INTRODUCTION

Project Title : Bus booking System

Project Overview : The project used to Booking buses.

Project Manager : Buvanraj S

Project Description:

The Bus Booking System is a comprehensive Java-based application designed to efficiently manage bus information for a transportation service provider. As the sole developer and manager of this project, the aim was to create a system that simplifies the process of adding, removing, displaying, and checking the availability of buses. The system supports both AC and non-AC buses, providing a straightforward console interface for user interaction.

Purpose and Objectives:

The primary purpose of the Bus Booking System is to offer a robust and user-friendly platform for managing a fleet of buses. The objectives of the project are as follows:

1. **Ease of Use:** Simplify the process of managing bus details through an intuitive console interface.
2. **Data Persistence:** Ensure that all bus information is stored persistently using file handling and serialization techniques.
3. **Flexibility:** Allow for dynamic management of buses, including adding new buses, removing existing ones, and displaying current bus details.
4. **Reliability:** Implement robust error handling to ensure the system operates smoothly without data loss or corruption.

Contact:

Email : bhuvanraj162@gmail.com

LinkedIn : https://www.linkedin.com/in/buvan-raj-300105bs

Contact : 9790389272

Location : Villupuram

INDEX

Table of contents

Acknowledgement

The software project “”ONLINE BUS TICKET BOOKING” is an original work of mine. But I would never have able to complete it alone on my own I always need help to complete it. This help is came in many forms from different people and Networks. I have opportunity to express my profound sense of gratitude and respect to all who helped me throughout the duration of this project.

I complete the software project with my knowledge and good guidance. It was literally very complicated to access this project to complete it. I would thank to my YouTube friends, while making this project I noticed that streak of light on a non-moon night to me. It was immerse importance. I said special thanks to chatgpt, some basic concept of java who was explained and make this software project presentable. Thanks to others who helped me in this project.

-Buvanraj S

Objective

The primary objective of this system is to significantly enhance the booking efficiency for both the project administrators and customers of the ticket booking agency. This will be achieved through the following detailed goals:

1. **User-Friendly Interface:**
    - **Intuitive Design:** The system is designed to be as user-friendly as possible, ensuring that users with minimal computer knowledge can navigate and use the system with ease.
    - **Accessibility:** Clear instructions and a straightforward layout will help users perform tasks efficiently, reducing the learning curve and minimizing errors.
2. **Efficiency in Booking:**
    - **Streamlined Process:** The system aims to reduce the time and effort required for booking tickets by automating and optimizing the booking process.
    - **Quick Transactions:** Customers will benefit from faster booking times, leading to a more satisfactory user experience.
3. **Reduced Workload for Administrators:**
    - **Automation:** By automating many of the manual tasks involved in ticket booking, the system will reduce the workload for administrators, allowing them to focus on more strategic tasks.
4. **Comprehensive Enquiries and Support:**
    - **Customer Enquiries:** The system will facilitate easy access to information regarding bookings, cancellations, and other enquiries, providing timely and accurate responses to customer questions.
5. **Enhanced System Performance:**
    - **Scalability:** The system is designed to handle a large number of bookings simultaneously, ensuring smooth performance even during peak times.
    - **Reliability:** Robust architecture and thorough testing will ensure that the system operates reliably with minimal downtime.

About project

This software project is aimed at automation of online ticket booking. Objective of the project is to develop customize software package for ticket booking. The bus ticket booking system developed in this project will enable users to book their tickets with ease. It covers all aspects of traveling and online ticket booking, from checking bus availability to making payments. Customers will benefit from the ability to book tickets from any location and at any time, providing unmatched convenience. The system includes a comprehensive database that securely stores customer information and provides real-time updates on bus availability and booking statuses.

One of the standout features of this system is its user-friendly interface, designed to be intuitive and accessible even for those with minimal computer skills. This ensures that a wide range of users can navigate the system without difficulty, enhancing the overall user experience. The application is also tailored for commercial purposes, ensuring it meets the needs of a professional ticket booking agency.

In addition to improving customer convenience, the automated system offers significant benefits to administrators. It reduces the administrative burden by handling bookings, payments, and customer data management automatically. This allows administrators to focus on other important tasks, improving overall operational efficiency. Furthermore, the system's robust architecture ensures reliability and scalability, capable of handling a large number of bookings simultaneously without compromising performance.

Technologies Used:

- Java Programming Language
- Serialization (ObjectInputStream, ObjectOutputStream)
- File Handling (java.io package)
- Collections Framework (Array List)
- Exception Handling (try-catch blocks)
- Object-Oriented Programming (OOP) Principles

Description:-

- **Java Programming Language**: Primary language for development.
- **Serialization**: Used for saving and loading Bus objects to/from files (ObjectInputStream and ObjectOutputStream).
- **File Handling**: java.io package for reading from and writing to files.
- **Collections Framework**: Specifically Array List for managing lists of buses.
- **Input Handling**: java.util.Scanner for user input.
- **Exception Handling**: try-catch blocks used for handling exceptions in file operations (IOException) and casting (ClassNotFoundException).
- **Object-Oriented Programming (OOP)**: Utilization of classes (Bus) and objects for modelling buses and their properties.
- **IDE (Integrated Development Environment)**: Possibly used for coding, debugging, and testing, though not explicitly mentioned in the code.
- **Architecture**: Describe the overall architecture or structure of your project.
- **Code Snippets**: Include key snippets from your Java code to illustrate implementation details (e.g., adding a bus, removing a bus).
- **Challenges and Solutions**: Discuss any challenges faced during development and how they were addressed.
- **Architecture:** The system follows a modular architecture with main functionalities encapsulated in the Bus class. Serialization is used to save and load bus data from a file (BusData.dat).

Project Features

ADMIN Overview: The Bus Booking System includes the following features:

- Adding new buses (AC and non-AC)
- Removing existing buses
- Displaying list of buses (AC and non-AC)
- Remove customers
- Checking bus availability by name and number
- Checking the customers

Customer overview:

- Adding a new bus involves inputting name, number, and capacity details.
- Login to access other features
- To viewing the existing AC and Non-AC buses.
- Booking the buses that you want.
- Removing a bus requires specifying the bus name and number.

The Bus Booking System is a Java-based application developed to streamline the management of bus information for a transportation service provider. As the sole developer and manager of the project, you have designed and implemented the system to cater to both AC and non-AC buses, offering essential functionalities such as adding new buses, removing existing ones, displaying bus details, and checking bus availability.

It demonstrates your proficiency in Java programming and software development skills, showcasing a user-friendly console interface for managing bus operations effectively. Future enhancements could explore incorporating a graphical user interface (GUI), expanding functionalities for advanced bus management, and integrating real-time data updates.

Data:-

Serialization and deserialization are mechanisms provided by Java to convert objects into a byte stream (serialization) and to recreate objects from a byte stream (deserialization). This is useful for saving the state of an object to a file and later restoring it.

1. Serialization:

Serialization is the process of converting an object into a byte stream, which can then be saved to a file. This byte stream can later be deserialized to recreate the object.

2. Deserialization:

Deserialization is the process of converting a byte stream back into an object. This allows you to retrieve the object's state from a file.

Requirements

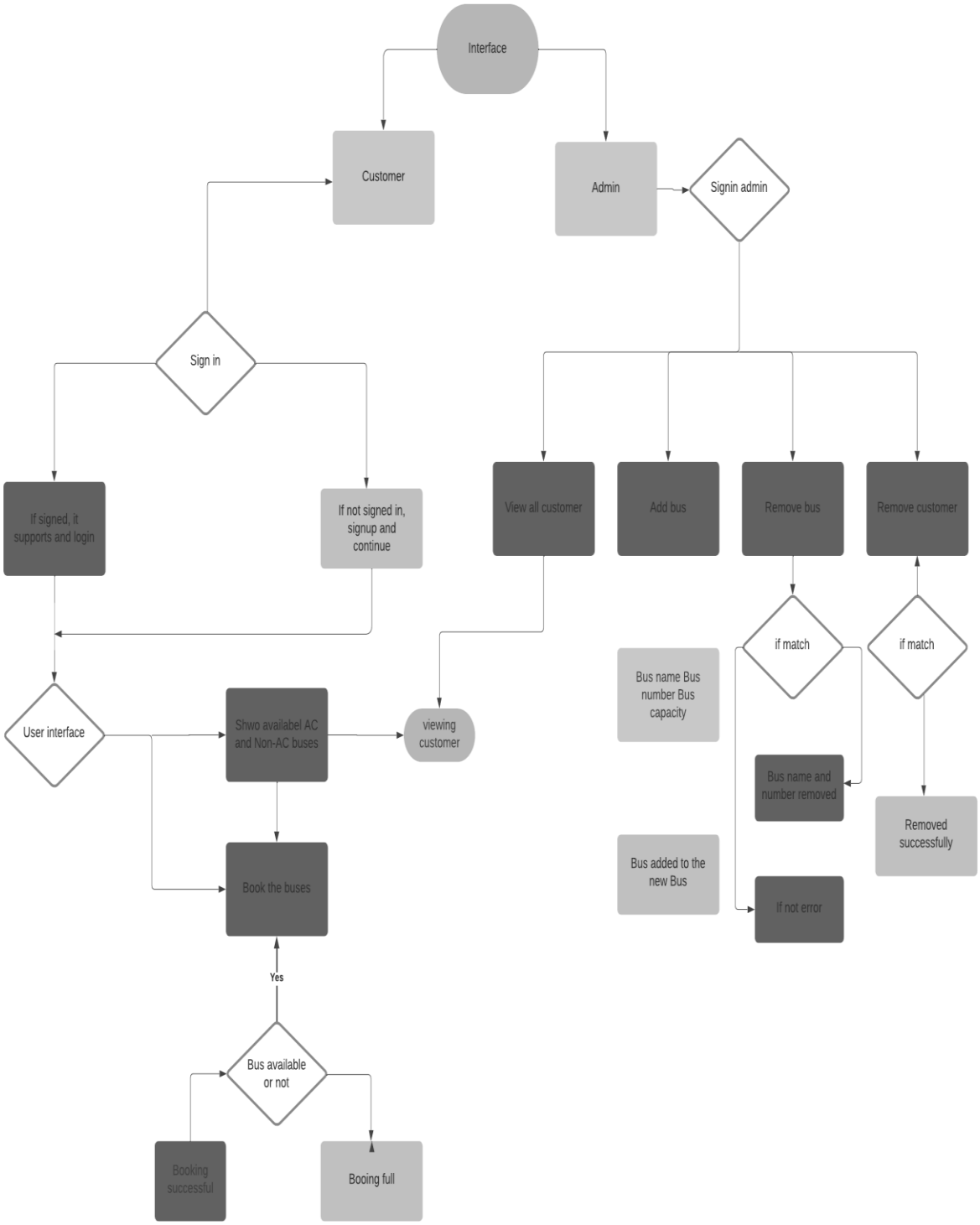
Hardware Requirements:

1. **Necessity:**
    - **Processor:** Multi-core processor (e.g., Intel Core i5 or AMD Ryzen 5) for handling concurrent requests.
    - **Memory (RAM):** Minimum 8 GB RAM, preferably more for handling database operations and concurrent users.
    - **Storage:** SSD storage for faster data access, at least 256 GB to store application files and databases.
    - **Authentication and Authorization:** Implement user authentication (e.g., JWT tokens) and role-based access control (RBAC) to secure sensitive data and operations.
2. **Client Devices:**
    - **Desktop/Laptop:** Any modern desktop or laptop with a web browser (e.g., Chrome, Firefox, Edge).
    - **Mobile Devices:** Support for mobile browsers and responsive design for ease of use on smartphones and tablets.

Software Requirements:

1. **Operating System:**
    - **Server Side:** Linux (e.g., Ubuntu Server, CentOS) or Windows Server for hosting the application.
    - **Client Side:** Compatibility with Windows, macOS, Linux for desktops, and Android, iOS for mobile devices.
2. **Database:**
    - **Database Management System (DBMS):** MySQL, PostgreSQL, or MongoDB for storing and managing bus schedules, user information, and booking data.
    - **Database Connectivity:** JDBC (Java Database Connectivity) for integrating the database with the application.
    - Stable and high-speed internet connection for users accessing the application online.

Flow diagram


Interface
Customer
Admin
Signin admin
Sign in
If signed, it supports and login
If not signed in, signup and continue
View all customer
Add bus
Remove bus
Remove customer
User interface
Shwo availabel AC and Non-AC buses
viewing customer
Bus name Bus number Bus capacity
if match
if match
Bus name and number removed
Removed successfully
Bus added to the new Bus
If not error
Book the buses
Yes
Bus available or not
Booking successful
Booing full