**Using AVL-TREE**

**Bus Management System**

**BusBookerPro**

# **PRODUCT DESCRIPTION:**

Our product, **BusBookerPro**, is a real-time bus booking system designed to enhance the user experience by providing up-to-the-minute availability. Upon logging into the system, users will be presented with bus schedules starting from their login time, ensuring that they only see upcoming buses. This dynamic approach simplifies the booking process by filtering out past bus timings, thereby streamlining the user's interaction with the system and ensuring that only relevant options are displayed.

# **PRODUCT FEATURES AND OUTPUTS:**

## **PRODUCT FEATURE NO\_1)**

## Our system operates in real-time, ensuring that users are presented only with bus schedules available from their current login time onward. This functionality effectively saves users time and eliminates any ambiguity by filtering out bus timings that have already passed relative to their current or previous login sessions. As a result, users can make informed decisions quickly and without confusion.

## **PRODUCT FEATURE NO\_2)**

We are providing our travel services for the following timelines only which are:

1. 12:00 hours.
2. 03:00 hours.
3. 06:00 hours.
4. 09:00 hours.

The user can book his ticket only with these available timelines.

## **PRODUCT FEATURE NO\_3)**

The complete program is based on four classes with the given timelines as explained in the above paragraph. With this, we are giving the user the ability to book him/herself a ticket via our console.

## **PRODUCT FEATURE NO\_4)**

Our program initially prompts the user to specify the number of seats they wish to book. The system then verifies the requested number of seats against the availability in the selected bus. Upon successful validation, the program displays a visual representation of the bus seating arrangement, utilizing color codes for clarity: white indicates available seats, red indicates the selected seats, and green indicates reserved seats.

### **OUTPUT\_SCREEN)**

A picture containing text, keyboard, electronics, calculator

Description automatically generated

## **PRODUCT FEATURE NO\_5)**

Our program also supports a refunding system. For this we are asking users name and ticket no. by this intel we are undoing the reserved tickets and making them available for booking again.

## **PRODUCT FEATURE NO\_6)**

Our program is also providing the VIEW STATUS, which provides the intel about all the passengers information on our booked buses, intel like their names and booking ticket number.

# **DATA STRUCTURES ALGORITHMS WHICH ARE USED HERE:**

We have implemented insertion, deletion, and searching functionalities using AVL trees in our bus booking system. This choice ensures that the operations can be performed with a time complexity of O(log(n)), significantly enhancing efficiency compared to an O(n) complexity. The AVL tree maintains balance through self-balancing properties, which ensures that users experience quick and responsive booking and refund processes. Pre-existing data is stored in these trees to provide a realistic and authentic user experience.

Additionally, we have employed a stack data structure to manage and display selected or reserved seats. Direct data retrieval from the AVL tree presented challenges due to the complexity of recursive calls. By pushing seat numbers onto the stack, we simplify the process and achieve constant time complexity, O(1), for these operations. This approach avoids the overhead associated with recursion and ensures a smooth and efficient user interface.

### **Benefits of Using AVL Trees:**

1. **Efficiency**: Ensures operations are completed in O(log(n)) time, improving the speed of booking and refunds.
2. **Balanced Operations**: Maintains a balanced tree structure, preventing performance degradation.
3. **Authenticity**: Preloaded data in AVL trees enhances the realism and reliability of the system.