# Design

Design, the third phase of software development life cycle. The objective of this phase is to transform business requirements identified during previous phases, into detailed system architecture which is feasible and valuable to organization. This system will provide the benefit for the customers, car owners and Car Rental Organization. Currently available system is providing the rental process between the customers and CarRental Organization. The objective of this phase is to transform business **requirements** identified during previous phases, into a **detailed**system architecture which is **feasible, robust** and brings **value**to the **organization.**

## 3.1 – Structural Design

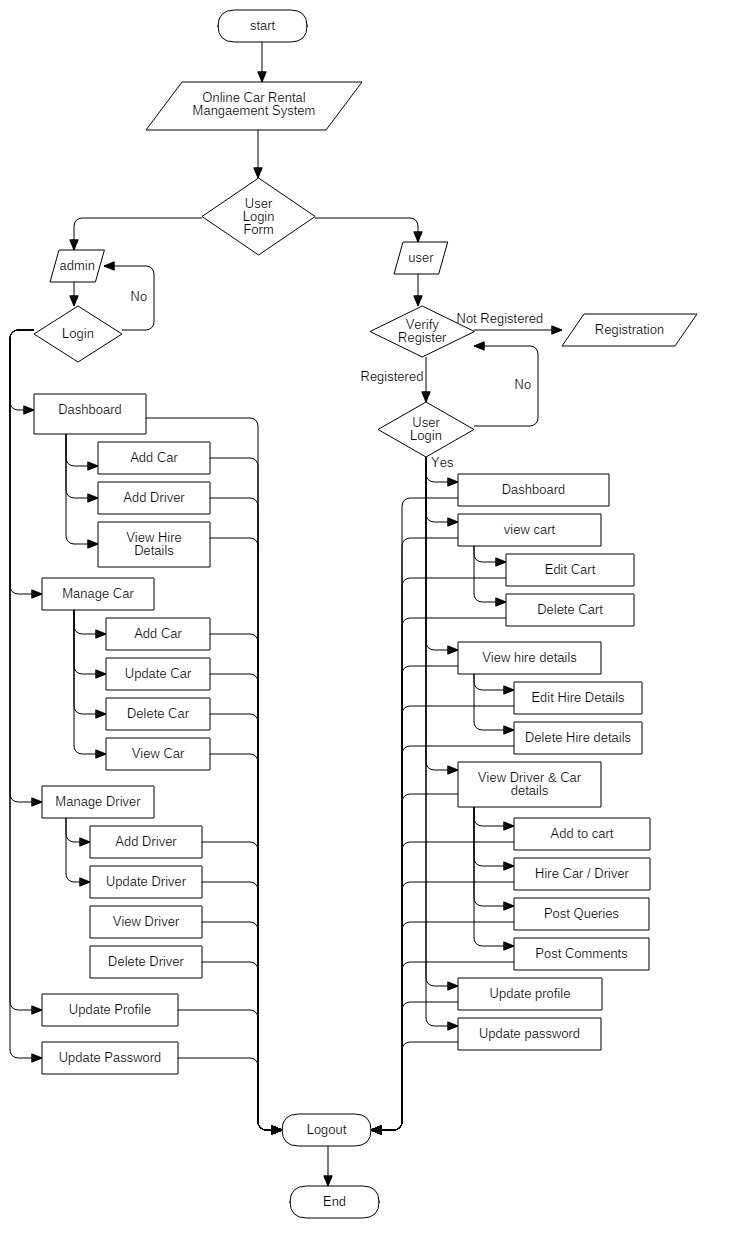
Design phase is third phase in SDLC model. This phase details and describes necessary features, operations and specification that will justify the proposed functional requirements. The design constructed in this phase will be later implemented in the implementation phase. The system is graphically represented for better understanding and sight of the context and the system. It helps to build architecture of the system. Once the designs are approved it is documented and directed towards development. There are many design models available, for this project I will be using the models are: Structural Model, Behavioral Model, Database Model, UI design.

### 3.1.1 – Class Diagram

### 3.1.2 – Flow chart

Flow chart is a diagram which illustrates the sequential process which is to be performed to get the solution to problem. In each step of the process it is shown by a different symbols and short description of the process step. It can be easily understanding by any people the movement of flow.

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Notation Used** | **Notation Name** | **Description** |
| 1 | Terminal symbol in flowchart of programming | Start/end | It represents the start and end point of the flow. |
| 2 | Flowline symbol in flowchart of programming | Flow Line | It shows the relationship between the representative shapes |
| 3. | Input/Output symbol in flowchart of programming | Input/output | It represent the input or output. |
| 4. | Processing symbol in flowchart of programming | Processing | It represent the process. |
| 5. | Decision making symbol in flowchart of programming | Decision | It decides the true and false result. |



Justification of creating Flow chart in my project are:

1. Helps for problem solving.
2. Helps for effective analysis.
3. Helps in coding phase.
4. Helps to understand the workflow.

## 3.2 – Behavioral Modeling

Behavioral Modeling is the describes the interaction in the system. It shows the dynamic nature of the system. It illustrates the interaction of the structural diagrams.

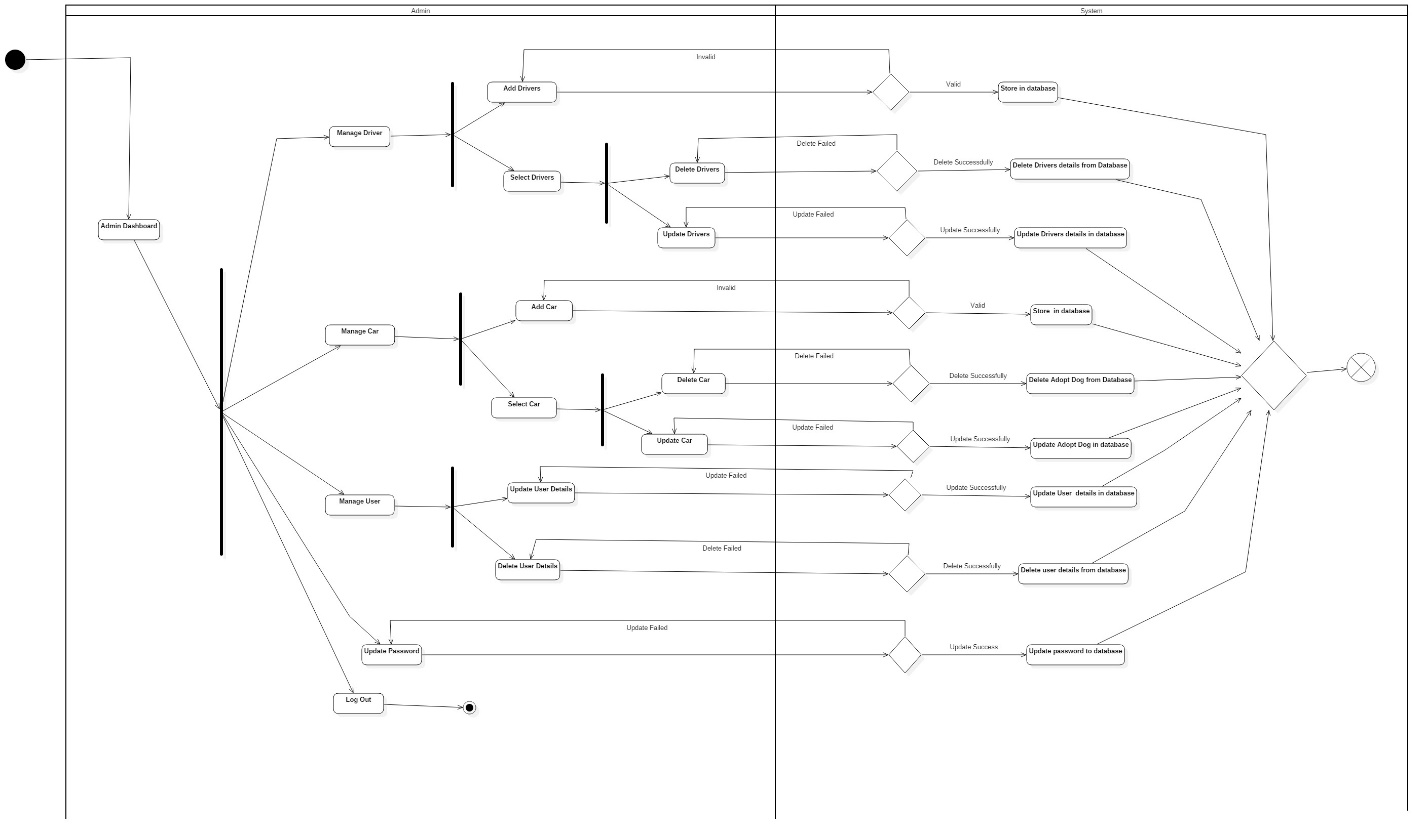
There are many types of behavioral modeling. They are listed below:

* State diagram
* Sequence diagram
* Activity diagram, etc.

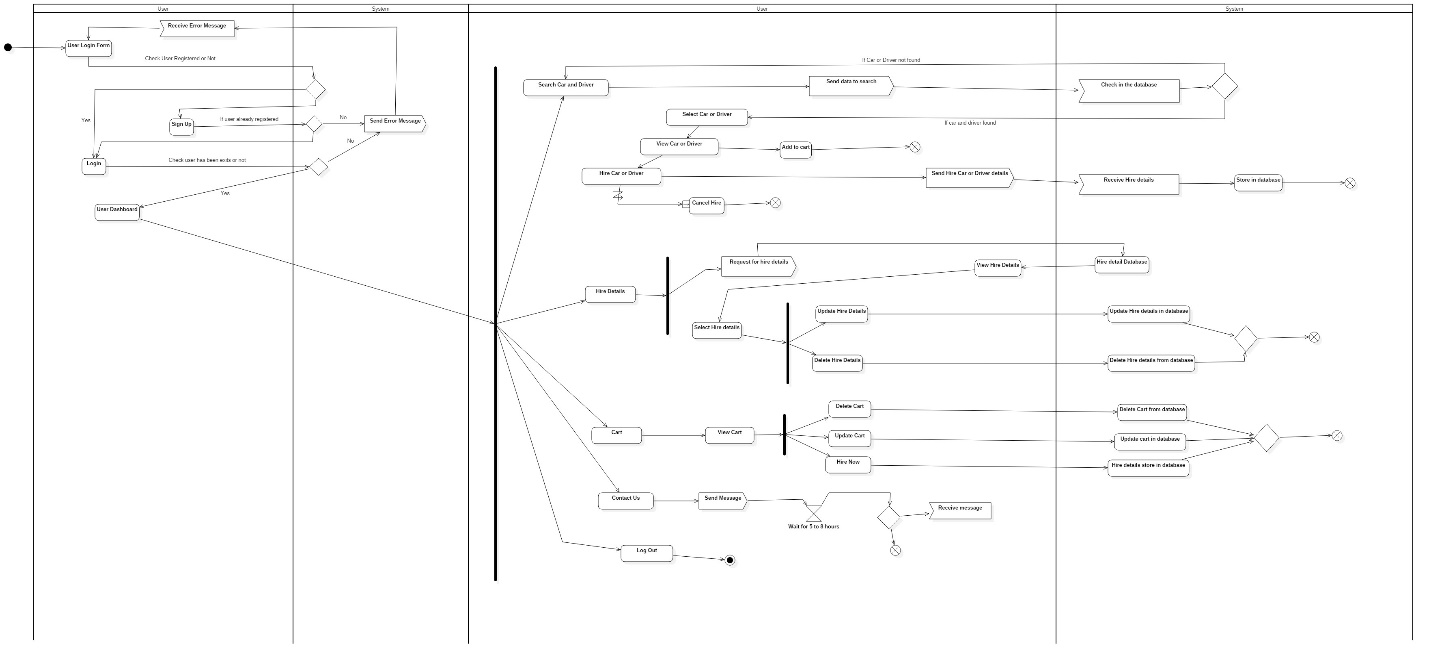
### 3.2.1 – Activity Diagram

Activity diagram is diagrammatically representation of the flow of activity from one activity to another. It is the most important UML diagram to explain the dynamic behavior of the system. The activity is also known as the operation of the system.

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Notation Used** | **Notation Name** | **Description** |
| **1.** |  | Initial State | Helps to start the flow. |
| **2.** |  | Sequence Flow / Control Flow | Helps to show the transitions from one action state to another. |
| **3.** |  | Activity | Helps to represent a set of actions. |
| **4.** |  | State | Helps to define current condition of an event. |
| **5.** |  | Decision | Helps to test true or false condition. |
| **6.** |  | Synchronization | Split single behavior into a set of parallel flow of activities. |
| **7.** |  | Time Event | Helps to define the waiting or end period of the operation. |
| **8.** |  | Merge | Helps to bring together multiple flows that are concurrent. |
| **9.** |  | Message Sent | Helps to send some message to the system. |
| **10.** |  | Message Receive | Helps to receive message from the sent signal. |
| **11.** |  | Final State | Helps to indicate the end of flow |

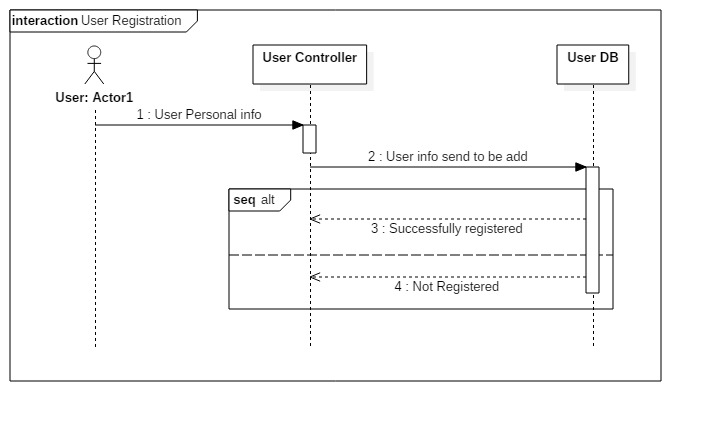
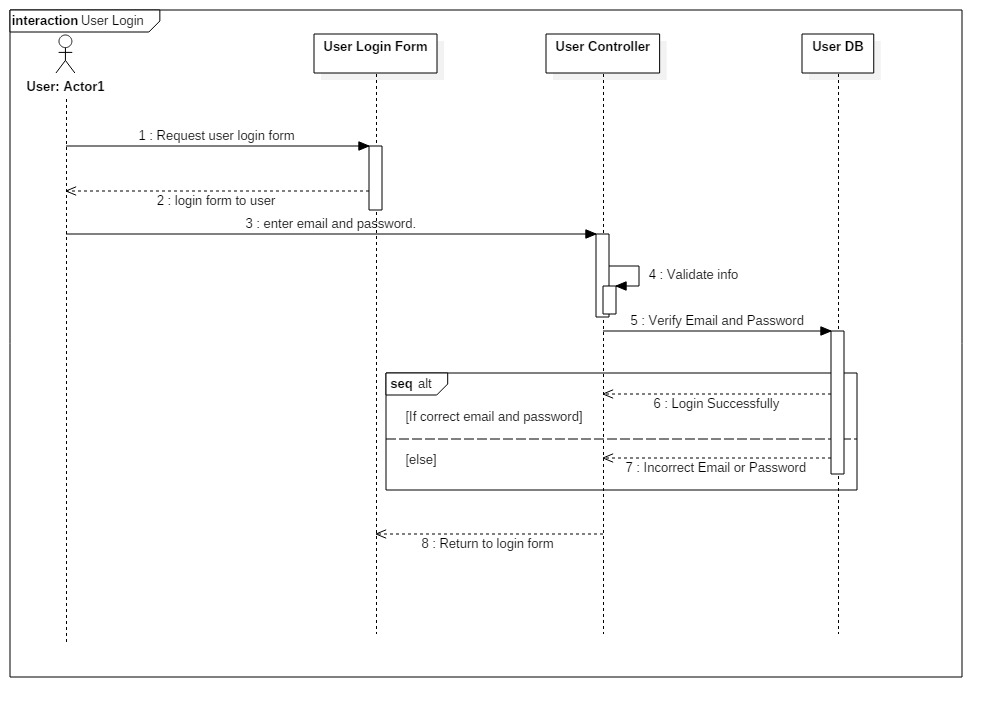
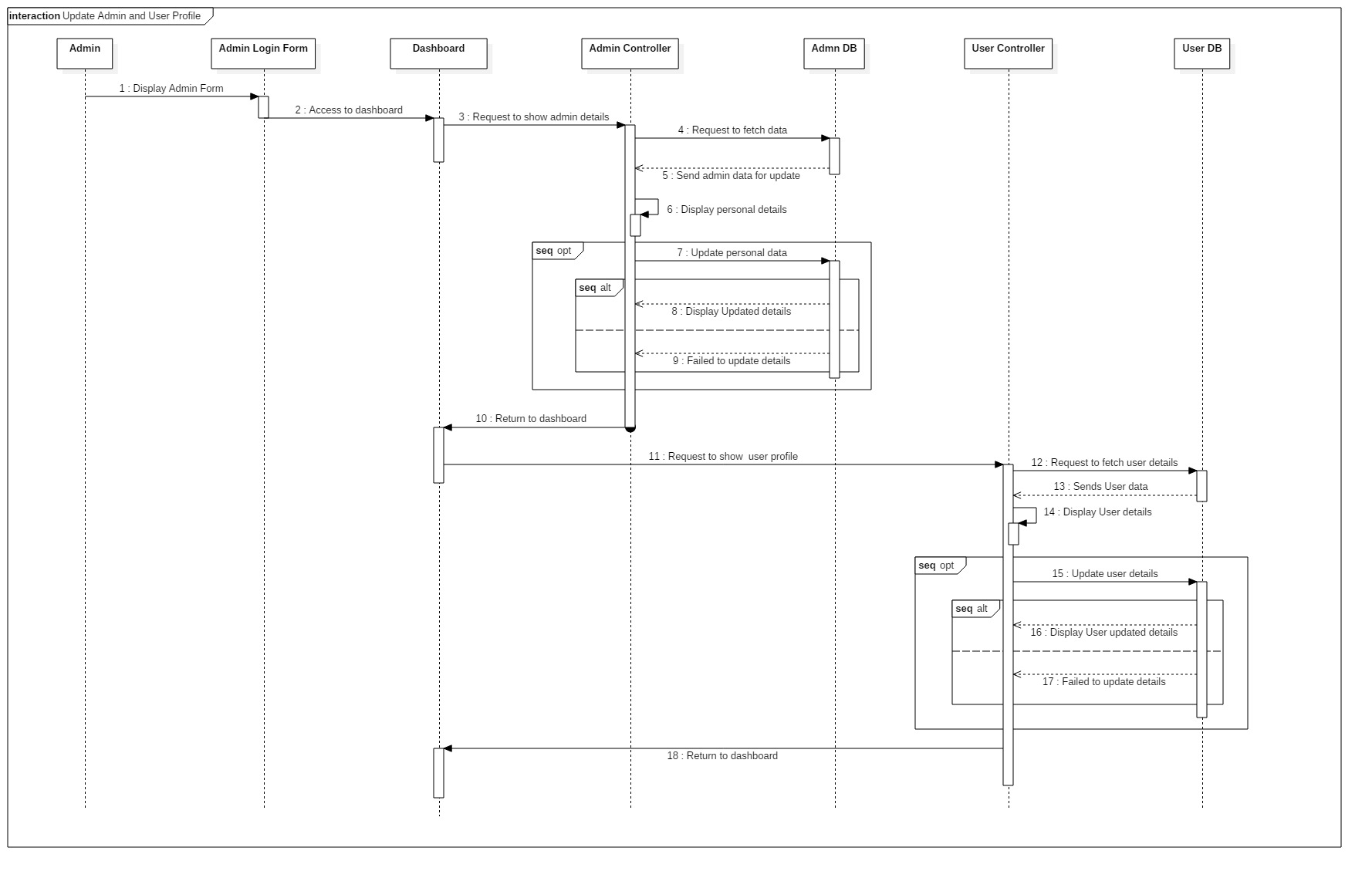
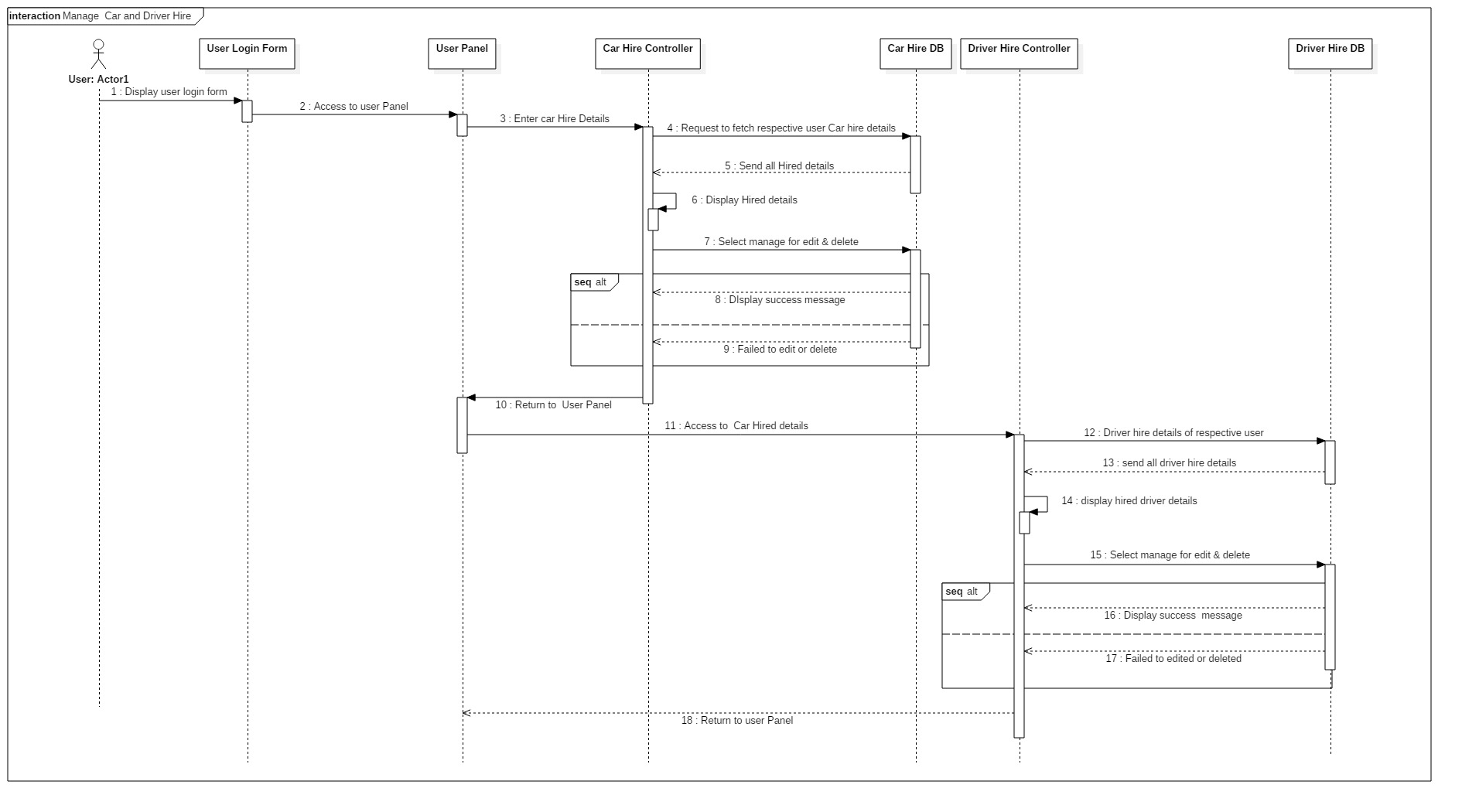
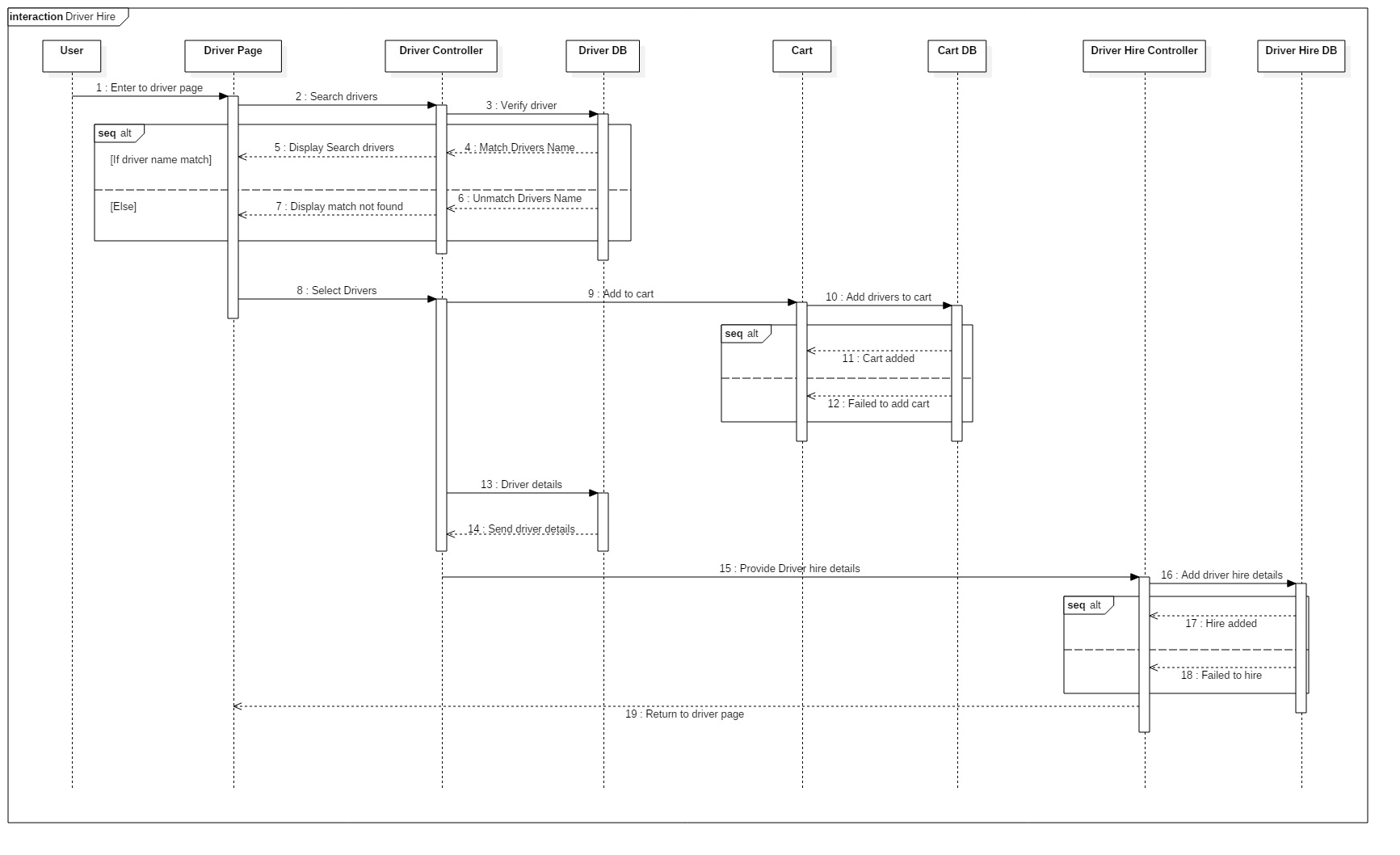
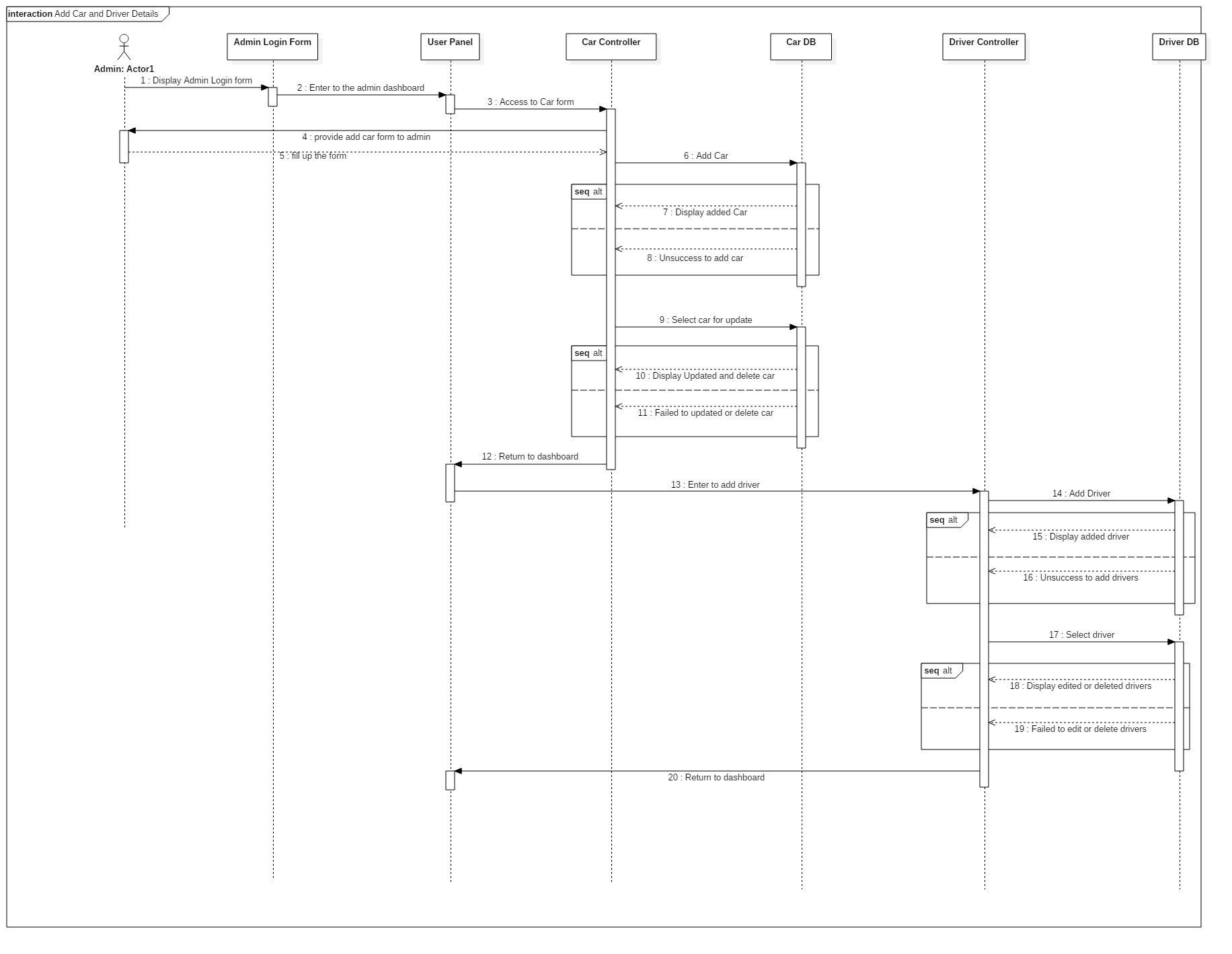
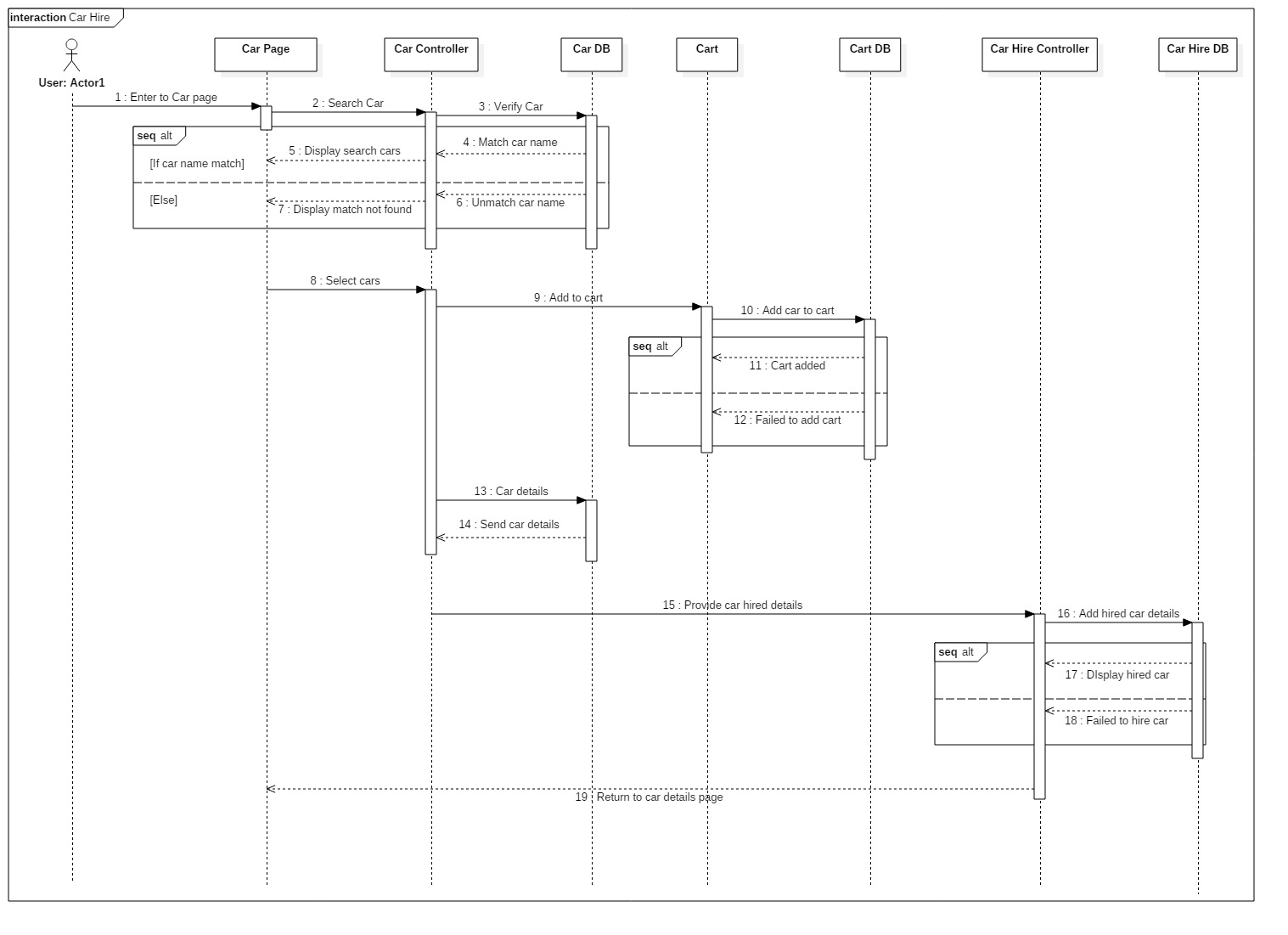
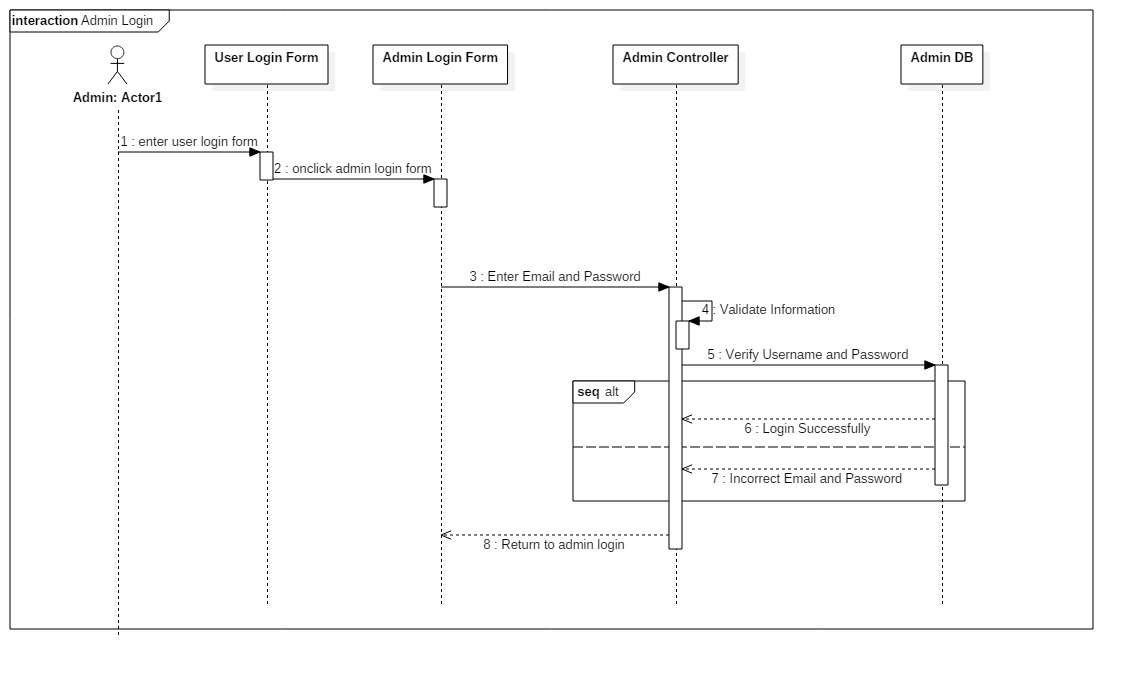


In the above diagram, there is shown the operation of the admin to the system. The black dot is an initial state which means starting point. After the admin login to the system he/she get access to the admin dashboard. After getting the access to the system admin has a permission to manage driver, car, and user and also he/she can able to change the respective password. At the manage driver, Admin can add, update and delete the driver which is stored or to be store in the database. At the manage car, also admin can add, update and delete the car details. And at the manage user section, admin can only update or delete the registered user details according to the user request. The last operation admin can process is update self-password.



### 3.2.2 – Sequence Diagram

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No** | **Notation Used** | **Notation Name** | **Description** |
| **1.** |  | Object | Helps to indicate the object. |
| **2.** | Actor | Actor | Helps to indicate the actor like user, admin, etc. |
| **3.** |  | Lifeline | Helps to show the lifeline of the object. |
| **4.** |  | Activation | Helps to represent the period during which an element is performing an operation. |
| **5.** | Call Message | Message Arrow | Helps to define a particular communication between lifelines of an interaction. |
| **6.** | Return Message | Reply Message | Helps to define a particular communication between lifelines of an interactions. |



## 3.3 – Database Modeling

### 3.3.1 – Data Dictionary

Data dictionary is a set of files that includes a database’s metadata. It contains record about other objects in the database. It is used to confirming data requirements and for database developers to create and maintain a database system. The data type used in data dictionary are integer, date, varchar and time. I have separated the primary key and foreign key of the entities and also the maximum length. The data dictionary I have created for my project are below:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Car Table** | | | | |
| **Column Name** | **Data Type** | **Primary Key** | **Foreign Key** | **Nullable** |
| car\_id | integer(10) | Yes | No | No |
| model | varchar(100) | No | No | Yes |
| brand | varchar(100) | No | No | Yes |
| type | varchar(100) | No | No | Yes |
| No\_of\_doors | varchar(100) | No | No | Yes |
| No\_of\_seats | varchar(100) | No | No | Yes |
| luggage | varchar(100) | No | No | Yes |
| transmission | varchar(100) | No | No | Yes |
| mileage | varchar(100) | No | No | Yes |
| Air\_conditioner | varchar(100) | No | No | Yes |
| Engine | varchar(100) | No | No | Yes |
| Fuel\_type | varchar(100) | No | No | Yes |
| image | varchar(100) | No | No | Yes |
| Price | varchar(100) | No | No | Yes |
| date | date | No | No | Yes |
| Admin\_id | Integer(10) | No | Yes | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Driver Table** | | | | |
| **Column Name** | **Data Type** | **Primary Key** | **Foreign Key** | **Nullable** |
| Driver\_id | integer(10) | Yes | No | No |
| First\_name | varchar(100) | No | No | Yes |
| Last\_name | varchar(100) | No | No | Yes |
| Date\_of\_birth | Date | No | No | Yes |
| Age | integer(10) | No | No | Yes |
| Phone\_no | varchar(100) | No | No | Yes |
| Liscence\_no | varchar(100) | No | No | Yes |
| address | varchar(100) | No | No | Yes |
| Image | varchar(100) | No | No | Yes |
| price | varchar(100) | No | No | Yes |
| date | date | No | No | Yes |
| Admin\_id | Integer(10) | No | Yes | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Car Hire Table** | | | | |
| **Column Name** | **Data Type** | **Primary Key** | **Foreign Key** | **Nullable** |
| Car\_hire\_id | integer(10) | Yes | No | No |
| From\_location | Varchar(100) | No | No | Yes |
| Pick\_up | date | No | No | Yes |
| Pick\_up\_time | time | No | No | Yes |
| To\_location | varchar(100) | No | No | Yes |
| Drop\_off | Date | No | No | Yes |
| Price | varchar(100) | No | No | Yes |
| Date | date | No | No | Yes |
| User\_id | integer(10) | No | Yes | No |
| Car\_id | integer(10) | No | Yes | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **User Table** | | | | |
| **Column Name** | **Data Type** | **Primary Key** | **Foreign Key** | **Nullable** |
| user\_id | integer(10) | Yes | No | No |
| First\_name | varchar(100) | No | No | Yes |
| Last\_name | varchar(100) | No | No | Yes |
| Address | varchar(100) | No | No | Yes |
| Email | varchar(100) | No | No | Yes |
| Password | varchar(100) | No | No | Yes |
| gender | varchar(100) | No | No | Yes |
| Phone\_number | varchar(100) | No | No | Yes |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Driver Hire Table** | | | | |
| **Column Name** | **Data Type** | **Primary Key** | **Foreign Key** | **Nullable** |
| Driver\_hire\_id | integer(10) | Yes | No | No |
| Car\_model | Varchar(100) | No | No | Yes |
| From\_location | Varchar(100) | No | No | Yes |
| Pick\_up | date | No | No | Yes |
| Pick\_up\_time | time | No | No | Yes |
| To\_location | varchar(100) | No | No | Yes |
| Drop\_off | date | No | No | Yes |
| price | Varchar(100) | No | No | Yes |
| date | date | No | No | Yes |
| User\_id | integer(10) | No | Yes | No |
| Driver\_id | integer(10) | No | Yes | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Admin Table** | | | | |
| **Column Name** | **Data Type** | **Primary Key** | **Foreign Key** | **Nullable** |
| admin\_id | integer(10) | Yes | No | No |
| username | varchar(100) | No | No | Yes |
| password | varchar(100) | No | No | Yes |
| email | Varchar(100) | No | No | Yes |
| Phone\_number | Varchar(100) | No | No | No |
| gender | Varchar(100) | No | No | Yes |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Question Table** | | | | |
| **Column Name** | **Data Type** | **Primary Key** | **Foreign Key** | **Nullable** |
| Question\_id | integer(10) | Yes | No | No |
| Question | varchar(100) | No | No | Yes |
| Date | date | No | No | Yes |
| User\_id | varchar(100) | No | Yes | No |

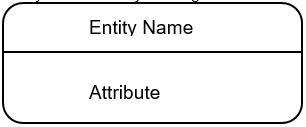
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Comment Table** | | | | |
| **Column Name** | **Data Type** | **Primary Key** | **Foreign Key** | **Nullable** |
| Comment\_id | integer(10) | PK | No | No |
| Comment | varchar(100) | No | No | Yes |
| Date | Date | No | No | Yes |
| User\_id | integer(10) | No | Yes | No |
| question\_id | integer(10) | No | Yes | No |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Cart Table** | | | | |
| **Column Name** | **Data Type** | **Primary Key** | **Foreign Key** | **Nullable** |
| Cart\_id | integer(10) | Yes | No | No |
| driver\_id | Integer(10) | No | Yes | No |
| car\_id | integer(10) | No | Yes | No |

### 3.3.2 – ER-Diagram

An Entity Relationship diagram manifest the relationship of entity sets stored in a database. It is a representation of data within a domain. An entity is identifying by extracting objects that are relevant and meaningful to the problem domain and the system to be develop. In entity relation modelling, the term entity has synonyms “table”, “database table” and “entity-type”.

The notations used in ER diagram are:



The relations used in ER diagram are:

* One to one 
* One to many 
* Many to many 

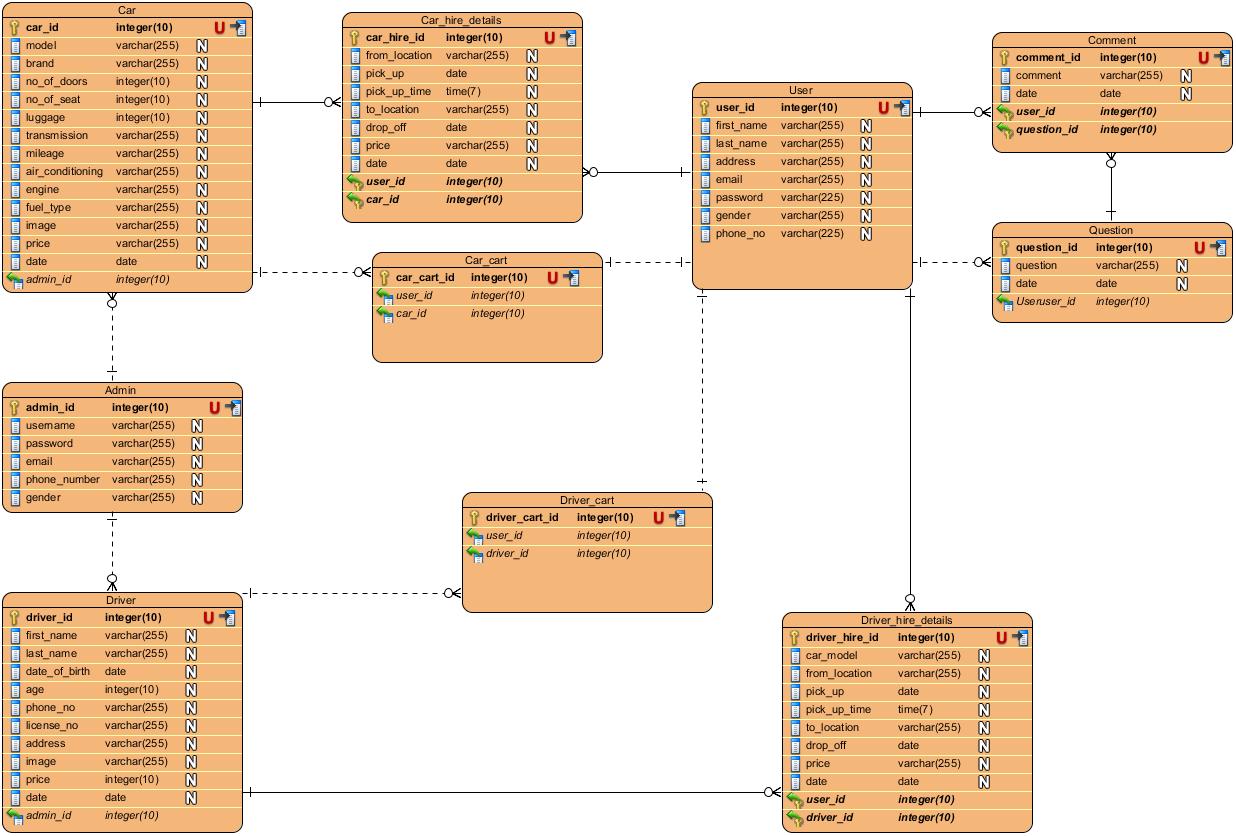


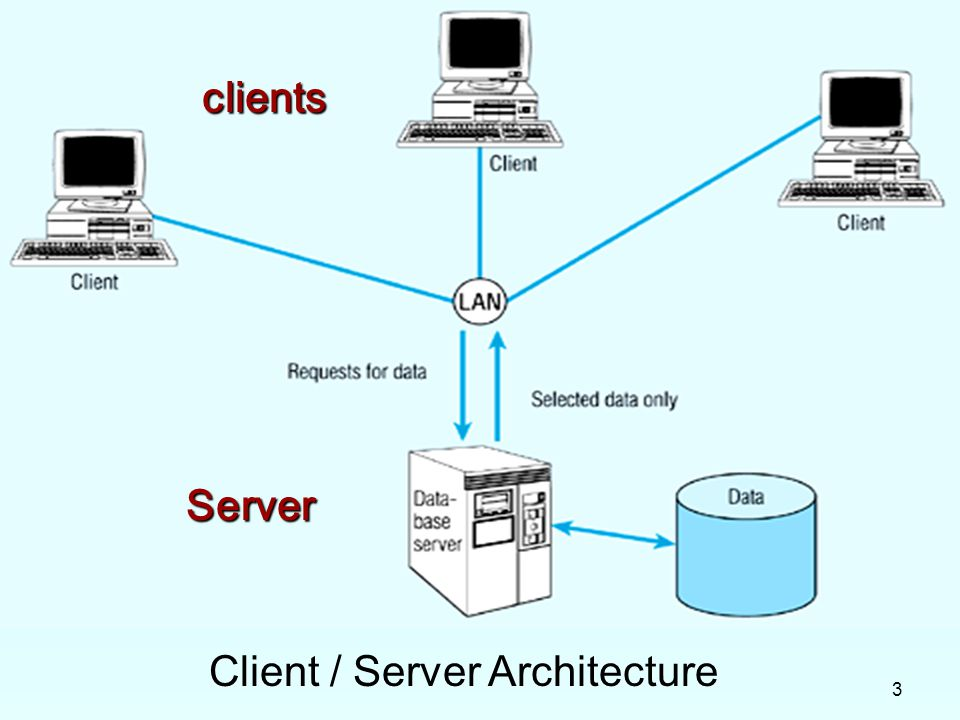
Fig: Entity relation diagram of online car rental management system.

Justification of creating ER diagram in my project are:

1. ER helps to highlights the entities of my system and the relationship between those entities.
2. Helps to reveals if any errors occurs.
3. Helps to visualize how data connected in general ways.
4. The model can be reverse enginerred to create database.
5. It amplify the logical structure of database.

## 3.4 – Architecture Diagram

Architecture is a coherent set of concepts for a structure. An architecture diagram is a graphical representation of a set of concepts, that are the part of an architecture, including their principals, elements and components. These concepts are often visualized at four levels of abstraction. These are conceptual, logical, physical and implementation level. I have chosen client server architecture in my project because it is a computing model in which the server hosts, delivers and manages most of the resources and services to be consumed by the client.



I have used client server architecture because this project is based on the online transactions between the customer and the system.

Justification of using Client-Server Architecture in my project are listed below:

* This architecture is easy accessible by the client from any places.
* It helps to secure the data of the clients.
* It enforces the security policies on the data.

## 3.5 – Prototype Design

Prototype design is the graphical user interface design of the entire system. I have used ***Balsamiq Mockups 3*** software to make the below prototyping design. It shows the interface of a new system.

