**Restaurant Reservation System**

* Overview

This is a simple Restaurant Reservation System implemented in Java. The system allows users to make reservations, cancel reservations, reschedule reservations, and manage a waitlist for customers when there are no available tables. The reservation details are stored in a binary search tree for efficient table size lookup, and a hash table is used for quick reservation lookup by customer name.

* Project Structure

- **Reservation**: Represents a reservation with customer name, date, time, and table size.

- **TableNode:** Represents a node in the binary search tree used for table management.

- **RestQueue**: Implements a simple queue for managing the waitlist.

- **ReservationManagementTable**: Manages reservations using a hash table for quick lookup.

- **RestaurantReservationSystem**: Main class implementing the core functionality of the system.

- **RestaurantReservationApp**: Interactive console application for user interaction.

* Features

1. **Make Reservation**:Users can make a reservation by providing their name, date, time, and the desired table size.

2. **Cancel Reservation**: Users can cancel an existing reservation by providing their name.

3. **Reschedule Reservation:** Users can reschedule an existing reservation by providing their name and new date, time, and table size.

4**. Display Occupancy:Shows** an overview of the current occupancy, including reserved tables and their details.

5. **Display Waitlist**:Displays the current waitlist with customer names.

6. **Add to Waitlist**:Adds a customer to the waitlist, even if there are no available tables.

7. **Display Reservation**: Allows users to look up and display details for a specific reservation by providing the customer name.

8. **Exit**:Exits the application.

* Usage

1. Run the `RestaurantReservationApp` class.

2. Choose from the menu options to perform various actions.

* Notes

- The system uses a binary search tree for efficient table size lookup when making reservations or canceling them.

- Rescheduling a reservation checks for available tables before updating the reservation.

- The waitlist is managed using a simple queue data structure.

* Dependencies

- Java 8 or higher.

* Authors

-Samuel Dartey-Baah

James Cole

Godhold Alomenu

License

This project is licensed under Ashesi University .