list of vehicles is returned. The visitor will then view the options available to select the vehicle from. He then selects the vehicle and the final details are displayed to them through a GUI. For booking purposes, the guest will enter the passenger details and an invoice will be generated. The guest enters the credentials for the credit card. There is a 30 second time window acting as the captcha meanwhile, the system verifies the provided credentials at the backend server. If the approval is received, the database is updated and the ride is booked successfully.*



*The user profiling section presents a more advanced overview of the system in general. The guest and host timeline are not as interactive. This diagram is a basic representation of the functionalities occurring at the backend such as verification of the registration details, credentials and updating of the database in case of additional information being received*

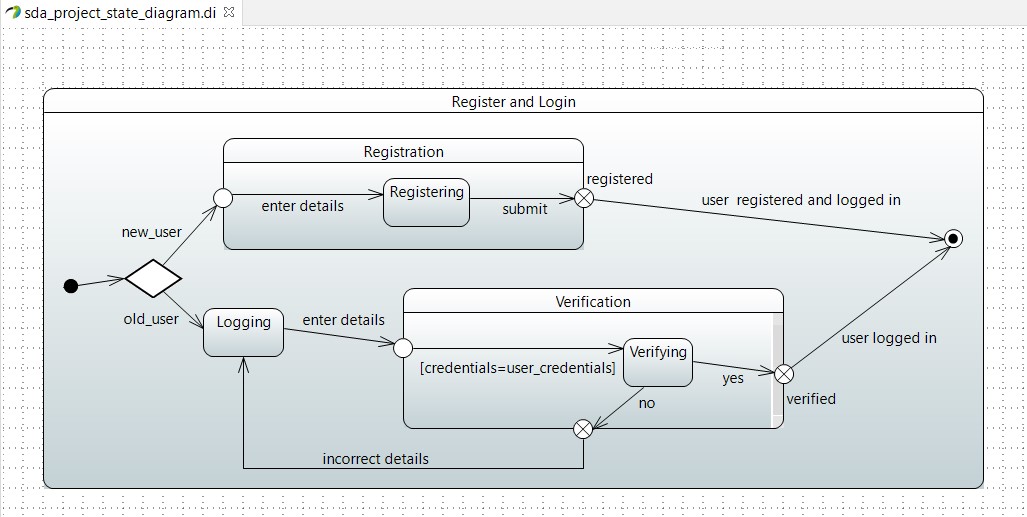
*.*

**Collaboration Diagram:**

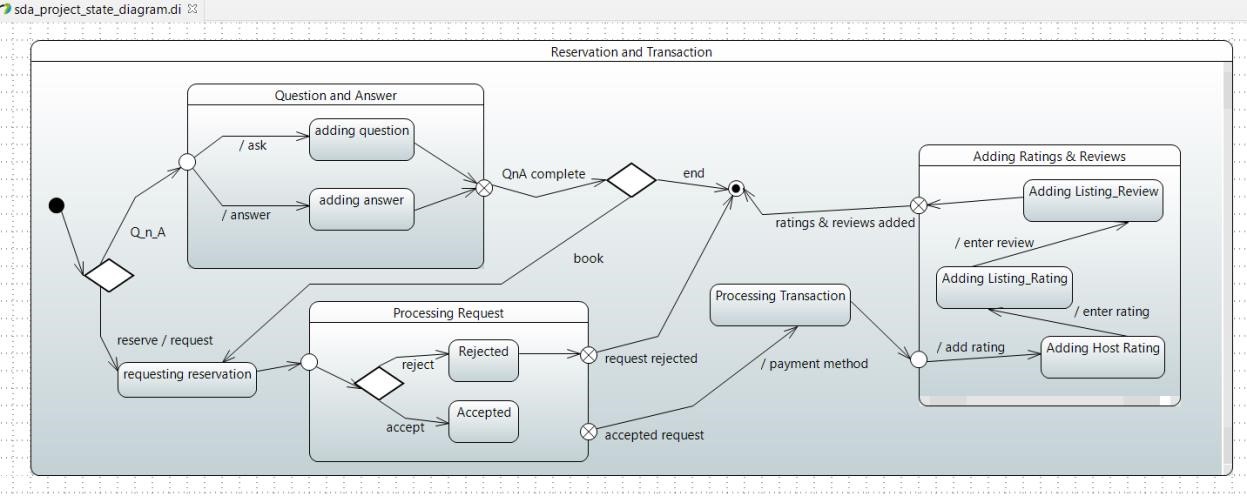
*The collaboration diagram shows 4 actors in guest, user, system and the host itself. The functionalities and features that can be availed by the following is depicted through the functions listed beside them. The numbering before the name of the function represents the flow of the website and the order in which they may be accessed.*



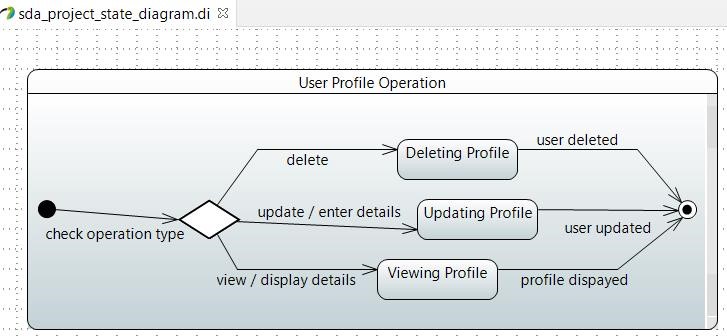
**State Diagram (Behavior Model):**



*The state diagram above refers to the register and login scenario. When a user first arrives at the website, he may be new to the system or a regular customer. In case of a new user, he first has to follow the signup procedure and register with the system. Whereas, the already registered visitor will have to log in to the system, where his provided credentials are verified. If the credentials entered were correct, the user will be given access to the features of the web application. Or else he will be redirected to the login page to reenter the correct credentials.*

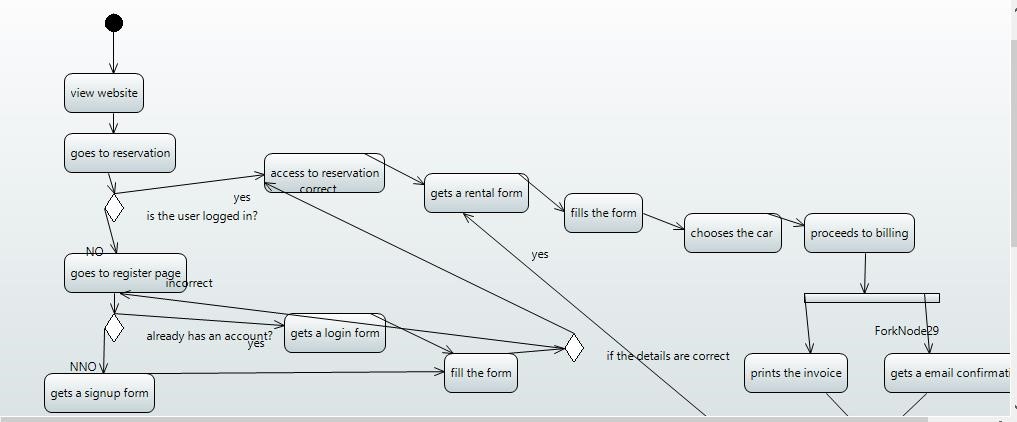


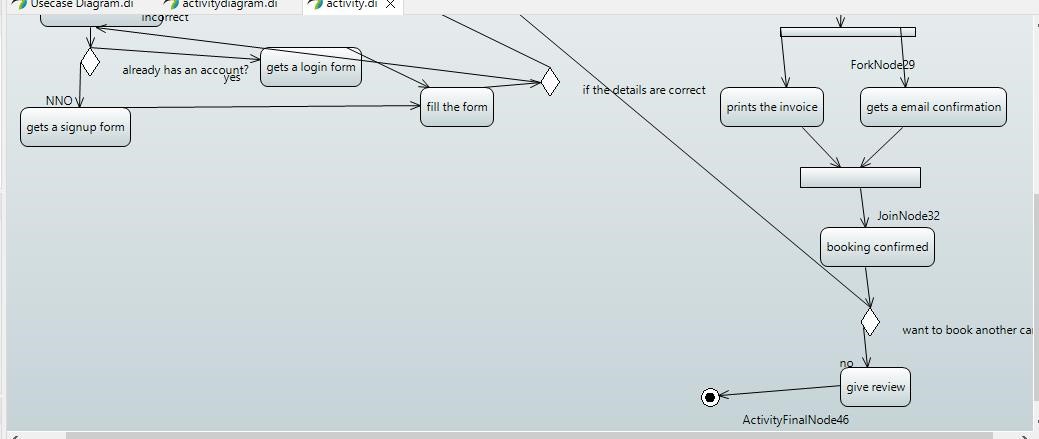
*Before reservation, a customer may have questions and answers regarding the services provided, such as is the desired car available, so a separate page caters to this need of the clients. If the customer is satisfied and wants to make a reservation, he sends a request, if the request is rejected, he cannot book a ride due to the website’s safety protocols. However, if the request is processed and approved, the transaction is made by the customer to receive a confirmation of the booking via email. In the end before leaving the system, they are required to leave a feedback through reviews and rating the overall experience.*



*The above state diagram is a simple model of the operations the system undergoes at the backend when a user requests to delete, to update or to view their previous entered profile information while setting up their accounts.*

**Activity Diagram (Flow-based Model):**





*A visitor to the website first arrives at the reservation page where he is asked to either login or signup to the system to continue. If the user is not registered, he will have to fill up the signup form and then is granted access to the reservation page. If the user was already registered with the system, he logs in and is directed to the reservation page. The user is then specified to fill the rental form which includes details such as selecting the car required by the user. He is then navigated to the billing page where an invoice is generated. After the user completes the transaction, he receives a confirmation email for the booking. Later on, if the customer wants to book another car as well, he is redirected to the make reservation page; else, he is sent to the page where he will leave the review for the experience.*

1. **Appendices:**

*The table of contents is designed to provide detailed information and supplementary materials for the Hash Limo Car Rental Management System. Users are encouraged to refer to this appendix for a comprehensive understanding of the system's architecture, functionality, and guidelines. Additionally, the appendix will be periodically updated to reflect any changes or improvements made to the system.*