**Introduction**

The Venue Booking System is a software that helps people book event venues easily. It connects users looking for venues with venue owners. The goal is to make reserving venues, like for weddings or parties, straightforward. Users can see venue options, check if they're available, and book them. Venue owners, known as agents, can add venues, manage bookings, and see reserved venues. This report covers the system's features, how it works, and how it uses object-oriented programming. It improves the experience for both users and venue owners.

**Features and Functionality**

The Venue Booking System offers a wide range of features and functionalities that cater to both users and venue owners. These features ensure a smooth and efficient experience when booking venues and managing reservations.

1. **User Registration and Profile Management**: Users can create accounts with their personal information, including name, mobile number, and email. Each user receives a unique user ID, making it easier to identify and manage bookings.
2. **Venue** **Catalog**: The system provides a catalog of available venues, listing their names, locations, and pricing. Users can view the availability status of each venue on specific dates.
3. **Venue Booking**: Users can search for venues by name, location, or date, making it easy to find suitable options. Bookings are made by specifying the venue and the desired date.
4. **Venue Management (Agent Portal)**: Venue owners or agents can add new venues to the system, including details like venue name, location, available dates, and pricing. They can also view a list of all booked venues and their respective dates.
5. **Persistence and Data Storage**: User and venue data are stored in CSV files to ensure data integrity and availability across sessions.
6. **Data Retrieval and Display**: The system retrieves user and venue data from CSV files to display relevant information to users and agents.
7. **User-Friendly Interface**: The system's interface is designed with simplicity in mind, making it easy for users and venue owners to navigate and utilize its features.

The Venue Booking System effectively bridges the gap between those seeking event venues and venue owners, offering a feature-rich experience for all involved parties. Users can quickly locate and reserve suitable venues, while venue owners can effortlessly manage their offerings and bookings. Next, we will explore the object-oriented programming concepts used to build and enhance the system's functionality.

The code for the Venue Booking System is structured as a Java program with object-oriented design. Let's break down the structure and flow of the code:

1. **Import Statements**: The code begins with import statements to include necessary Java libraries for file handling, data structures, and input/output.

2. **ReservationItem Class:**

- This is a base class that defines common attributes for both venues and users.

- It contains fields for 'id', 'name,' and 'location.'

- There are constructors for initializing these attributes.

3. **Venue Class:**

- Venue extends ReservationItem, inheriting its attributes (id, name, location).

- Additional attributes for venues include 'vid' (venue ID), 'date' (a map of dates and availability status), and 'price.'

- There are constructors to initialize these attributes.

- It also has a method 'cancelBooking' to handle the cancellation of bookings.

4. **User Class**:

- User also extends ReservationItem and inherits the basic attributes.

- User-specific attributes include 'uid' (user ID), 'u\_name' (user name), 'mobile,' 'email,' 'bid' (booking ID), and 'bdate' (booking date).

- Constructors are provided for various scenarios, such as user creation and booking.

- User class also has a 'cancelBooking' method to handle booking cancellation.

5. **Agent Class:**

- Agent is a class responsible for managing venues and performing operations such as adding venues and listing booked venues.

- It also has methods to read data from and write data to CSV files.

- The 'addvenue' method is used to add a new venue to the system.

- 'listbookedvenue' is used to display booked venues.

- 'write\_to\_file' is used to write venue data to a CSV file.

- 'add' allows an agent to add multiple venues.

- 'listallvenue' displays all available venues.

- 'readfromcsv' reads venue data from a CSV file.

6. **Test Class:**

- The 'Test' class contains the main method and is the entry point of the program.

- It initializes instances of Venue, User, and Agent classes.

- User data is read from a CSV file into the 'first' User instance.

- Venue data is read from a CSV file into the 'list' Venue instance.

- The code then enters a loop for the main menu.

- Users can choose between the 'User Portal,' 'Agent Portal,' or 'Exit.'

7. **User Portal:**

- In this section, users can register, book venues, and return to the main menu.

- Registration includes entering user details and saving them to a CSV file.

- Users can book venues by specifying a venue ID and a date.

- The 'book\_venue' method is used to handle booking and cancellation.

8. **Agent Portal:**

- Agents can add venues, list booked venues, list all venues, and return to the main menu.

- Venue information is collected from agents and saved to a CSV file.

- Agents can also view the list of booked venues.

9. **Exiting the Program:**

- Users can exit the program by choosing the 'Exit' option from the main menu.

Overall, the code follows an object-oriented structure, with classes for Venue, User, and Agent, and provides functionalities for venue booking, user management, and agent actions. Data is read from and written to CSV files for persistence. The code is organized into methods that encapsulate specific actions, making it easy to understand and maintain.

In the Venue Booking System code, CSV file reading and writing operations are used to store and retrieve data. This allows the system to persist data, making it available even after the program is closed. Here's how CSV file reading and writing are used in the code:

**Concepts used in the code:**

The code you provided demonstrates the use of various Java concepts and object-oriented programming principles. Here are some of the key Java concepts used in the code:

1**. Classes and Objects**: The code defines classes like `ReservationItem`, `Venue`, `User`, and `Agent`, which are used to create objects and represent different entities in the system.

2. **Inheritance:** The `Venue` and `User` classes inherit from the `ReservationItem` class, showcasing inheritance in Java.

3. **Constructor Overloading**: Multiple constructors are defined for some classes, allowing for flexibility when creating objects.

4. **Data Structures**: The code uses data structures like HashMap to manage venue booking dates.

5. **File Handling**: Java's file handling mechanisms are used to read from and write to CSV files for venue and user data storage.

6. **User Input Handling**: The code takes user input using the `Scanner` class, providing interaction with the user.

7.**Control Structures**: Control structures like loops and conditional statements are used throughout the code for decision-making and iteration.

8. **Exception Handling**: The code does not handle exceptions explicitly but should include error handling for a complete system.

9. **CSV File Handling**: Reading and writing data in CSV file.

These concepts collectively demonstrate a fundamental understanding of Java programming and object-oriented design principles in the development of the Venue Booking System.

**File Handling:**

1. **Writing Data to a CSV File**:

- Data is written to a CSV file using the `FileWriter`, `BufferedWriter`, and `PrintWriter` classes.

- When a new venue is added or when user data is created, the relevant information is written to the respective CSV files.

These methods open the CSV files in append mode (`FileWriter(file, true)`) and write data in a structured format, as defined by the CSV file's column headers.

2. **Reading Data from a CSV File**:

- Data is read from CSV files using the `BufferedReader` and `Files` classes.

- When the program starts, the user data and venue data are loaded into memory from their respective CSV files.

These methods open the CSV files in read mode (`Files.newBufferedReader(pathToFile, StandardCharsets.US\_ASCII)`) and parse the data to create instances of `Venue` and `User` objects.

The CSV files used in the code follow a structured format with specific column headers. When writing data, the code formats the data in a way that matches the expected structure of the CSV file. When reading data, it parses the data accordingly.

Overall, the combination of reading and writing data to CSV files allows the system to maintain data persistence between program runs. This is particularly useful for storing information about venues, bookings, and users, ensuring that the data is not lost when the program is closed and can be used for future interactions.

**Java FileWriter** class of java.io package is used to write **data in character** form to file. Java FileWriter class is used to write character-oriented data to a file. It is a character-oriented class that is used for file handling in java.

**APPLICATIONS**

**Venue Listing and Rental Marketplace**: You can create an online marketplace where venue owners or hosts list their spaces for rent, and users can book them for various purposes, such as parties, photo shoots, or workshops.

**Event Management System**: The code can be used to manage and book venues for different events, such as weddings, conferences, concerts, and parties. Event organizers can use the system to find and reserve suitable venues.

**Hotel Room Booking System**: With some modifications, this code can be adapted for hotel room bookings. Users can check room availability and make reservations.

**Restaurant Table Reservations**: Restaurants can implement table reservation functionality for customers who want to book a table in advance.