TABLE OF CONTENTS

1.Introduction

Online Bus Ticket Booking System is a Web based application that works within a centralized network, should be used in a bus transportation system, a facility which is used to reserve seats, cancellation of reservation and different types of route enquiries used on securing quick reservations. OBTBS is built for managing and computerizing the traditional database, ticket booking and tracking bus and travel made. It maintains all customer details, bus details, reservation details. In order to achieve the design, In addition, HTML, Javascript , Css, language was used for the front-end of the software while the back end was designed using MySQL, PHP. The software achieved is capable of improving the customer hand and relationship management operations. It is recommended that despite the present functionality of the designed software, an additional functionality such as the use of E-mail to send tickets and notifications to the customer and an online payment using credit cards/debit cards should be implemented into the system.

2.Design considerations

This section describes many of the issues that are needed to be able to addressed or resolved before embarking on a complete design solution. This document is based on the version v1.0 as in

SRS document. There is a need for reference in case any part is not understood or felt incomplete

2.1 Assumptions

This OBTBS design makes several assumptions about the software and hardware requirements as is in the SRS. All the environmental operating requirements of both the user interface and the database can be found in the OBTBS requirements. Both the database and the user application make the following assumptions about the operating environment. The system can be described by the operating requirements associated with this document and in the SRS. The system application in execution will have the necessary resources availed as required. This entails sufficient memory and permanent storage space and the adequate CPU for the application. The application makes the following assumptions about its operating environment. The user machine will have MYSQL database components installed, as they are required for the system implementation. The machine will also have necessary database setup

2.2 Constraints

The OBTBS shall be a web based system. This system shall be developed using HTML, CSS, Javascript, PHP, Mysql database.

2.3 Design methodology

In designing the OBTBS , the following approach shall be used:

Water fall model will be used as the best language for this kind of system. This is because water fall model is suitable for visualizing, specifying, constructing and documenting the features of the system.

The design will take the following approach:

1.Designing the database

2.Creating relationships

3.Designing the user interfaces and the system processes.

2.4 System environment

System scalability and security are the requirements for the system architecture of the OBTBS. The system will accommodate scalability allowing flexibility within the system to expand, modify or downsize easily to meet the evolving business and technology change.

3. Architecture

3.1 System design

After the system has been implemented the mapping shall take place according to following:

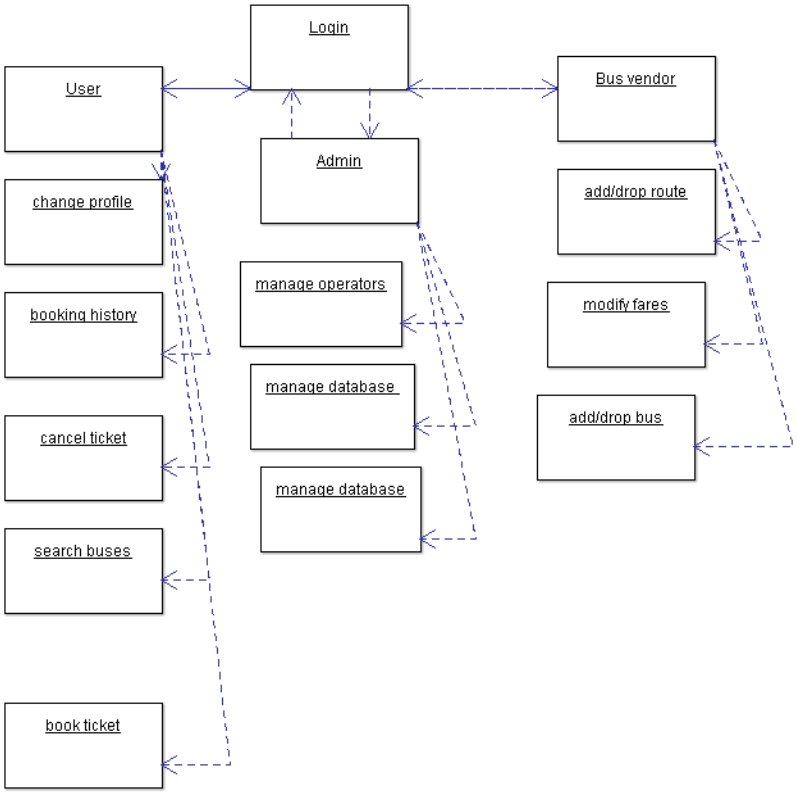

Figure 1 **System Design**

3.2 Functional decomposition tree

The main functions of the system is decomposed into smaller sub functions or sub-modules and further. The System shall take place following structure of organization after implementation. The decomposition is stable and functions should be made highly cohesive.

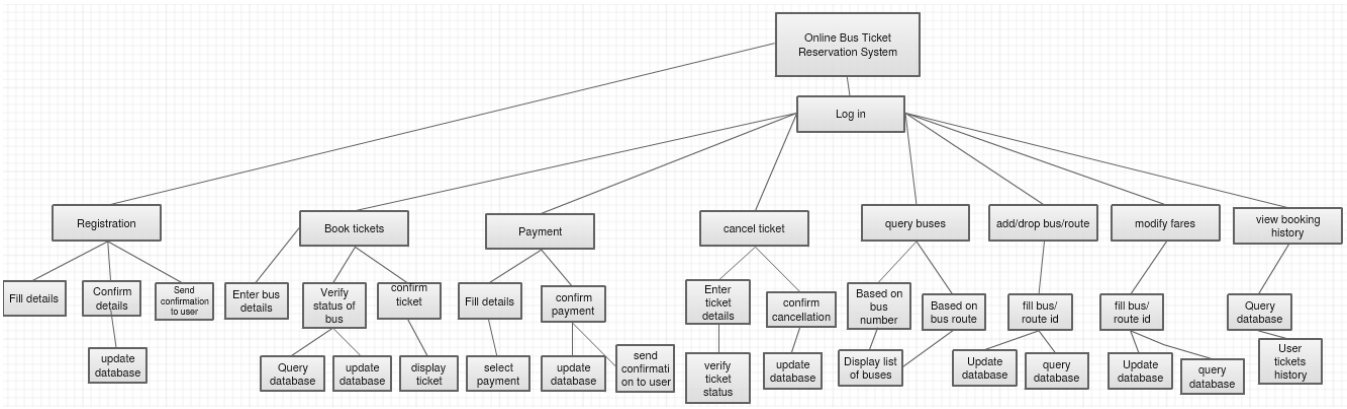

Figure 2 **Functional decomposition tree**

Modules involved in the system:

1. Registration: This module is required for the registration of new users or us vendors into accessing the system. This module has three sub-modules

1.1 Fill details: This sub-module is required to input the user entered details from the interface and store it in a data structure.

1.2 Confirm details: This sub-module is required to verify the details stored in previous data structure

1.2.1 Update database: Update new user into database

1.3 send confirmation: Intimate user about the registration credentials and confirmation via email

2. Book tickets

2.1 Enter bus details: Input user information of bus based on route/bus and store in data structure

2.2 Verify status of bus: check if bus entered is in service and has enough amount of seats

2.2.1 query database

2.2.2 Update database: update ticket in database if status is verified

3. Payment

3.1 Fill details: Get input from user regarding payment mode and credentials

3.2 Confirm payment: Verify payment and confirm the ticket and update database

4. Cancel ticket

4.1 Enter ticket details: Input ticket number and credentials

4.1.1 verify status of ticket: Verify if ticket is not expired and in cancellation period

4.2 confirm cancellation: Confirm cancellation after ticket is verified and update database

5.Query buses: Search the buses between selected route on particular date

5.1.1 Display list: Obtain list from database and display on interface

6.Add/drop route/bus: To add/drop a bus or a route of a vendor

6.1 Fill id: Input Bus/Route id from vendor to be added/dropped

6.1.1 Update database: If there is any change in bus route

6.1.2 Query database: If there is bus/route to be dropped

7.Modify Fares

7.1 Fill id: Input Bus id from vendor to be modified fare

7.1.1 Update database: Update database with new fare for the bus

7.1.2 Query database: Search for the bus fare to be modified

8.View booking history

8.1 Query database: for a particular user by query type user booking history

8.2 Display list: for the found list in database

3.3 Context diagram

Context diagram describes the main actors interacting with the system.

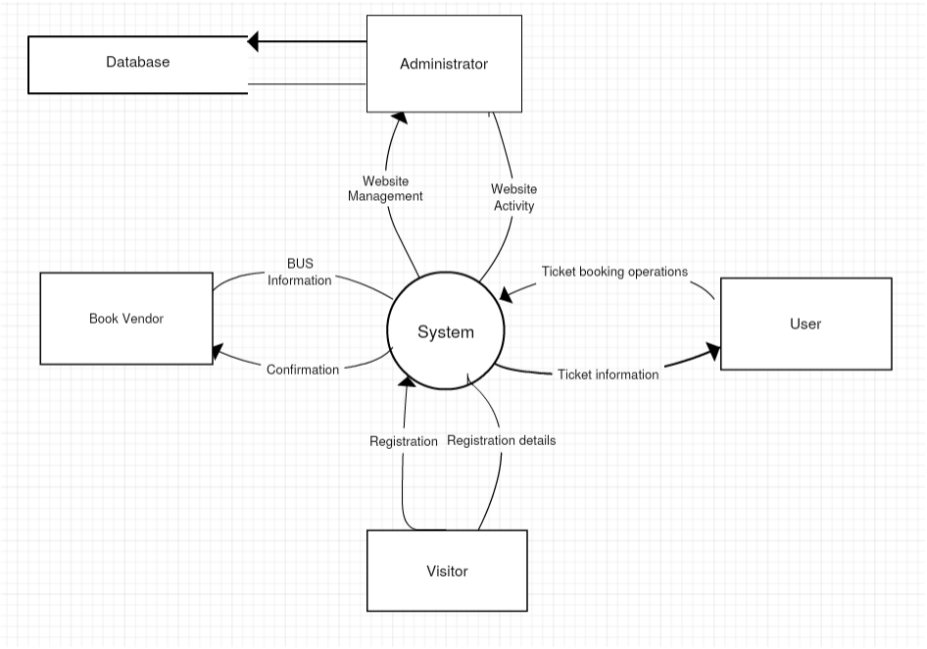

Figure 3 **Context diagram**

3.4 Data flow diagrams

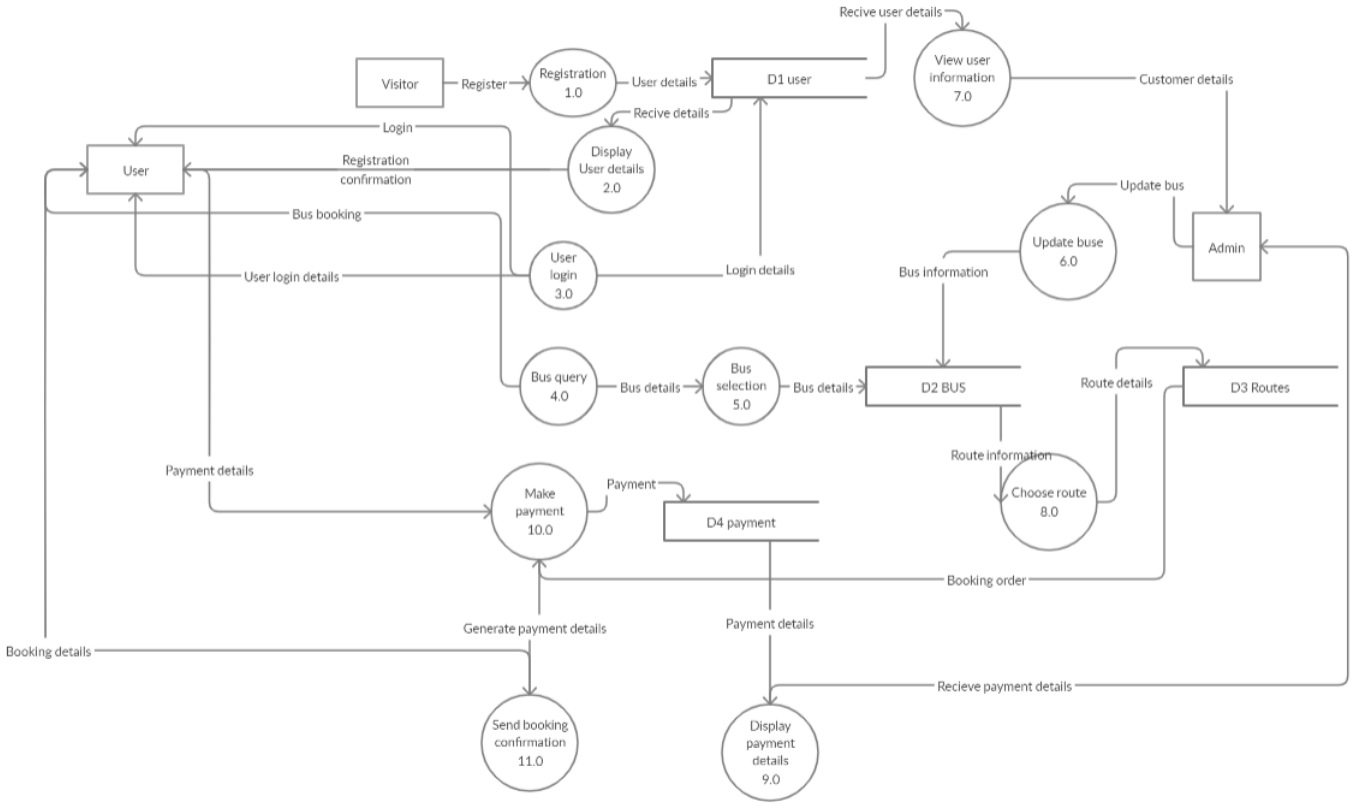

Figure 4 **Level 1 DFD for system**

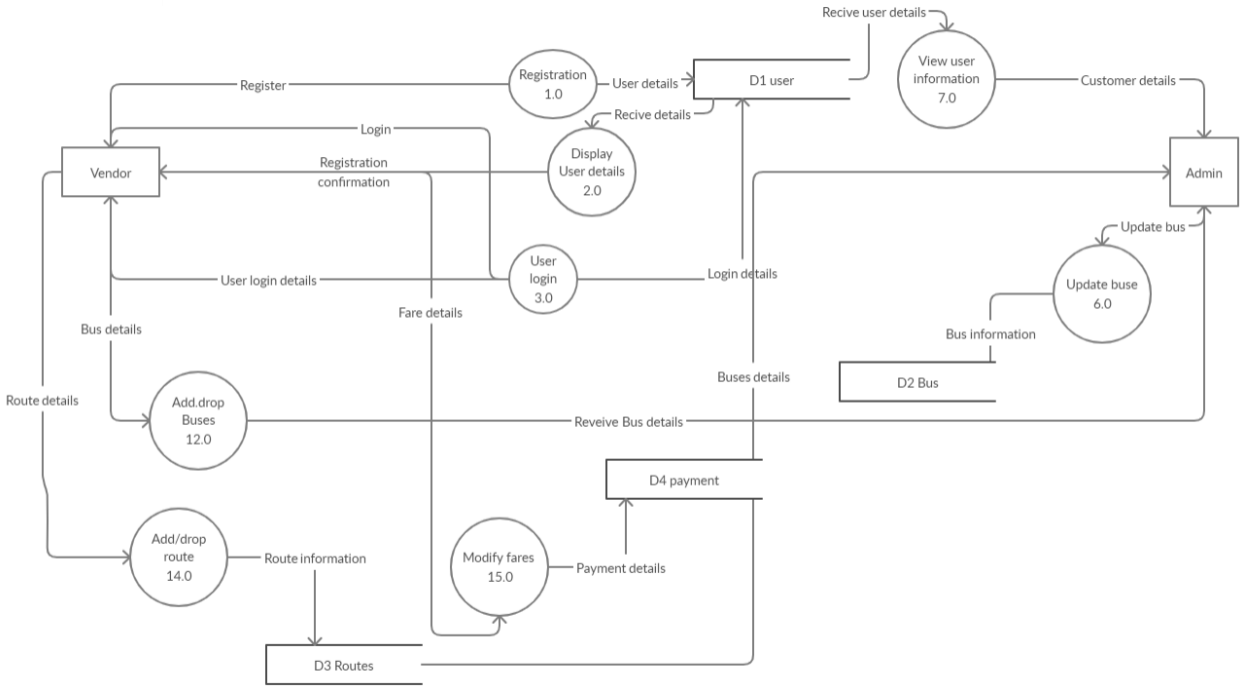

Figure 5 **Level 1 DFD for Bus Vendor**

3.5 Data dictionary

Table 1. Admin

Field	Type	NULL	Default
uname	varchar(30)	NO	None
Password	varchar(30)	NO	None
Name	varchar(30)	NO	None

Table 2. Bus

Field	Type	NULL	Default
Bid	varchar(11)	NO	None
Bname	varchar(50)	NO	None
Type_ac	Char(3)	NO	None
Type_sl	Char(3)	NO	None

Table 3. Card

Field	Type	NULL	Default
Num	varchar(16)	NO	None
type	varchar(50)	NO	None
Expdate	date	NO	None
cvv	Int(3)	NO	None
Bank	Varchar(30)	NO	None

Table 4. Net banking

Field	Type	NULL	Default
uname	varchar(30)	NO	None
password	varchar(30)	NO	None
bank	varchar(30)	NO	None

Table 5. User

Field	Type	NULL	Default
pid	varchar(11)	NO	None
name	varchar(20)	NO	None
email	varchar(50)	NO	None
Mob	Varchar(10)	NO	None

Table 6. Ticket

Field	Type	NULL	Default
Pnr	Int(11)	NO	None
rid	Int(11)	YES	NULL
Pid	varchar(11)	YES	NULL
status	Varchar(11)	YES	NULL
Dot	timestamp	NO	Current_timestamp

Table 7. Route

Field	Type	NULL	Default
rid	int(11)	NO	None
bid	varchar(11)	YES	NULL
fromloc	varchar(10)	YES	NULL
toloc	Varchar(10)	YES	NULL
fare	Double	YES	NULL
Dep_date	Date	NO	None
Dep_time	time	YES	NULL
Arr_time	time	YES	NULL
Arr_date	date	NO	None
Avalseats	Int(10)	NO	40
Maxseats	Int(10)	NO	40

4. Component design

4.1 Activity diagram

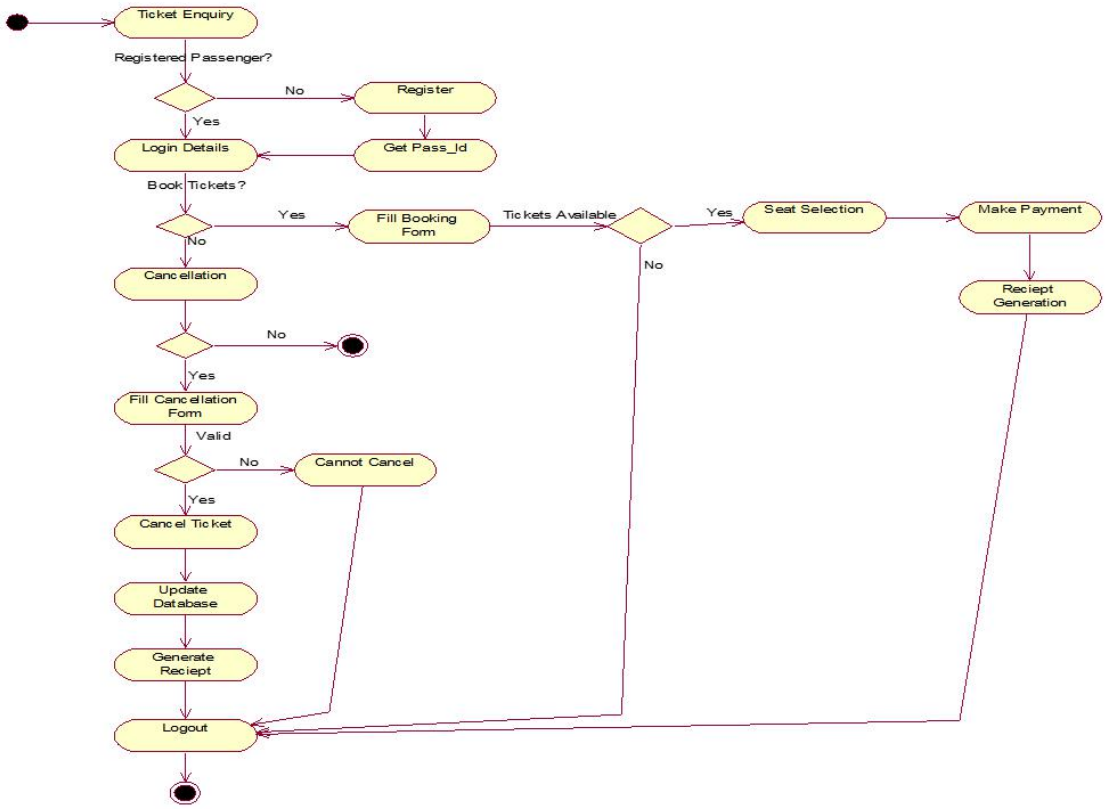

Figure 6 **Activity diagram of entire system**

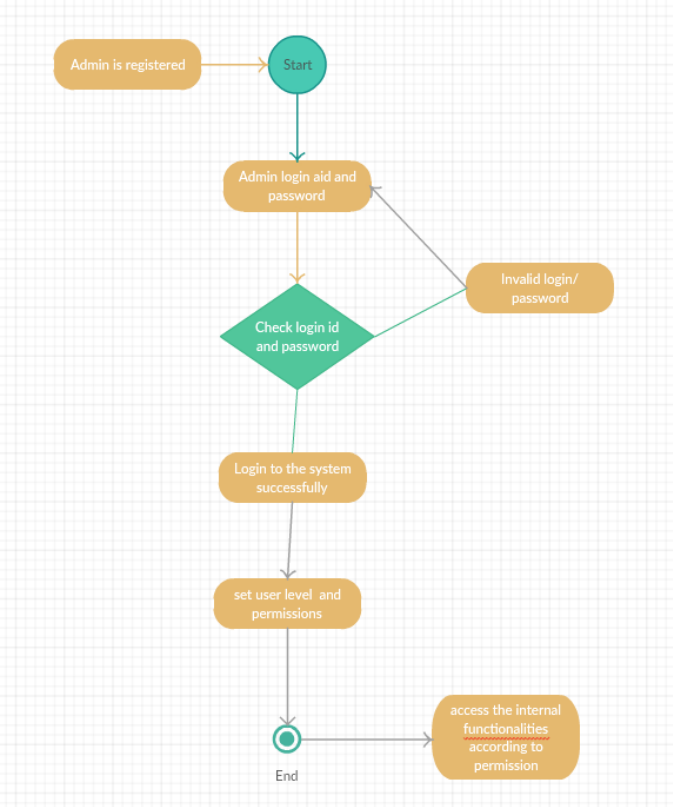

Figure 7 **Activity Diagram for admin**

5. User Interface Design

UI is designed according to UI design principles.

The structure principle: UI is organized in such a way that related things are combined together and unrelated things are separated.

The simplicity principle: It is easy to follow the provided interface. In the case of mistake, system displays error message.

The visibility principle: All system's functions are available through UI. It does not overwhelm users with too many alternatives.

The feedback principle: Through the system of messages, the design keeps users informed of actions, errors, or exceptions.

The reuse principle: In design, same names were used to perform the same operations with different objects in order to reduce ambiguity.

User interface will consist following main screens, login screen which will consist of a user dialogue box text boxes and three labels for data input. The login screen will be used to authenticate the user to the system.

The sample user interface forms and screens that the user will interact with include:

1. **Home page**: It is the main page for the system user can use it for navigation around the website.
2. **Login page**: It is the user logon page where user should enter the credentials required to book tickets.
3. **Booking page**: It is the booking history of user and user can see the history of tickets once logged in.
4. **Payment page**: This page is the section for payment and processing of ticket confirmation it is linked to third party payment vendor which user has no concern about.
5. **About us page**: This page describes the company details
6. **Contact us page**: Users can access this page to contact us in case of any miscommunications or feedback for the system
7. **Cancellation page**: Users should logon to access this page and displays the available tickets that are eligible to be cancelled.

8. Print tickets: Users can print the tickets once they are confirmed by accessing this page.

Figure 8 **cancellation page**

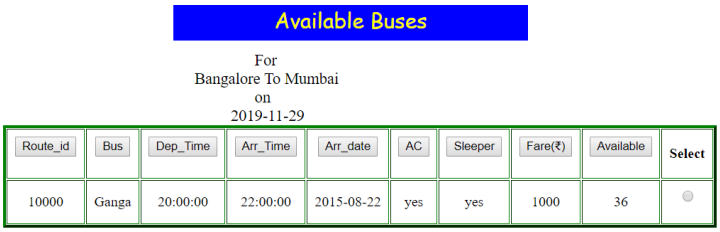

Figure 9 **Bus list page**

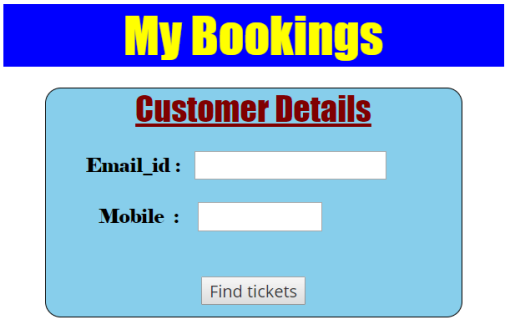

Figure 10 **Login page**

Figure 11 **Payment success page**

Figure 12 Booking page

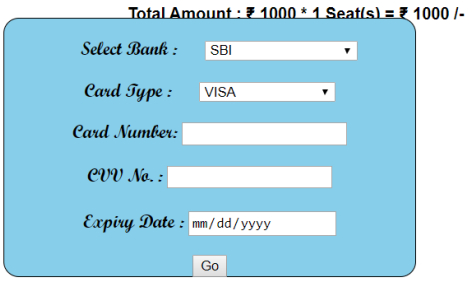

Figure 13 payment page

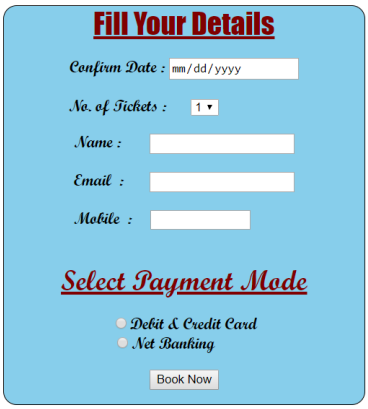

Figure 12 Payment details page

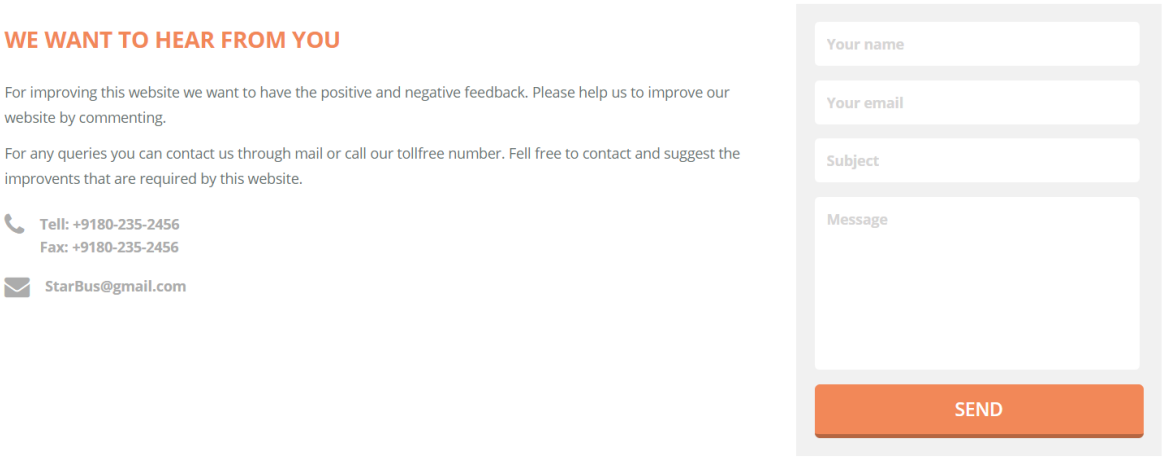

Figure 13 Feedback page