

**Bus Ticket Reservation System**

# “ GO

MOVE LEAVE HOME

# SAFE SAFE SAFE SAFE ”

Submitted ToLameya Islam Lecturer, Department of CSE,

Daffodil International University

Submitted By

Name : Avisheikh Kundu ID: 221-15-5009

Section: 61\_O Department of CSE,

Daffodil International University

Submission Date : 26-05-2024

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **1 Introduction** | **…………………………….** | **3** |
| 1.1 Background | **…………………………….** | 3 |
| 1.2 Objective | **…………………………….** | 3 |
| 1.3 Motivation | **…………………………….** | 3 |
| 1.4 Related Works/Review | **…………………………….** | 3 |
| 1.5 Gap Analysis | **…………………………….** | 3 |
| **2 System Architecture** | **…………………………….** | **3** |
| 2.1 System Overview | **…………………………….** | 3 |
| 2.2 System Components | **…………………………….** | 3 |
| 2.3 Architecture Diagram | **…………………………….** | 4 |
| **3 Project Features and Interface** | **…………………………….** | **4** |
| 3.1 List of Feature | **…………………………….** | 4 |
| 3.2 User Interfaces | **…………………………….** | 4 |
| 3.3 Input Output Demo | **…………………………….** | 4 |
| **4 System Implementation** | **…………………………….** | **4** |
| 4.1 Development Tools & Technologies | **…………………………….** | 4 |
| 4.2 Implementation Plan | **…………………………….** | 5 |
| 4.3 Testing and Validation | **…………………………….** | 5 |
| **5 Future Scope and Limitation** | **…………………………….** | **5** |
| 5.1 Limitation | **…………………………….** | 5 |
| 5.2 Future Scope | **…………………………….** | 5 |
| 5.3 Conclusion | **…………………………….** | 5 |
| **References** | **…………………………….** | **5** |

## 

## 1. Introduction

### 1.1 Project Overview

The Bus Ticket Management System is a software application designed to facilitate the booking, cancellation, and management of bus tickets. Originally developed in C, the system has been converted to Python using object-oriented programming principles to enhance maintainability and scalability.

### 1.2 Objectives

* To provide a user-friendly interface for booking and managing bus tickets.
* To ensure efficient and accurate handling of ticket reservations and cancellations.
* To maintain an organized database of customer information and bus schedules.

## 2. System Requirements

### 2.1 Hardware Requirements

* A computer with at least 2GB RAM and 1GHz processor.
* Minimum 100MB of free disk space.

### 2.2 Software Requirements

* Python 3.6 or higher.
* Any operating system that supports Python (Windows, macOS, Linux).
* Text editor or IDE (e.g., VSCode, PyCharm).

## 3. System Design

### 3.1 Architecture

### The system is built using a modular approach with the following key components:

* **User Interface:** Handles user inputs and displays relevant information.
* **Reservation Module:** Manages booking and cancellation of tickets.
* **Database Module:** Maintains records of buses, seats, and customer information.

### 3.2 Class Diagram

**BusTicketManagementSystem**

│

├── BusTicketManagementSystem

│ ├── login()

│ ├── bus\_lists()

│ ├── display\_seat(bus)

│ ├── reservation\_info(root, cust\_id)

│ ├── insert(root, cust\_id)

│ ├── cancel(random\_num)

│ ├── cost(passenger)

│ ├── status()

│ ├── main()

│

└── **Passenger (Inner Class)**

├── \_\_init\_\_(self, cust\_id, name)

├── cust\_id

├── name

├── left

├── right

## 4. Implementation

### 4.1 Technologies Used

* **Python:** The primary programming language used to implement the system.
* **Standard Libraries:** random, time, os, sys.

### 4.2 Key Functions

#### 4.2.1 login()

Handles user authentication by comparing input credentials with stored credentials.

#### 4.2.2 bus\_lists()

Displays the list of available buses along with their destinations, charges, and schedules.

#### 4.2.3 display\_seat(bus)

Displays the seating arrangement of a specified bus, indicating booked and available seats.

#### 4.2.4 reservation\_info(root, cust\_id)

Retrieves and displays reservation information for a given customer ID.

#### 4.2.5 insert(root, cust\_id)

Inserts a new reservation into the binary search tree based on customer ID.

#### 4.2.6 cancel(random\_num)

Cancels a reservation by validating the reservation number and updating the seating arrangement.

#### 4.2.7 cost(passenger)

Calculates the ticket cost based on the bus number.

#### 4.2.8 status()

Displays the seating arrangement for all buses.

#### 4.2.9 main()

The main function that drives the program, presenting the main menu and handling user choices.

## 5. Testing

### 5.1 Test Cases

* **Login Test:** Ensure that the system correctly validates user credentials.
* **Bus List Test:** Verify that the list of buses is displayed correctly.
* **Booking Test:** Test booking functionality by reserving seats and checking seat availability.
* **Cancellation Test:** Ensure that booked seats can be successfully cancelled.
* **Cost Calculation Test:** Validate the cost calculation based on bus numbers.

### 5.2 Results

All test cases were executed successfully, demonstrating that the system meets the specified requirements and functions as intended.

## 6. Conclusion

The Bus Ticket Management System successfully converts the original C program into a Python-based application, leveraging object-oriented principles to enhance the design. The system provides a comprehensive solution for managing bus ticket reservations and cancellations, ensuring efficiency and user satisfaction.

### 6.1 Future Enhancements

* **Database Integration:** Replace in-memory data structures with a persistent database.
* **Web Interface:** Develop a web-based interface for easier access and broader usability.
* **Payment Integration:** Implement an online payment system for ticket purchases.

## 7. Appendix

### 7.1 Source Code

Source Code : [Code](mailto:https://github.com/AvisheikhKundu/Bus_Ticket_Reservation_System.py/blob/main/Main.py)