# **Chapter-3:- Design**

Design is the process of planning and identification for the development of a system or software. In this we implement an activity and process or result of that plan in the form of prototype.

## **3.1:- Structural Modelling.**

Structural Modelling of the software display the organization of system in term of components that make up the system and their relationships. In this part we are going to focus on class diagram for modelling of the structure of classes in our software system. Structural Modelling is grouped into different categories i.e. class diagram, flow chart etc.

### **3.1.1:- Final Class Diagram.**

**Definition:-**

Class Diagram is the type of the static modelling that shows the static view of an application. Class Diagram shows the operations and attributes of the class. They are mostly used in the modelling for the object oriented system.

**Justification for use of class diagram:-**

* It illustrates the shape of a system by illustrating the system class, their attributes, operation and relationships between objects.
* It reduce the maintenance time by providing blueprint for maintenance programmers to get an idea that how the application is structured before examining the actual code.

**Notation Used:-**

The commonly used notation for class diagram is shown below:-

|  |
| --- |
| Class Name |
| Attributes |
| Operation |

* Association
* Inheritance

**Diagram:-**

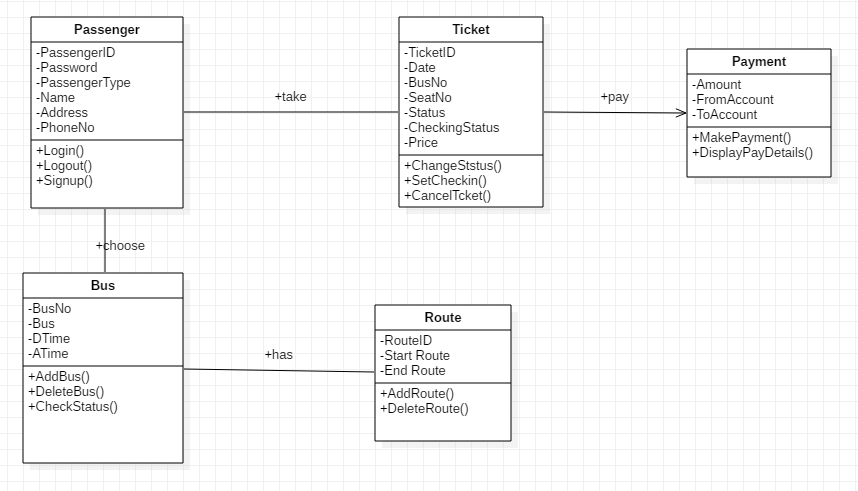


Figure : Final Class Diagram for Bus Ticket Management System.

**Diagram Explanation:-**

For Our Project Bus Ticket Management System, we have five classes and namely Passenger, Ticket, Payment, Bus, and Route. Passenger can choose Bus, Route and take Ticket and had to do payment for that. Passenger can Login and Logout from the system and their destination route bus etc.

### **3.1.2:- Flowchart Diagram.**

**Definition:-**

A flowchart is a formalized graphic representation of the logical sequence, work or manufacturing process, organization chart or similar formalized structure. The purpose of a flow chart is to provide people with common language point when dealing with a project or process.

**Justification for the use of the Flowchart Diagram:-**

* It is an excellent way of documenting each stage of process.
* It acts as a blue print guiding to the programmer through the development.

**Notation Used:-**

* Process
* Start/End
* Comment/Note
* Decision Symbol
* Connector
* Off-Page Connector

**Diagram:-**

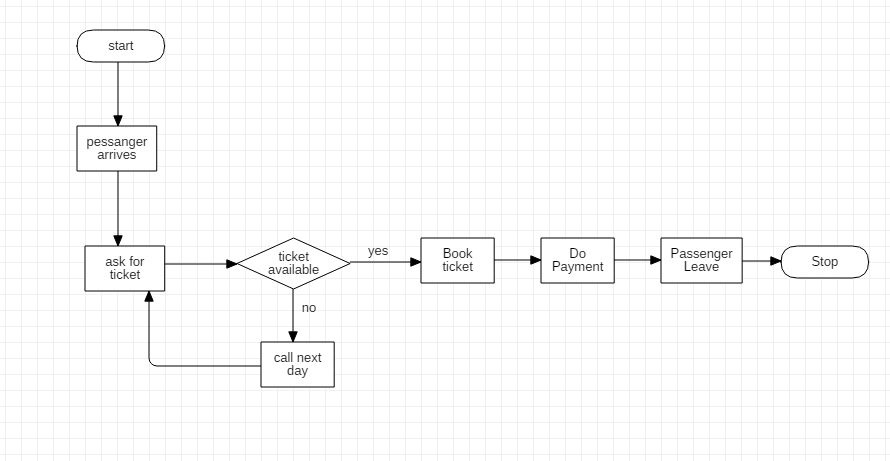
****

Figure : Flowchart Diagram for Bus Ticket Management System.

**Diagram Explanation:-**

In this diagram, passenger goes to the bus station and ask for ticket, if ticket is available then passenger will book ticket and do payment and passenger took ticket and leave the station otherwise if ticket is not available then passenger will be called next day.

## **3.2:- Behavioral Modelling.**

Behavioral modelling is used to visualize, specify, construct and document the dynamic aspects of a system. It deals with runtime behavior of the system. Behavioral modelling are grouped in different categories i.e. use case, activity diagram, sequence diagram etc.

### **3.2.1:- Activity Diagram.**

**Definition:-**

Activity Diagram illustrates the flow of the activities that are ongoing non-atomic operations in a state machine. Activities leads to action that are atomic operation. Activity diagrams composed of activity states and action states, transitions and Objects.

**Justification for the use of Activity Diagram:-**

* It is used for modelling of workflows as viewed by actors, interacting with the system.
* It is used for modelling of details of operations or computations using flowcharts.

**Notation Used:-**

* Process
* External Entities
* Data Store
* Data Flow

**Diagram:-**

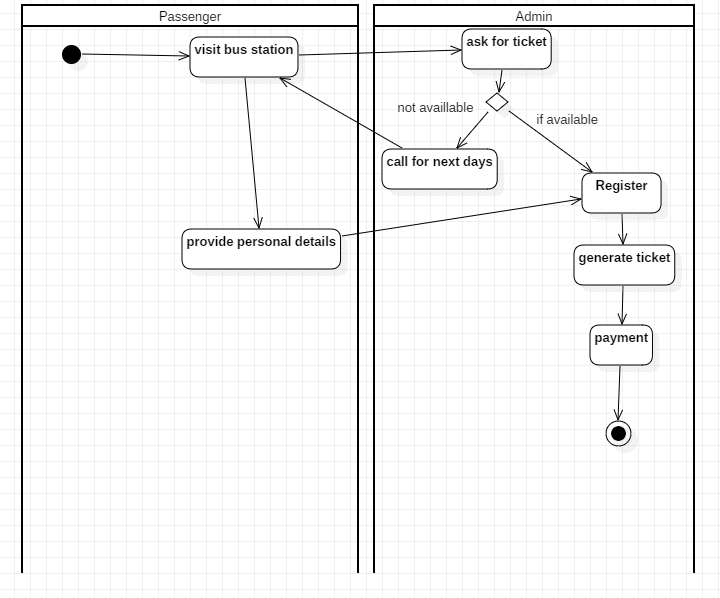


Figure : Activity Diagram for Bus Ticket Management System

### **3.2.2:- Sequence Diagram.**

**Definition:-**

Sequence diagrams are the interaction diagrams that illustrates the ordering of the messages according to time. They illustrate the interaction between objects in context of a collaboration. They detail that how the operation are carried out.

**Justification for the use of Sequence Diagram:-**

* They model the high-level interaction between active objects in the system.
* They model the interaction between object instances with the collaboration that realize a use case.
* They model the interaction between objects with collaboration that realize an operation.

**Notation Used:-**

* **Start**
* **Activity**
* **Decisions**
* **Terminate**
* **Basic flow**

**Diagram:-**

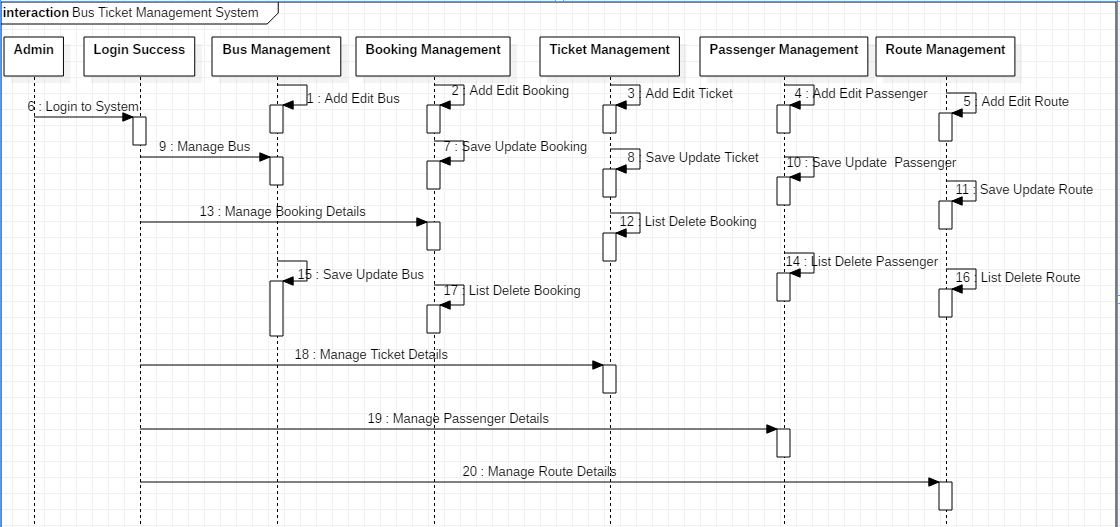


Figure : Sequence Diagram for Bus Ticket Management System

## **3.3:- Database Modelling.**

Database Modelling is the type of data model which states the logical structure of a database and fundamentally determines in which format data can be stored, organized and manipulated.

### **3.3.1:- Data Dictionary.**

Data Dictionary contains the metadata i.e. data about database. It is very important because it contains the information such as what is in database, who is allowed to access and where is database physically stored etc.

The data dictionary contains the following things:-

* It contain information about the names of all the database tables and their schemas.
* It contain detail information about all tables in the database i.e. owners, security constraints, date of creation of database.
* It contain the physical information about tables i.e. where they are stored and how.
* It contain information about table constraints such as key attributes, foreign key information etc.
* It contain the information about the database views that are visible.

The following are the data dictionary of the table I have created in data base for the project Bus Ticket Management System.

1. Table: Bus.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Field Name | Data Type | Length | Constraints |
| 1 | BusNo | Varchar | 25 | Primary Key |
| 2 | Bus | Varchar | 50 | - |
| 3 | DTime | DateTime | - | - |
| 4 | ATime | DateTime | - | - |
| 5 | RouteID | Varchar | 50 | Foreign Key |

b. Table: Passenger

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Field Name | Data Type | Length | Constraints |
| 1 | PassengerID | Varchar | 100 | Primary Key |
| 2 | Password | Varchar | 100 | - |
| 3 | Name | Varchar | 255 | - |
| 4 | Address | Varchar | 255 | - |
| 5 | PhoneNo | Varchar | 50 | - |
| 6 | PassengerType | Varchar | 100 | - |

1. Table: Ticket.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Field Name | Data Type | Length | Constraints |
| 1 | TicketID | Integer | 10 | Primary Key |
| 2 | Date | Date | - | - |
| 3 | BusNo | Varchar | 25 | Foreign Key |
| 4 | SeatNo | Varchar | 50 | - |
| 5 | Price | Varchar | 30 | - |
| 6 | Status | Enum | - | -- |
| 7 | PassengerID | Varchar | 100 | Foreign Key |

1. Table: Route.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.N | Field Name | Data Type | Length | Constraints |
| 1 | RouteID | Varchar | 50 | Primary Key |
| 2 | Start Route | Varchar | 255 | - |
| 3 | End Route | Varchar | 255 | - |

### **3.3.2:- ER Diagram.**

**Definition:-**

ER diagram is known as Entity Relationship Diagram that graphical representation of an information system which illustrates the relationships among people, objects, places, concepts or events within the system. It help in defining business processes and can be used as foundation for relational database.

**Justification for the use of ER Diagram:-**

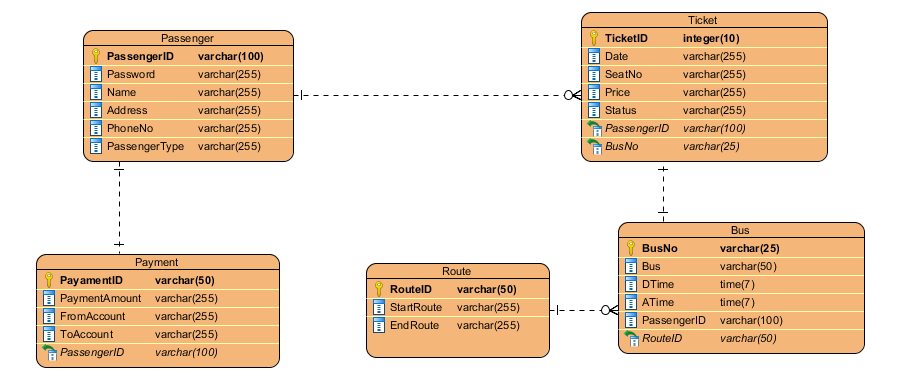
* ER Diagram illustrates logical structure of databases.
* ER Diagram are used to sketch out the design of database.

**Notation Used:-**

* Entity Type
* Weak Entity Type

* Relationship Type
* Identifying Relationship Type
* Attribute

**Diagram:-**



: ER Diagram for Bus Ticket Management System

## **3.4:- UI Modelling.**

User Interface Modelling is the method of designing interfaces in software or computerized devices with a focus on look or style. UI modelling refers to the graphical user interfaces but also includes voice controlled one.

### **3.4.1:- Prototyping.**

Prototyping is the draft representation that is built to test ideas for layout, behavior, and flow in a system. Prototype have an ability to determine an idea with clearly and higher level of engagement between developers and clients.

1. **Login Interface:-**

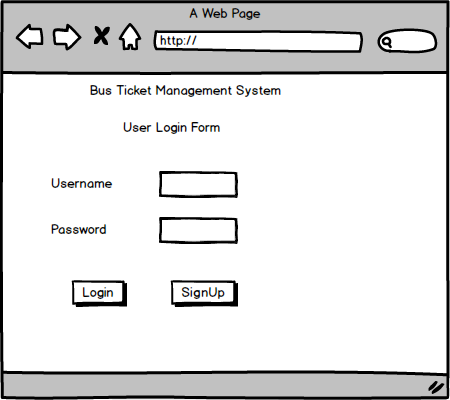


Figure : Login Interface Prototype

1. **Registration Interface:-**

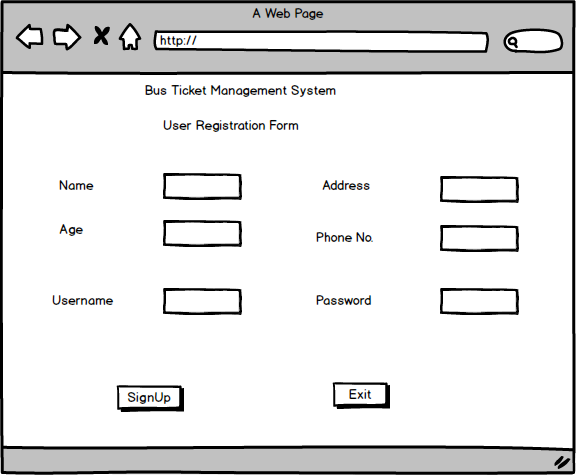


Figure : User Registration Prototype