# **Chapter 3: Design**

# **3.1 Introduction**

Design is the one of the most crucial phase in SDLC. Based on the user requirements and the analysis of the system, the system must be designed. The logical design made during the analysis phase starts to taking shape into physical design during this phase. Problems are identified during the previous phase and a detailed description of what is needed to solve that problems are produced this phase. This is the phase where we take decision about how the system will meet the requirements identified during analysis phase.

# **3.2 Structural Modelling**

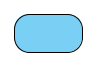
## **3.1.1 Final class diagram**

Class diagram is UML structural diagram that shows the relationship between class(entities) of the system. It is static diagram that represents the static view of the system. Class diagram describes the class’s attributes, operations, and the system’s constraints.

# **3.3 Behavioral Modelling**

## **3.2.1 Activity Diagram**

Activity diagram is flowchart to represent the flow from one activity to another activity. The activity can be represented as the operation of the system. It shows the workflow in the system. It shows the dynamic behavior of the system and also describe the sequence from one activity to the another.

**Notations:**

**Activity symbol**



**Start symbol**

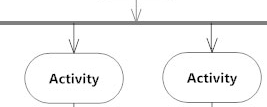


**Flow final symbol**



**End Symbol**

**Interruption symbol**



**Fork symbol**



**Decision symbol**

**Fig:**

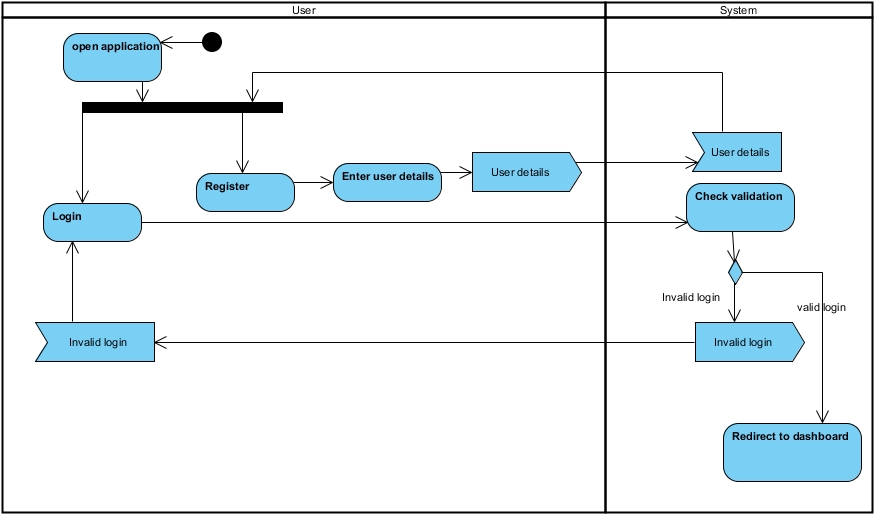


Fig: Login and register

This figure shows the activity diagram of login and register. User must register and then signed in to use the system. For registration, user send their details to the system and after this, user can login. While logging in, system validates the login details and redirect to dashboard if login is valid.

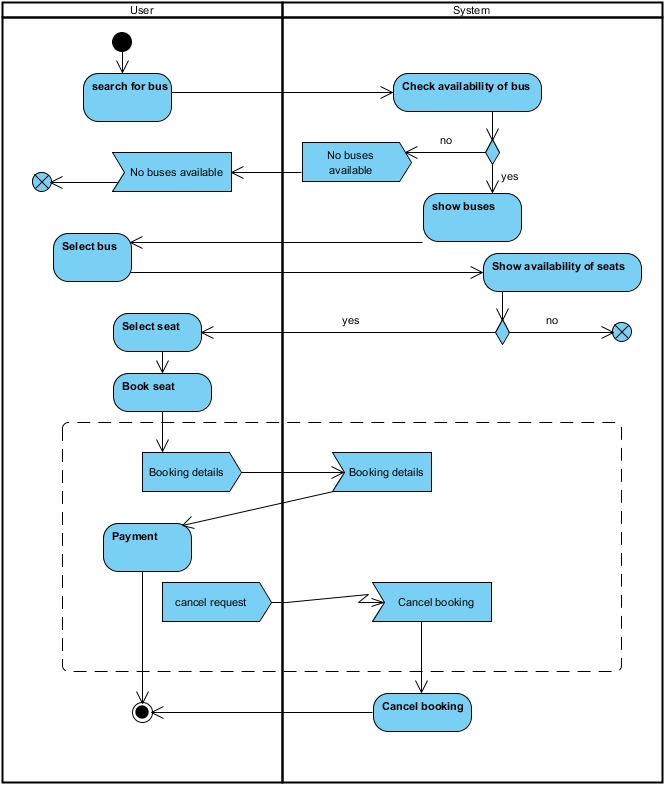


Fig: Main activity

This diagram shows the main activity of the system after login. User search for the bus and system check it. If it is available, system show it to the user. Otherwise, no buses are available message is sent. Then, user select the bus and select the available seats. After this, user book the seat and payment is done.

## **3.2.2 Sequence Diagram**

Sequence diagram is UML behavioral diagram that describe the interactions among class and object in term of an exchange of messages over time. It is also known as “Event diagram”. It shows how the system will behave in different scenarios and helps to validate the logic of complex operations and functions.

**Notations:**

**Lifeline**

 **Send message**

 **Return message**

 **User**

**Fig:**

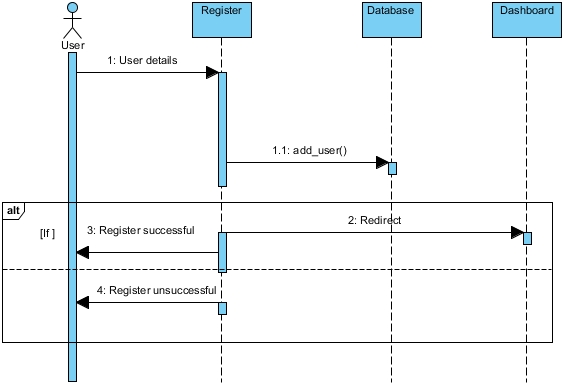


Fig: Registration

This figure shows the sequence diagram of registration. User sends user details to the system and the system sends it to the database. If registration is successful, successful message is displayed and redirected to the dashboard. Otherwise, unsuccessful message is sent.

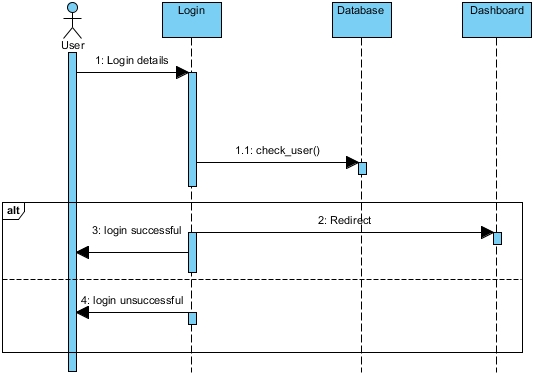


Fig: Login

This figure shows the sequence diagram of login. User sends login details to the system and the system validate it with the database. If login is successful, successful message is displayed and redirected to the dashboard. Otherwise, unsuccessful message is sent.

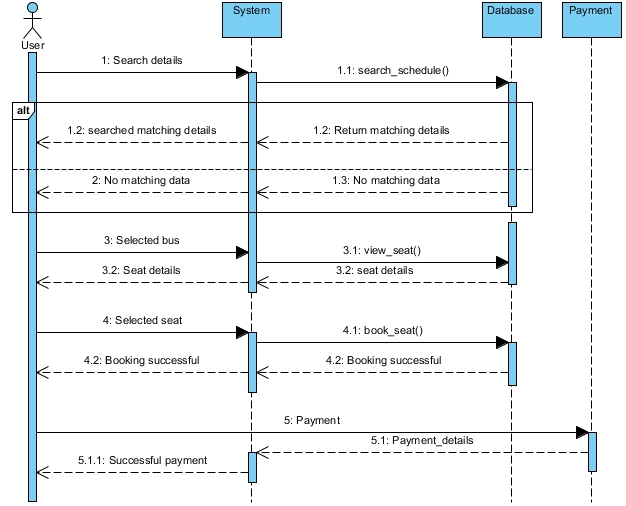


Fig: System Flow Sequence diagram

This figure shows the sequence after user logged in. User send search details to the system and system search it in database. If it is matched, then matching details are sent to the user. Otherwise, no matching data message is sent. User select the bus and system view the seat of the selected bus and show it to the user. Then, user selects the available seats and send it to the system for booking. After booking, system sent the successful message. Payment is done and payment details are sent to the user.

# **3.4 Database Modelling**

## **3.3.1 Data Dictionary**

Data dictionary is file that contains the database’s metadata. It is also known as “Data definition matrix” or “Metadata repository”. It provides the names, definitions and attributes about the data elements.

**User:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Datatype** | **Length** | **Constraint** | **Nullable** | **Key** |
| user\_id | Integer | 10 | pk\_user\_id | No | Primary |
| first\_name | varchar | 20 | **-** | No | - |
| last\_name | varchar | 20 | - | No | - |
| email | varchar | 20 | - | No | Unique |
| phone\_num | varchar | 20 | - | No | Unique |
| address | varchar | 20 | - | No | - |
| password | varchar | 20 | - | No | - |
| Type | varchar | 10 | - | No | - |

**Bus:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Datatype** | **Length** | **Constraint** | **Nullable** | **Key** |
| bus\_id | Integer | 10 | pk\_bus\_id | No | Primary |
| name | Varchar | 20 | - | No | - |
| owner | varchar | 20 | - | Yes | - |

**Route:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Datatype** | **Length** | **Constraint** | **Nullable** | **Key** |
| route\_id | Integer | 10 | pk\_route\_id | No | Primary |
| starting | Varchar | 20 | - | No | - |
| destination | Varchar | 20 | - | No | - |
| cost | Int | 10 | - | No | - |

**Seat:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Datatype** | **Length** | **Constraint** | **Nullable** | **Key** |
| seat\_id | Integer | 10 | pk\_seat\_id | No | Primary |
| seat\_num | Integer | 10 | - | No | - |
| status | Varchar | 10 | - | No | - |
| bus\_id | Integer | 10 | fk\_bus\_id | No | Foreign |

**Schedule:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Datatype** | **Length** | **Constraint** | **Nullable** | **Key** |
| schedule\_id | Integer | 10 | pk\_schedule\_id | No | Primary |
| dept\_date | Date | - | - | No | - |
| dept\_time | Time | - | - | No | - |
| arrival\_date | Date | - | - | No | - |
| arrival\_time | Time | - | - | No | - |
| bus\_id | Int | 10 | fk\_bus\_id1 | No | Foreign |
| route\_id | Int | 10 | fk\_route\_id | No | Foreign |

**Booking:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Datatype** | **Length** | **Constraint** | **Nullable** | **Key** |
| booking\_id | Integer | 10 | pk\_booking\_id | No | Primary |
| date | Date | - | - | No | - |
| user\_id | Integer | 10 | fk\_user\_id | No | Foreign |
| schedule\_id | Integer | 10 | fk\_schedule\_id | No | Foreign |
| seat\_id | Integer | 10 | fk\_seat\_id | No | Foreign |

**Payment:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Column** | **Datatype** | **Length** | **Constraint** | **Nullable** | **Key** |
| payment\_id | Integer | 10 | pk\_payemnt\_id | No | Primary |
| booking\_id | Integer | 10 | fk\_booking\_id | No | Foreign |
| amount | Integer | 10 | - | No | - |
| card\_num | integer | 16 | - | No | - |
| card\_name | Varchar | 25 | - | No | - |

## **3.3.2 ER diagram**

ER diagram is a type of structural diagram in which the relationships of entities stored in the database are shown. It can be used as the foundation for relational database. It illustrates the logical structure of the database.

**Notations:**



**Entity**

 **One to many relationship**

 **One to one relationship**

**Fig:**

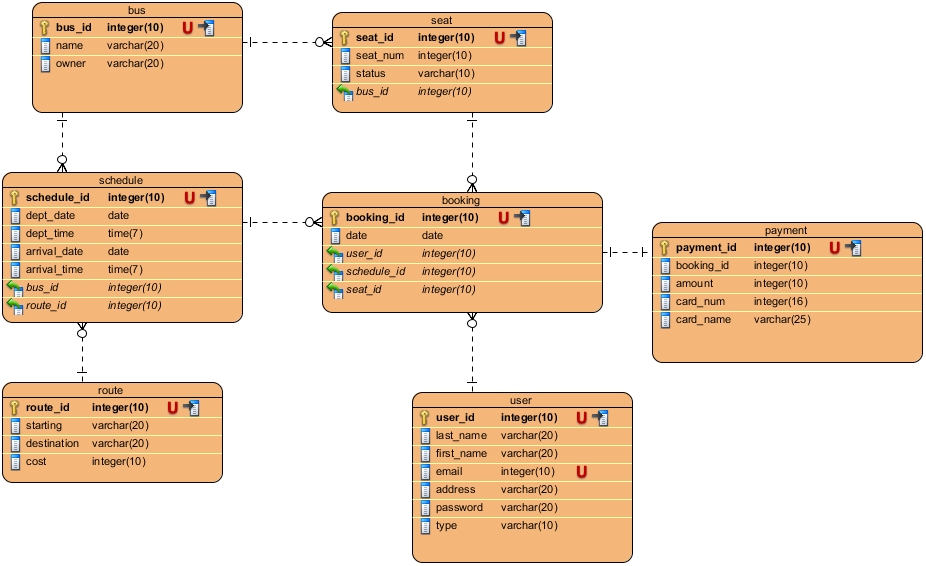


Fig: ER diagram

Above ER diagram is evolved from the data dictionary created above with the relationships that the entities have with each other.

# **3.5 Architectural Model**

# **3.6 Prototyping**

Prototyping refers to the creating the prototypes (model)of the system. It shows how the system will look like and how it will work. It facilitates in system implementation since feedback can be received from the users.

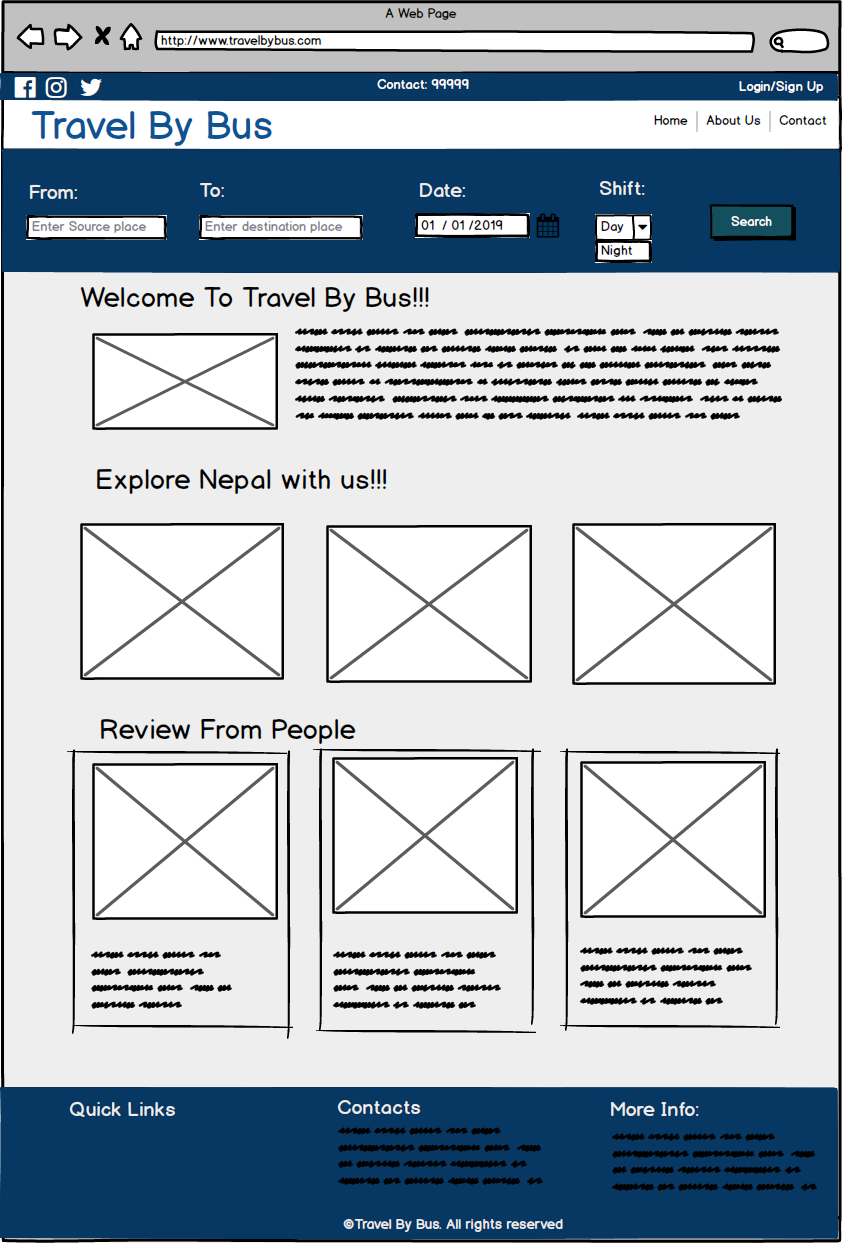


Fig: Homepage

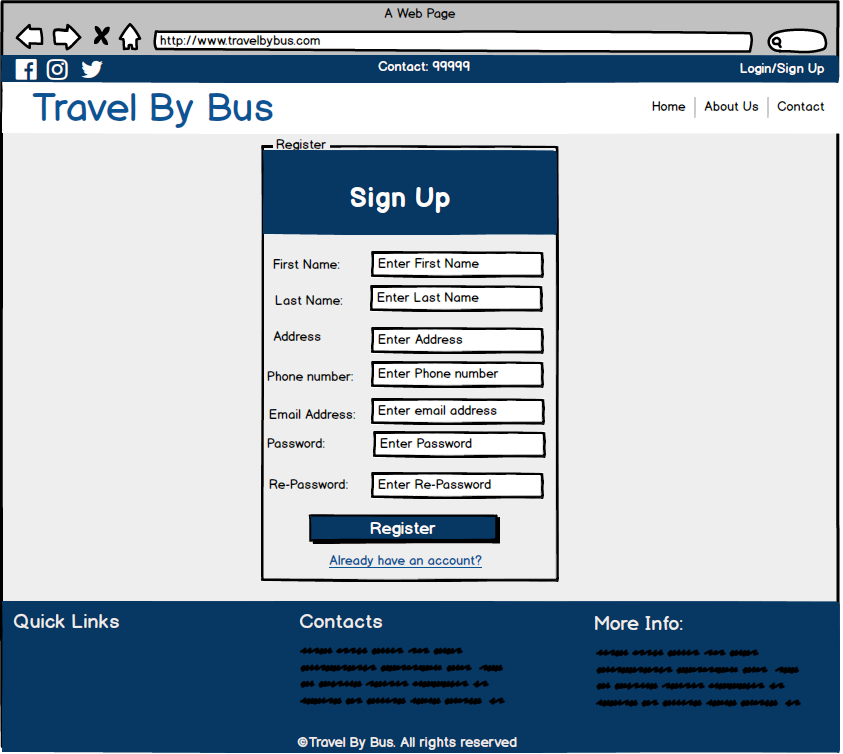


Fig: Register

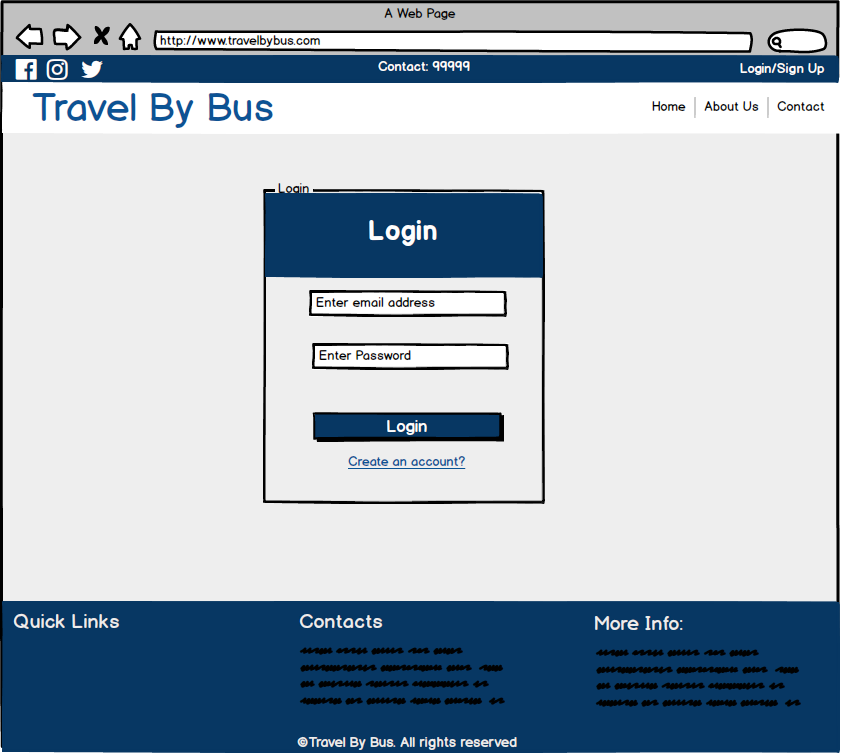


Fig: Login

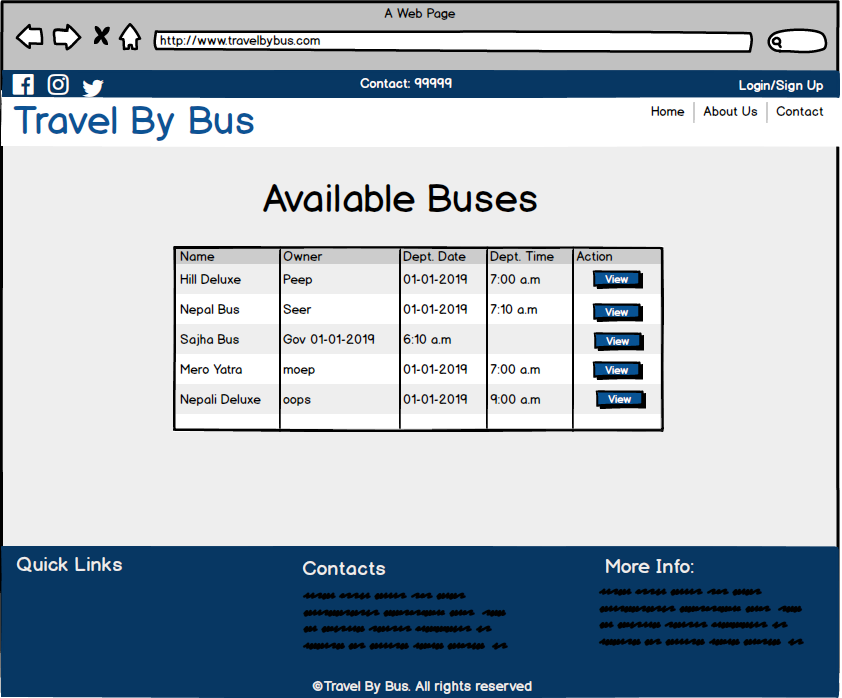


Fig: View Bus

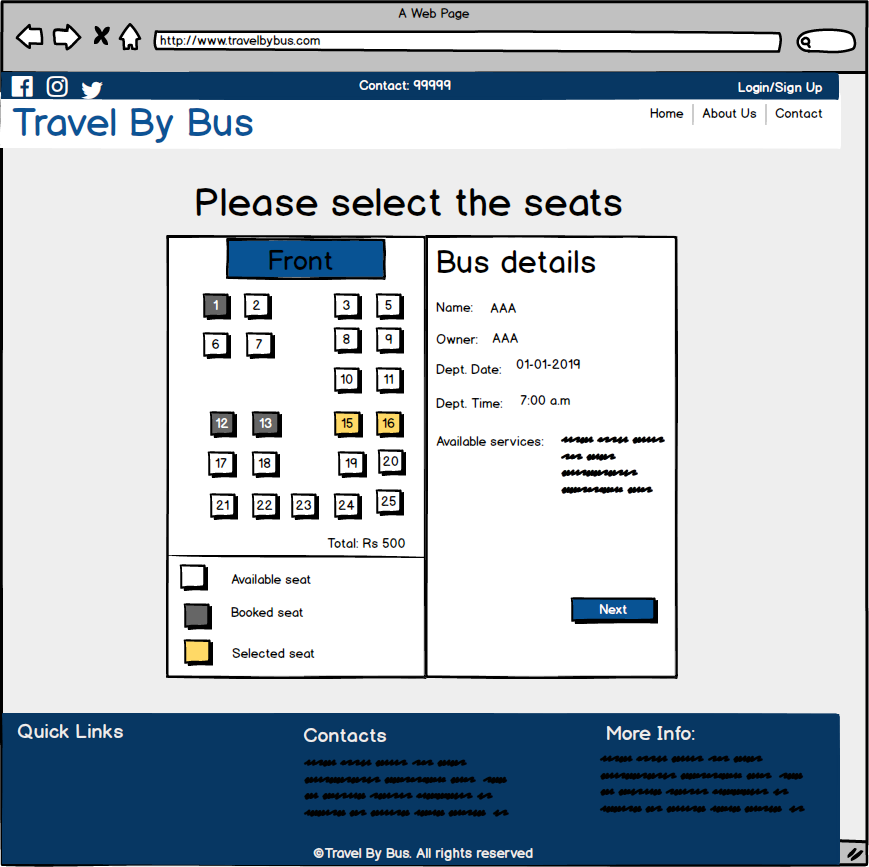


Fig: Seat Booking

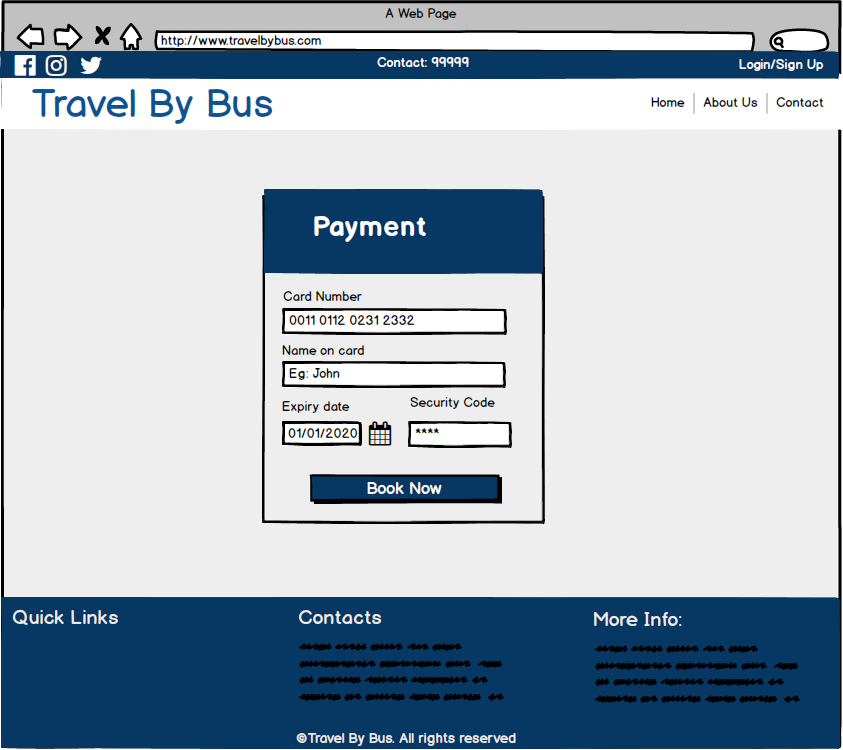


Fig: Payment

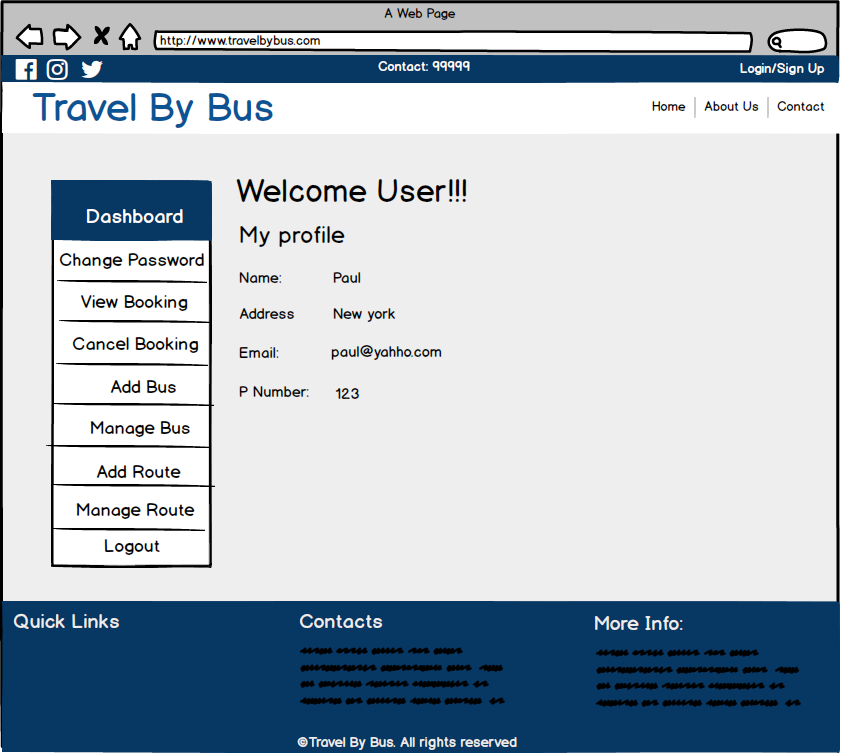


Fig: Admin Panel

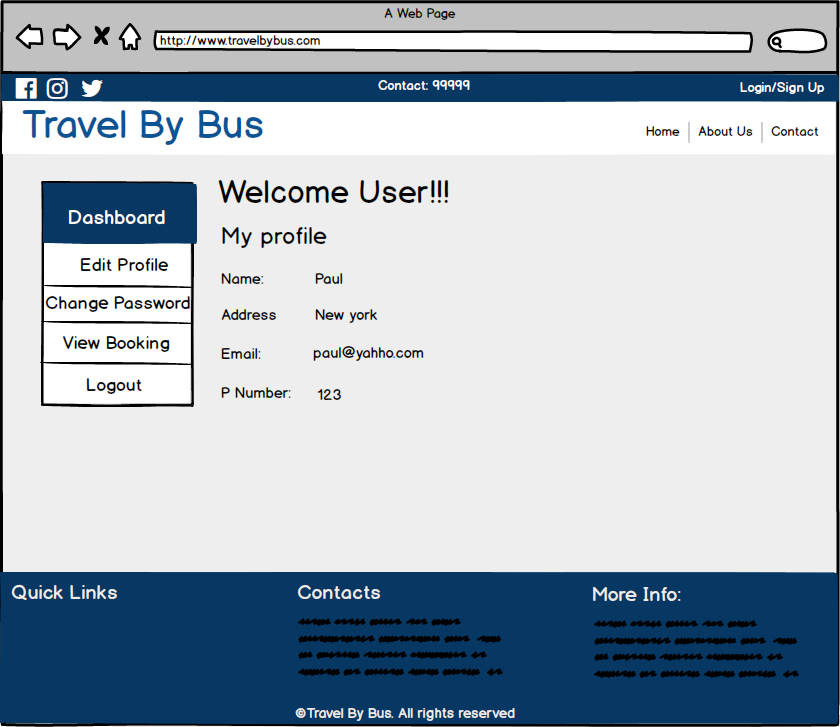


Fig: General user panel