SOFTWARE DESIGN SPECIFICATION

“LIMITER”

THE SMART PARKING SYSTEM

1. Introduction

The purpose of this Software DesignSpecification SDS document is to provide a detailed overview of our Smart parking system software, its parameters and goals. The product whose software requirements are specified in this document is an android based simulation system that locates and tracks available parking spots near CC3 building in IIIT Allahabad campus.

1.1. PURPOSE

The ever-increasing traffic has enforced the use of Smart Parking Systems. This application will aid in the optimization of parking spaces, regulation of traffic flow, and improve the efficiency of parking operations. Well this document gives the overall design of our application. This document has state diagrams, sequence diagrams and class diagrams and pseudo codes which will help the developers to get a detailed knowledge of our project and its different applications.

From Driver’s point of view :-

* Driver’s are unknown whether the slot of parking is available or not. So they need to travel the full area and if all slots are occupied then a traffic like situation is created.
* It's difficult to see the lines of parking slots and sometimes people on purpose or by mistake park their car in a wrong way which also occupies another slot.
* Few driver’s leave their car/bike for a prolonged period.

From Admin’s point of view :-

* The management is unaware of the status whether a vehicle is parked properly or not.
* It's difficult to keep a record of all complaints.
* When a vehicle obstructs another vehicle it leads to arguments and complaints.
* It is hard to keep track of people entering and leaving the building.
* The trivial parking system takes a lot of space due to improper alignment of vehicles.
* Less security is given to the parking spaces.

This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

1.2. Document Conventions

MAIN SECTION TITLES

Font: Arial Face: None Size: 20

Subsection Titles

Font: Arial Face: None Size: 16

Standard Text

Font: Arial Face: None Size: 12

1.3 Intended Audience and Reading Suggestions

The audiences for this document include the system developers and the users. The system developer uses this document as the authority on designing and building system capabilities. The users review the document to ensure the documentation completely and accurately describes the intended functionality.

This version – version 1.0 - provides general descriptions of the system. The system developer should review the document to ensure there is adequate information for defining an initial design of the system. The users should review the document to affirm the features described are needed, to clarify features, and to identify additional features needed within the system.

1.4 SCOPE:

LIMITER will allow the person to save their time in hectic and trivial parking, thus allowing a person to do something more productive, which makes them happy and mindful. We have mentioned briefly the features that are In-scope and Out-of-Scope using our application :

In-Scope:

* Efficient parking space division as the sensors will guide the driver to park the vehicle properly.
* The driver can update the extra amount of time he will be occupying the parking space in the application so that the new user is updated with the waiting time.
* It assists the user in finding a parking spot in a matter of minutes.
* With the Smart Parking System, it is almost impossible to steal a vehicle. Thus, giving extra security.
* There is a good chance for the accidents to reduce, considering that there will be fewer distracted drivers on the road looking for open parking spaces. Thereby improving road safety.
* The Smart Parking System helps the management to have a record of people entering and leaving the building.
* It also results in fewer parking complaints.

Out-of-Scope:

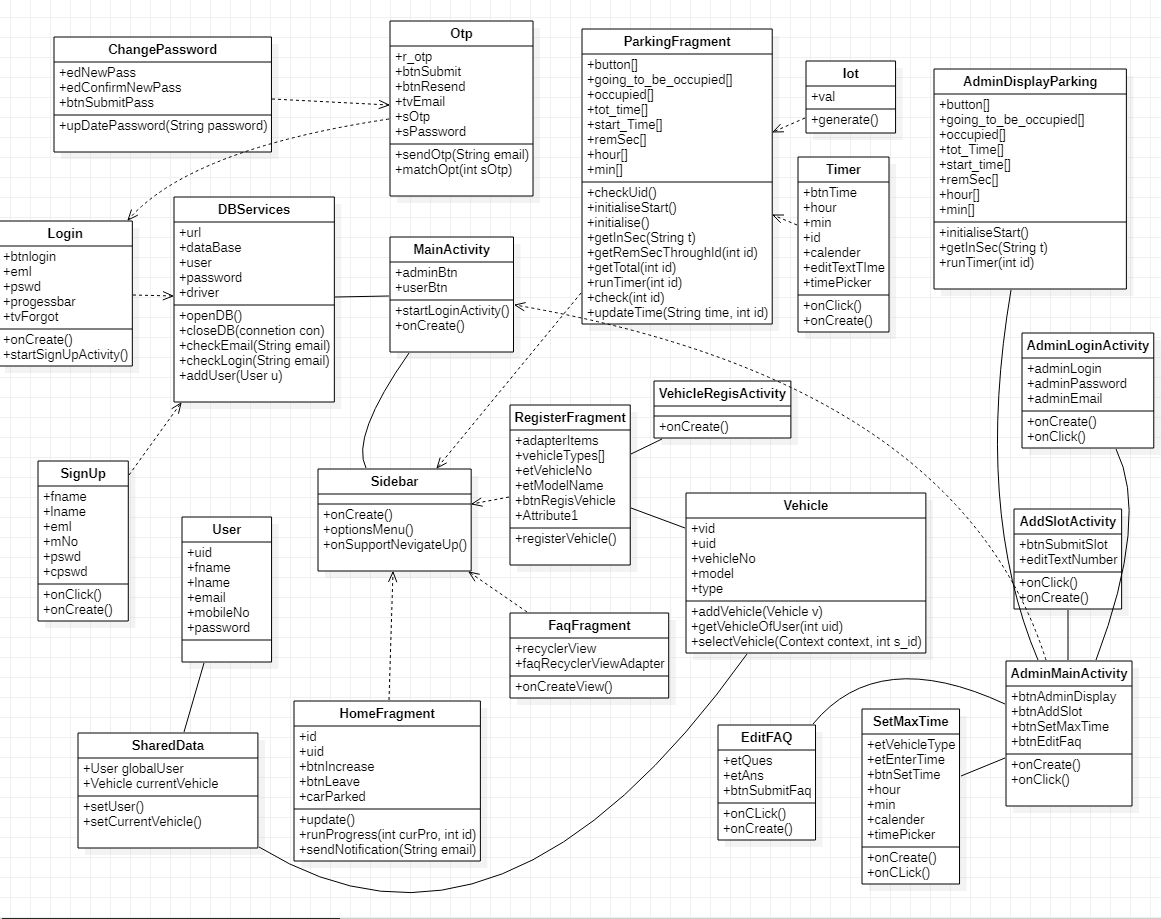
* Restricting the parking time.
* Always finding an empty parking spot.

1.5 REFERENCES:

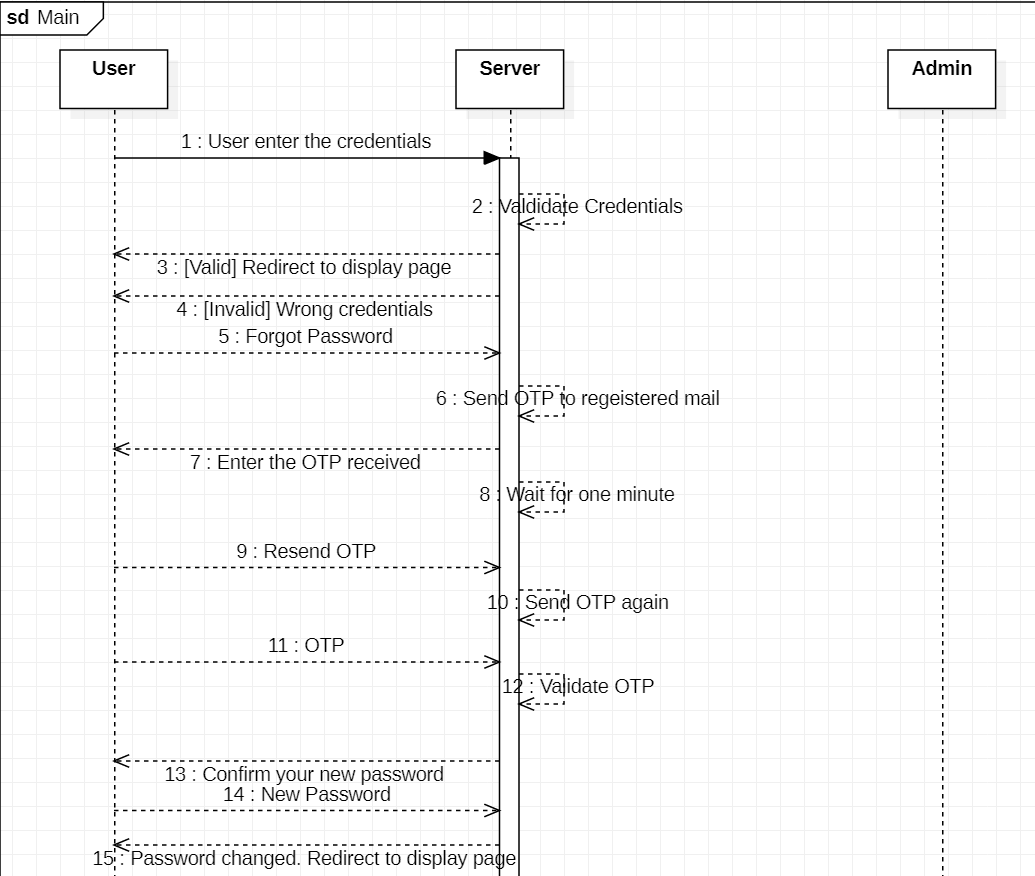
IEEE SDS Format.

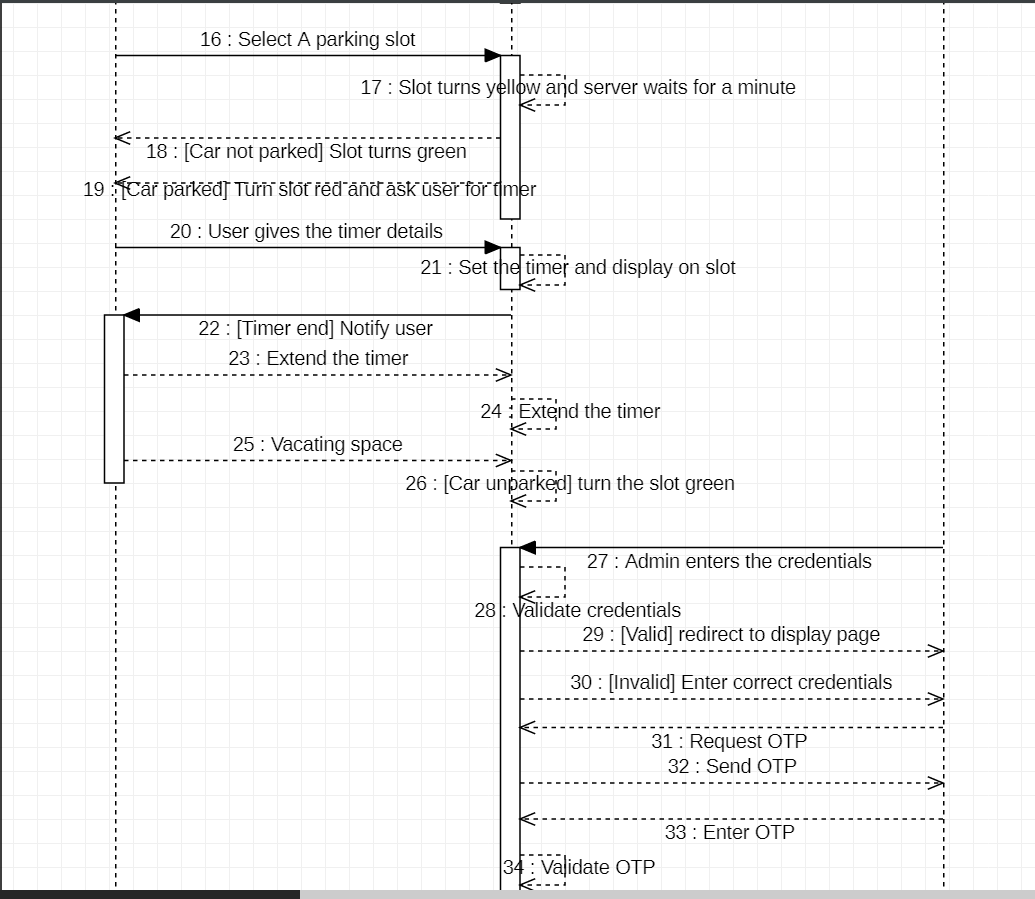
2. Logical Architecture (Class Diagram, Sequence Diagram, State Diagram)

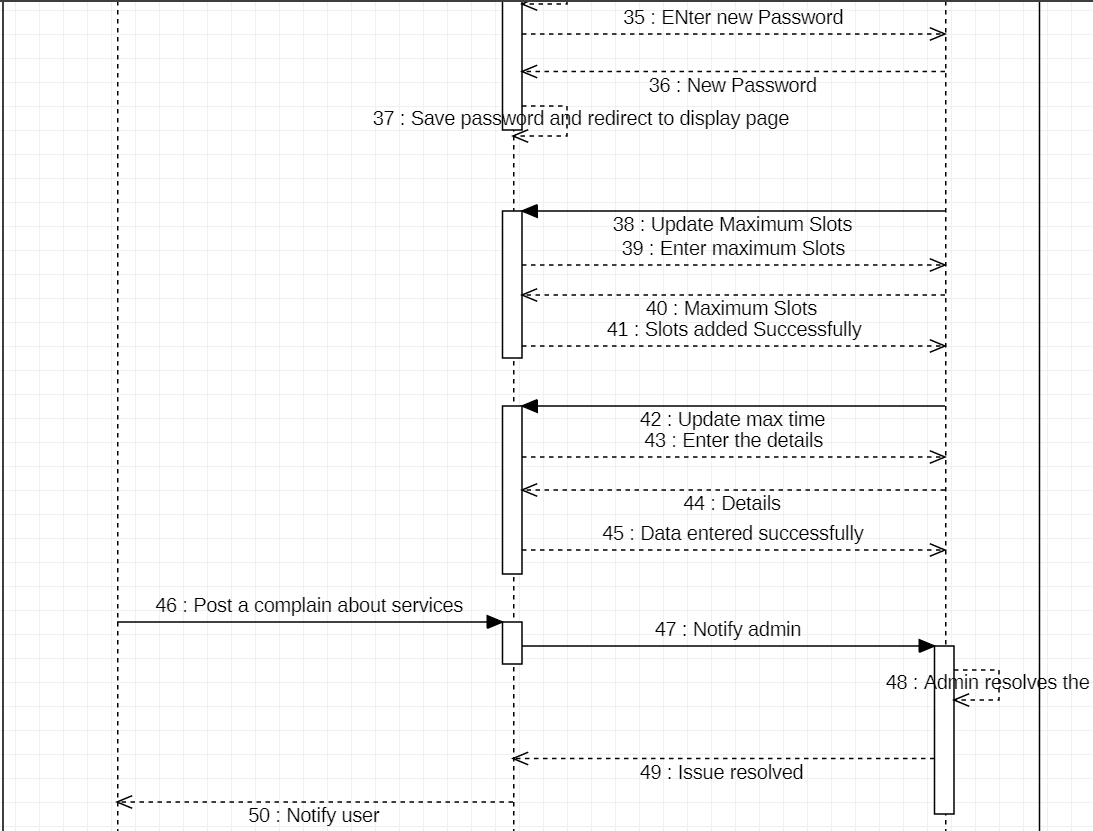
2.1.1.Class Diagram :

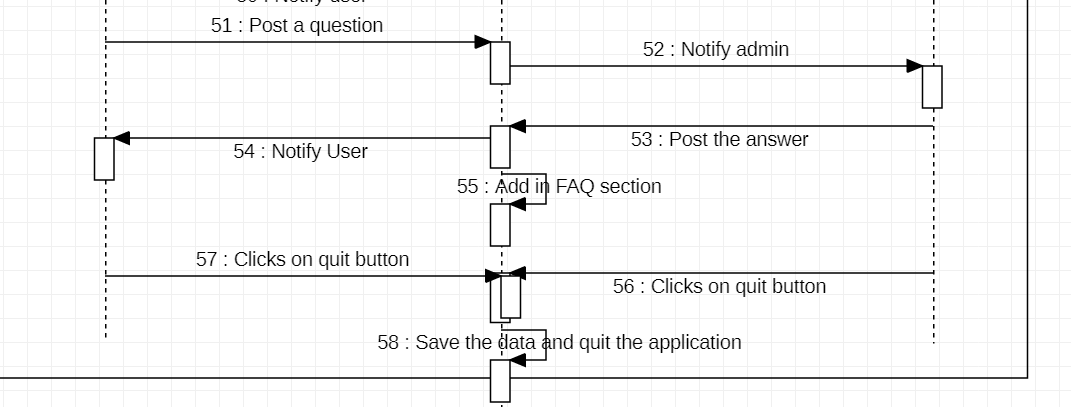


2.1.2. Sequence Diagrams :

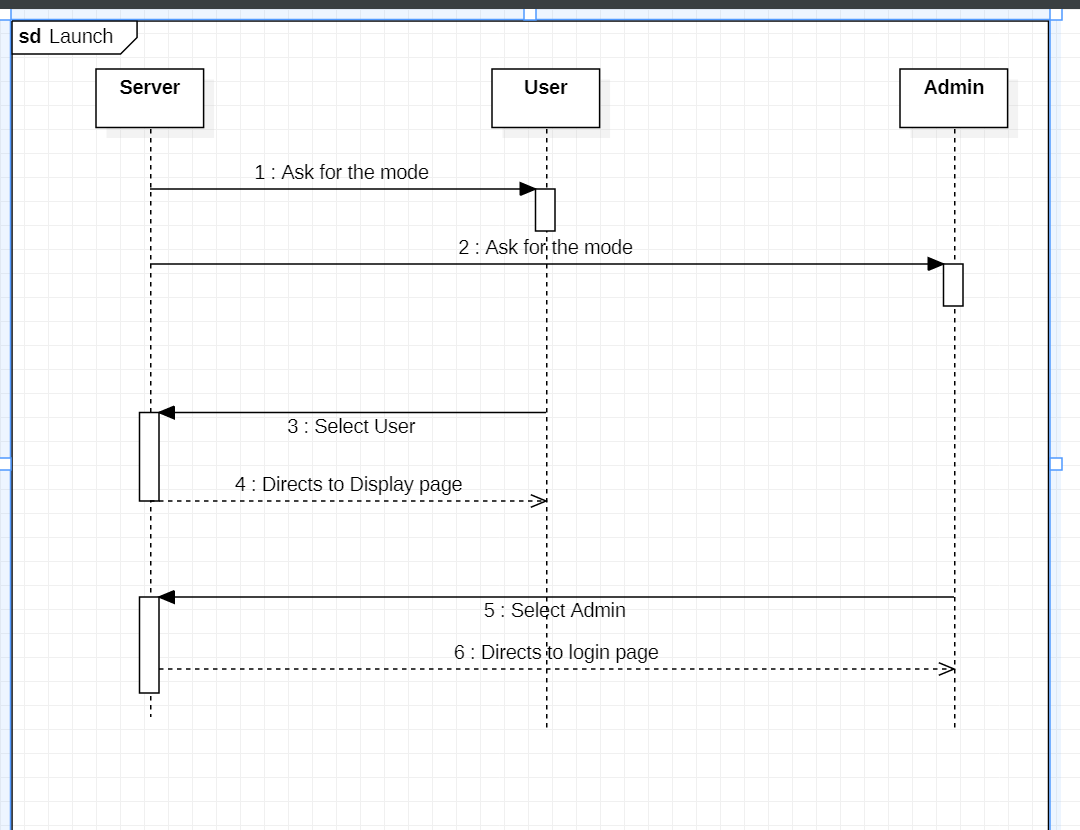




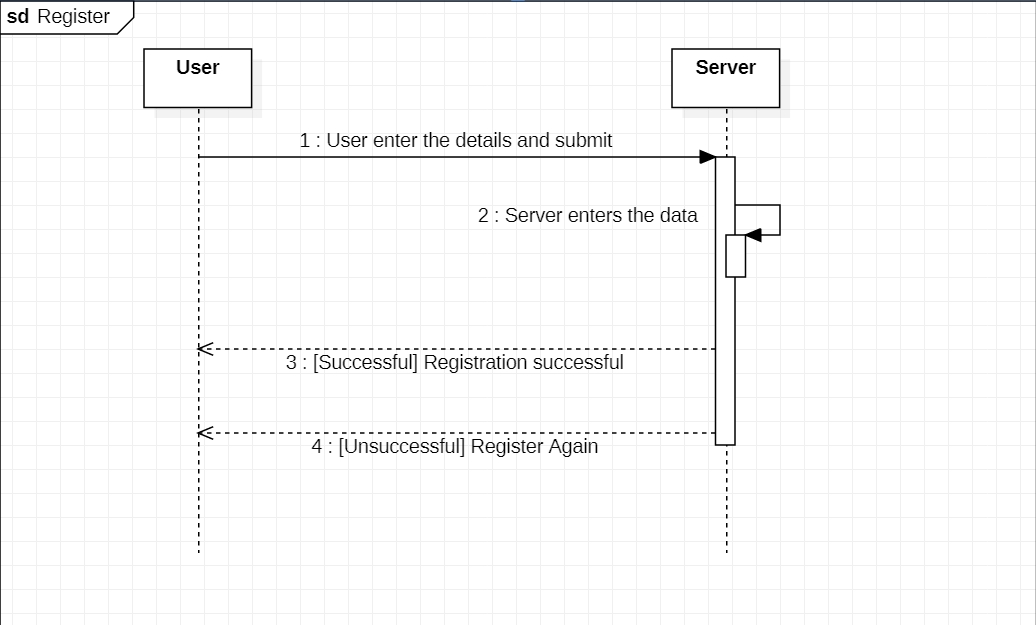




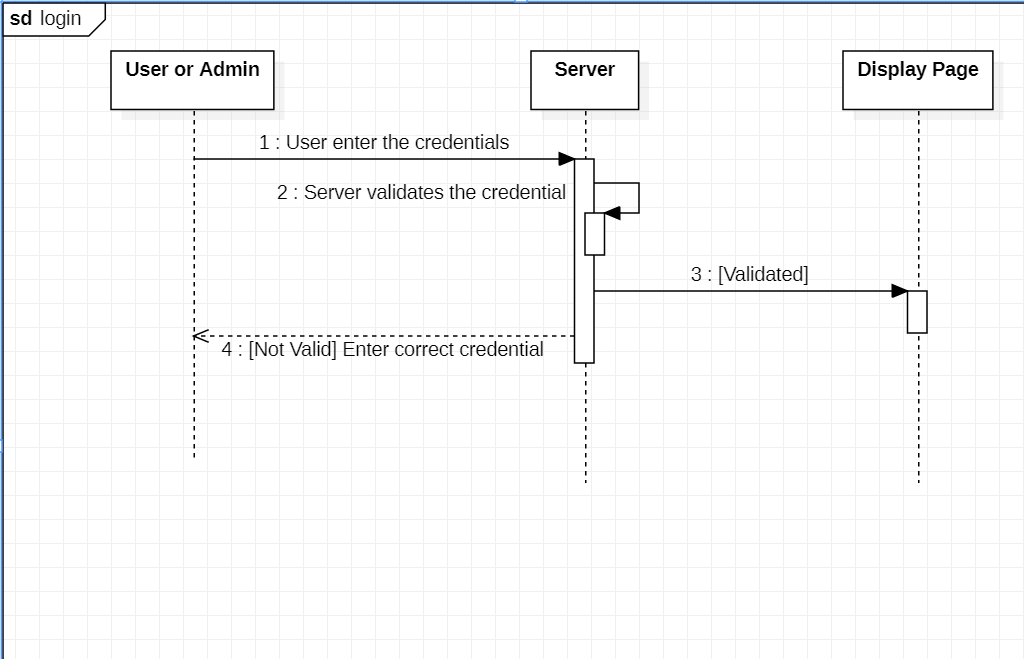
2.1.2.1 : Sequence Diagram : Launch



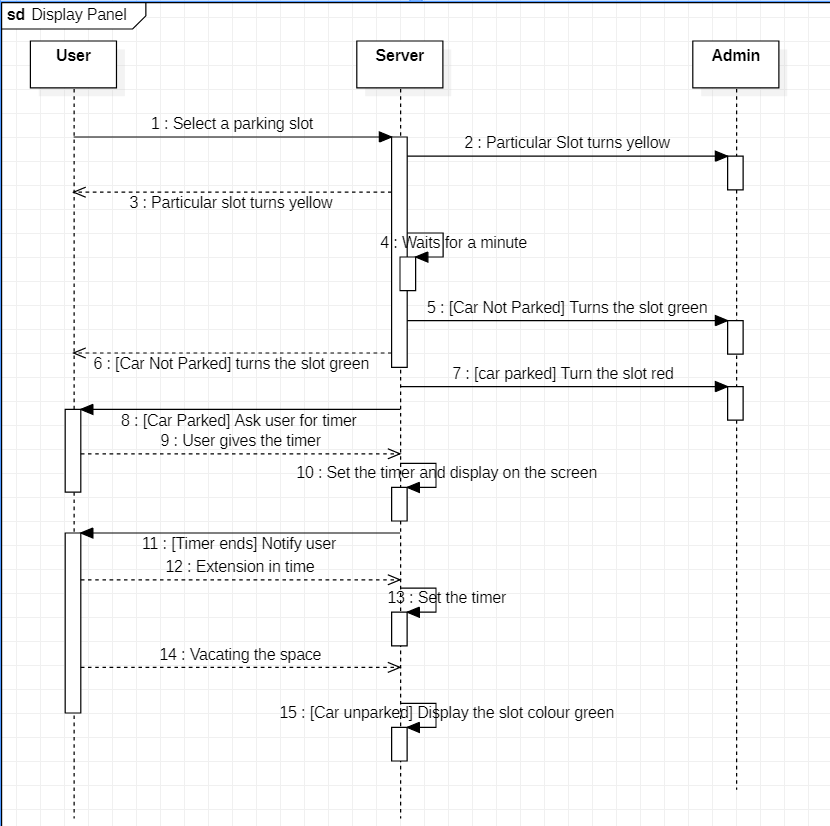
2.1.2.2 : Sequence Diagram : Register



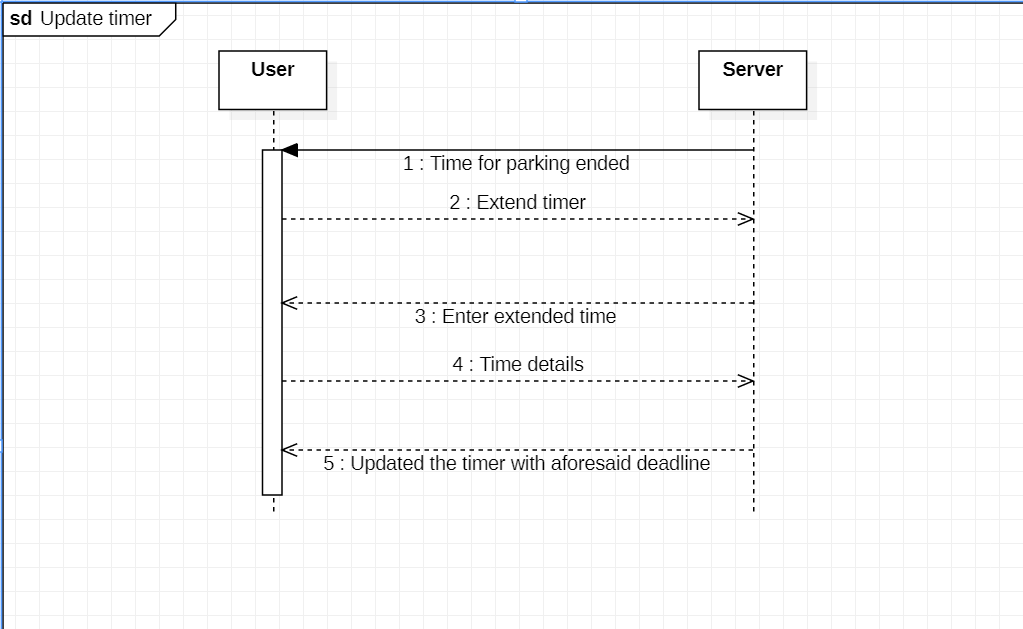
2.1.2.3 : Sequence Diagram : Login



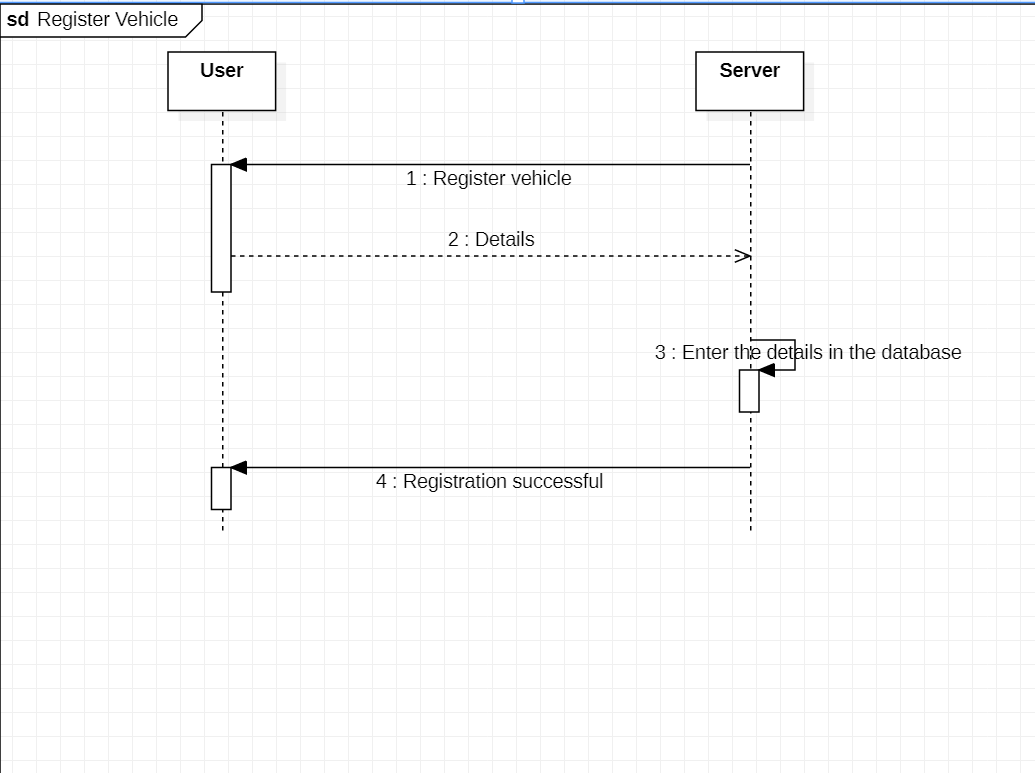
2.1.2.4 : Sequence Diagram : Display



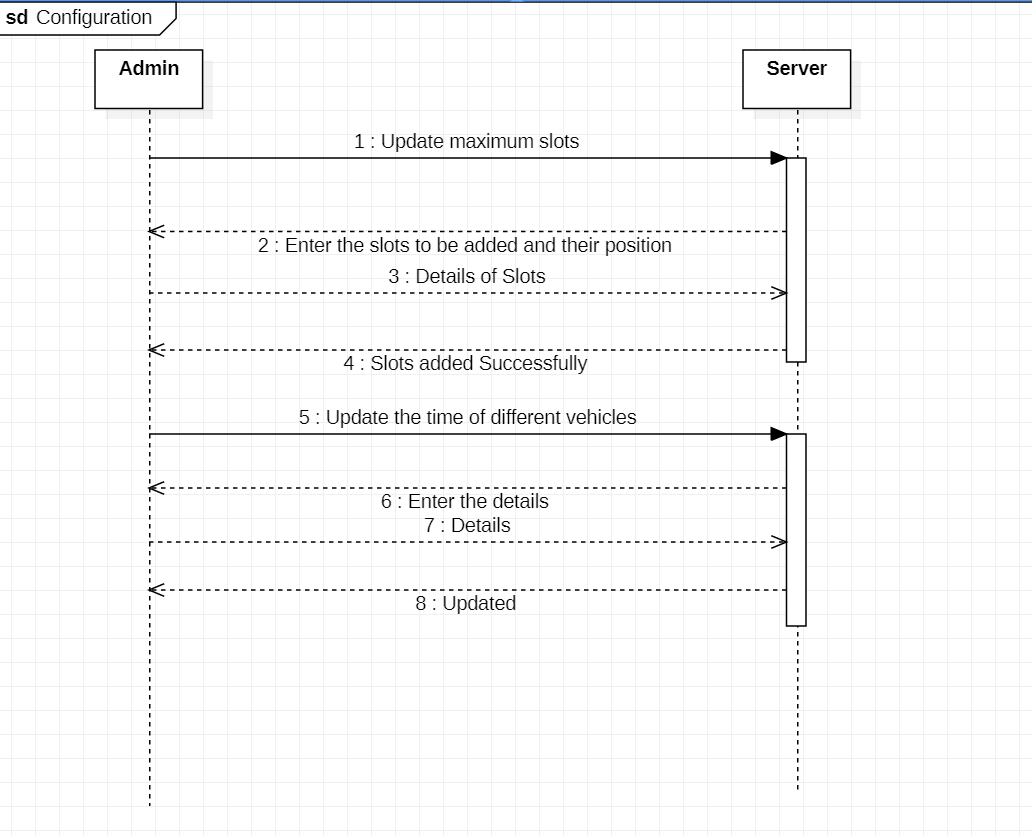
2.1.2.5 : Sequence Diagram : Update timer



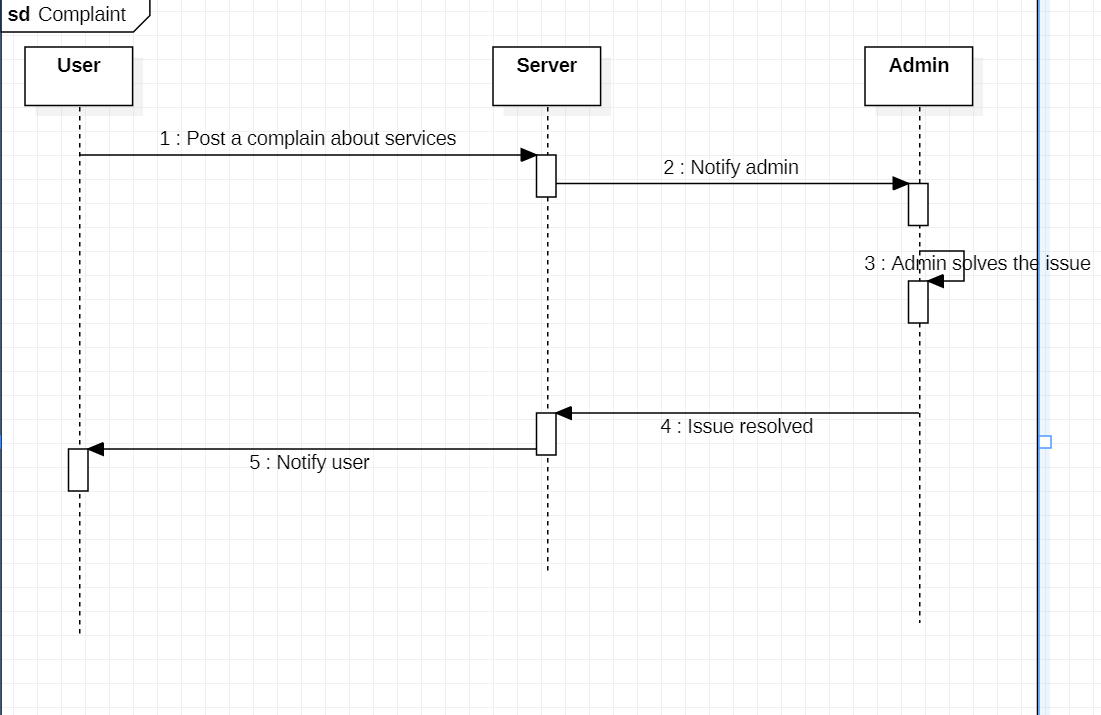
2.1.2.6 : Sequence Diagram : Register Vehicle



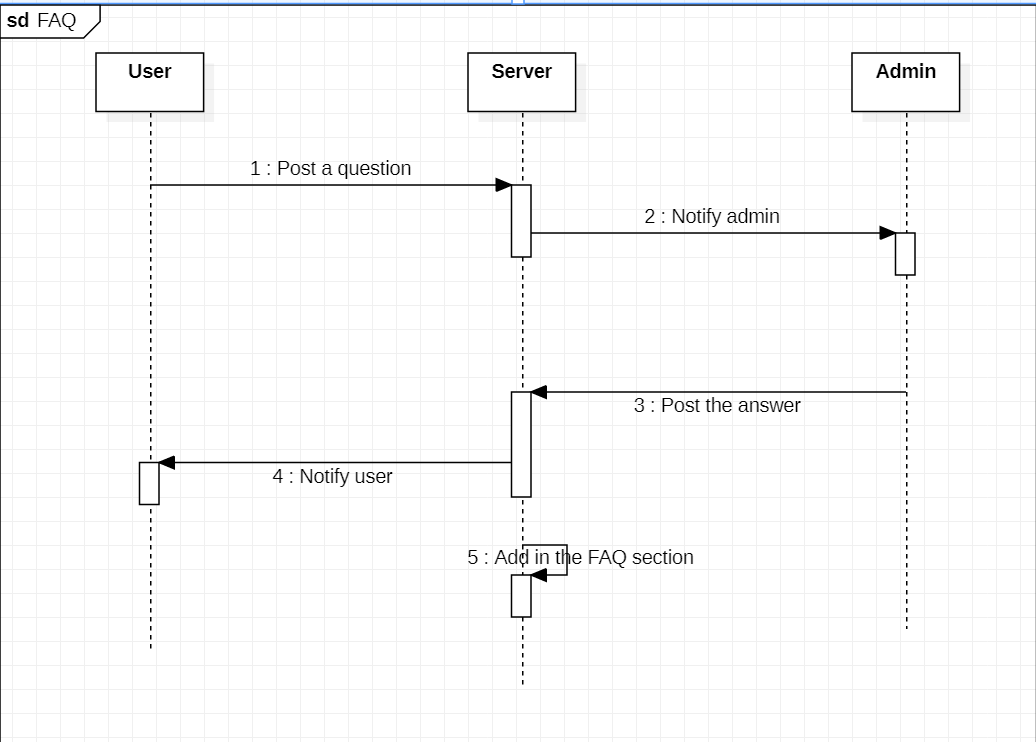
2.1.2.7 : Sequence Diagram : Configuration



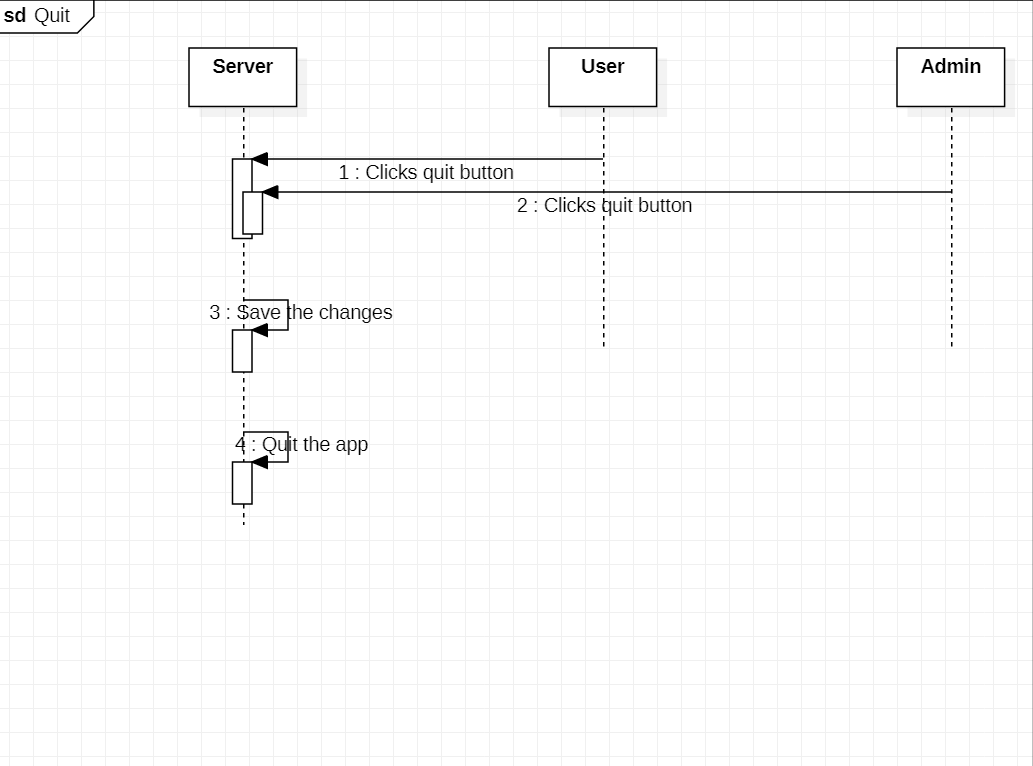
2.1.2.8 : Sequence Diagram : Complain



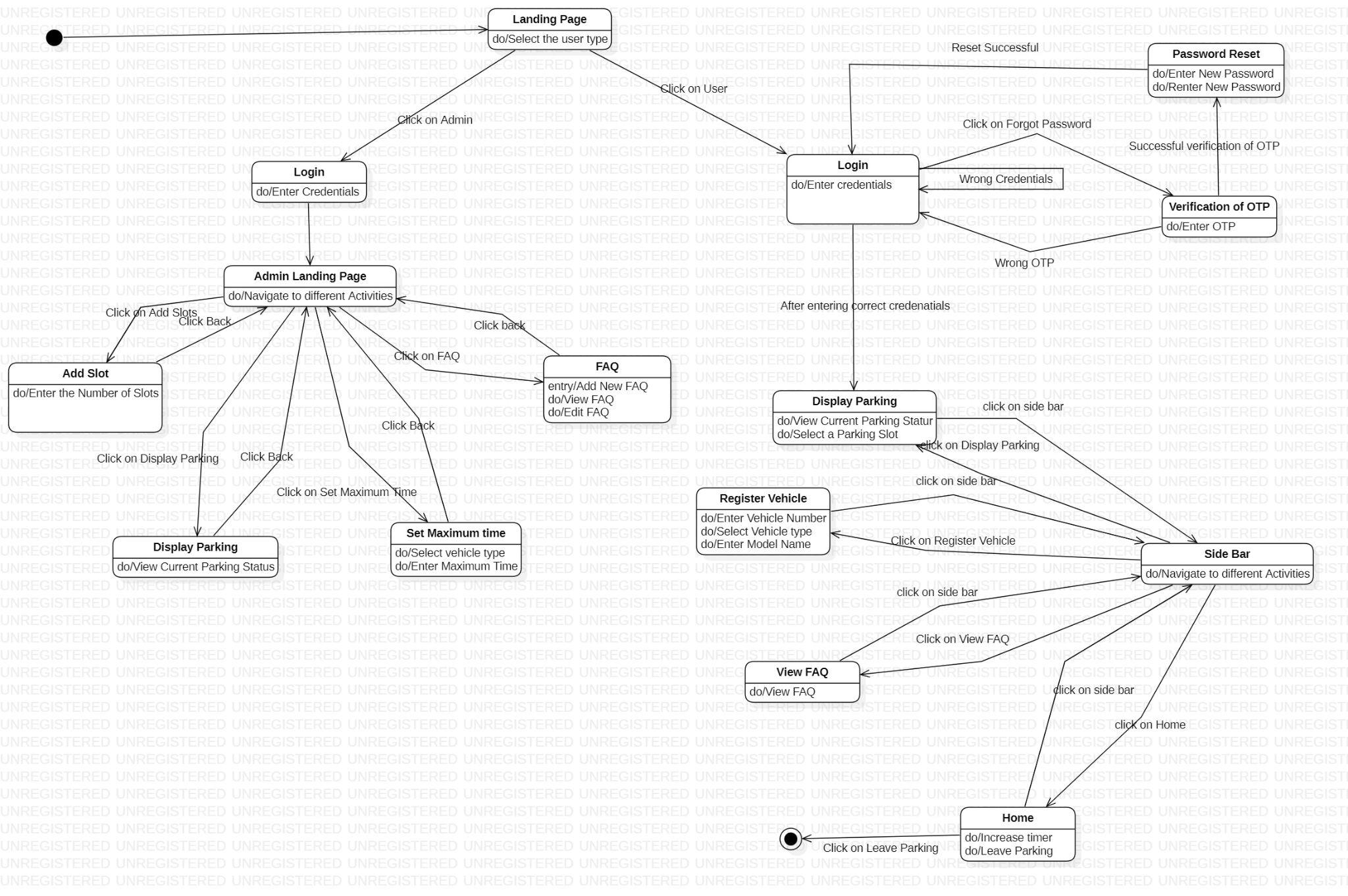
2.1.2.9 : Sequence Diagram : FAQ



2.1.2.10 : Sequence Diagram : Quit



2.1.3. State Diagrams



Above diagram clearly shows different states of the given project “Limiter” and different activities which a user can / has to perform if he/she is in a particular state.

2.2. Logical Architecture Description

2.2.2. Sequence Diagram Explanation

2.2.2.1. **Launch :** User/ Admin launches the application and gets two options. He chooses the mode he wants to enter and is redirected to the display page if he is a user , else to the login page if he is admin.

2.2.2.2. **Register :** Users can use this to register themselves beforehand to save time. They click on the register button or are forced when they park their vehicle. Server asks for the details , and the user enters them. Server then enters them in the database and if it is updated successfully it sends back the message registration successfully else asks the user to register again.

2.2.2.3. **Login :** When user and admin are in the login page, the server requests them to enter their credentials. Server checks the credentials in the database and if it is present it directs them to the display page but if the entered credentials are wrong serven asks them to enter the correct credentials.

2.2.2.4. **Display :** This is the main page of the whole application. Whenever a user wants to park his vehicle, he selects a slot which turns yellow. Server waits for 1 minute and if the user is able to park in that time limit the server turns the slot red else it turns it green. If the park is successful, the server asks for a timer which is given by the user and the server displays the timer on the display page.

2.2.2.5. **Update timer :** When the server notifies the user to vacate space he can extend the timer by choosing that particular option. Once chosen, the server asks for the extension in the timer and the user sends the details which are finally updated by the server.

2.2.2.6. **Register vehicle :** Once a user has parked his car he is prompted to register his vehicle and he sends the details which the server updates or saves in the database and displays the registration vehicle successful page.

2.2.2.7. **Configuration :** Once admin is in the configuration page he gets many values which can be updated. He sends a message to the server regarding what he wants to update. Server asks for values and the admin enters them which is updated in the application. This way admin can update maximum slots and the maximum time per vehicle.

2.2.2.8. **Complaint :** User can post their complaint which will be notified to the admins and admins can resolve these complaints. Once resolved user is notified about this.

2.2.2.9. **FAQ :** Users can post their question in the FAQ section. Admin will be notified about the question and he can answer those questions.

2.2.2.10. **Quit :** When the user/admin wants to quit the application, they can click on quit and the application saves the changes and quits the application.