ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

UNIVERSITY FOR INDUSTRY

ADDIS ABABA SCIENCE AND TECHNOLOGY UNIVERSITY

College of Mechanical and Electrical Engineering

Department of Software Engineering

Movie Ticketing System

Section: D

Group Members	ID.No
1. **Miraf Debebe**	**ETS1110/14**
2. **Misganaw Tesfaye**	**ETS1112/14**
3. **Natnael Abayneh**	**ETS1237/14**
4. **Nahom Tensaebirhan**	**ETS1142/14**

Submitted To: Inst. Abdi M.

Submission Date: 01/23/2023

Table Of Contents

Background

A movie ticketing system is a software application that allows users to book movie tickets online. The system is designed to provide a convenient and efficient way for users to search for movies, view show times, select seats, and make secure online payments. This saves time and effort, providing convenience and flexibility to moviegoers. The traditional method has several disadvantages like long queues, limited availability, inconvenience, limited payment options and information, and missed promotions or discounts. To address these challenges, online movie ticketing systems have emerged as a more convenient and efficient alternative. Moreover, the system provides real-time updates on seat availability ensuring that users have accurate information and make informed decisions before purchasing tickets.

For our project, we aim to develop a robust and user-friendly online movie ticketing system that leverages the power of Java programming language to deliver a seamless and efficient ticketing experience for both customers and theater operators. Our system will be designed to handle a high volume of concurrent users, support secure payment transactions, and provide real-time information about movie show times, availability, and seating options. We will also focus on creating a responsive and intuitive user interface that makes it easy for customers to browse movies, select show times, and complete their purchases with minimal effort.

In conclusion, the background of the movie ticketing project highlights the growing demand for convenient and efficient ticket purchasing in the entertainment industry. With the rise of online platforms and mobile applications, consumers have come to expect seamless and user-friendly experiences when it comes to buying tickets for movies. An online movie ticketing system for our project holds great potential to revolutionize the way customers purchase tickets for their favorite movies. By offering convenience, flexibility, and efficiency, our system aims to address the limitations of traditional ticket purchasing methods and provide a seamless and hassle-free experience for both customers and theater operators. This project aims to address these needs by developing a comprehensive and intuitive ticketing system that enhances the overall movie-going experience for customers. It seeks to revolutionize the way people access and enjoy movies.

Statement of the problem

The traditional method for movie ticket purchasing presents several significant challenges for moviegoers including inconvenience, uncertainty about seat availability, and lack of flexibility. This leads to frustration and dissatisfaction among consumers, ultimately impacting their overall movie-going experience. These problems highlight the need for more efficient and user-friendly solutions. By addressing these issues, the system aims to provide users with a convenient, and accessible movie ticket-purchasing experience.

It is also important to understand the specific problems faced by users in the traditional movie ticket purchasing process hinder the potential for increased ticket sales and customer retention for movie theaters. Therefore, there is a pressing need to develop a solution that streamlines the ticket purchasing process and provides a seamless and convenient experience for moviegoers.

Project Objectives

General Objectives:

- Efficient Ticketing Process: Develop a system that streamlines the process of booking movie tickets, providing a convenient and user-friendly experience to customers.
- Enhanced User Experience: Create a platform that allows users to explore movie options, select preferred seats, and complete the booking process with ease, enhancing their overall movie-going experience.

Specific Objectives:

- User Registration and Authentication: Implement a robust user registration and login system to facilitate user access and personalized experiences.
- Comprehensive Movie Information: Provide detailed movie listings, including genre filtering, movie summaries, cast details, and Show Time availability to assist users in making informed choices.
- Seamless Seat Selection and Booking: Enable users to view available seats, select seat preferences, and finalize ticket purchases within a few simple steps, ensuring a hassle-free booking process.
- Booking Confirmation and Management: Offer users the ability to confirm bookings, view and download booking details, and obtain digital booking confirmations for easy access.

Scope

This project aims to solve the problem of inconvenience and inefficiency of buying movie tickets at a cinema. Users have to wait in long queues, face the risk of tickets being sold out, and have limited choices of seats and movies. The project will provide a convenient and efficient way of booking movie tickets online, saving users time and money. It assumes that the cinema will provide a method (file-based) for fetching data about the relevant services and information.

Our project scope encompasses the development and implementation of a Movie Ticketing System with the following key features:

- User Management: User registration, login, and profile management functionalities.
- Seat Selection and Reservation: Displaying seating layouts, enabling seat selection, and managing seat reservations.
- User Interface: Creation of an intuitive and visually appealing interface to improve user interaction and engagement.

Limitations

- Require an internet connection to access and use.
- Has to comply with data protection and privacy laws and regulations.
- Use a third-party service to provide certain services(payment, security).

Requirement Analysis

Stakeholders:

- Customers
- Website administrators
- Third-party Service providers

Functional Requirements:

1. User Registration and Authentication:
    - Customers must be able to create accounts and log in.
    - User authentication should be secure.
2. Movie Catalog:
    - Display a list of movies currently playing or coming soon in the cinema.
    - Customers can search for movies by title, genre, rating, or popularity.

- Show the plot summary and synopsis of each movie.

3. *(future implementation)* Checkout and Payment Processing:
    - Show the seating layout and availability of each cinema hall.
    - Allow customers to select the number and type of tickets they want.
    - Allow customers to choose their preferred seats by clicking on the seat map.
    - Calculate the total price of the tickets and display it to the customer.
    - Allow customers to enter their payment details and confirm their booking.
    - Issue a booking confirmation and show the booking details to the customer.
4. User Profiles:
    - Allow customers to edit their personal information.
    - Allow customers to change their passwords and log out from their accounts.
5. Admin Panel:
    - Provide an admin panel for administrators to view and manage the bookings, payments, and customers.
    - Allow administrators to generate and view sales, revenue, and customer feedback reports.

Non-Functional Requirements:

1. Performance:
    - Ensure fast loading times, even with a high number of concurrent users.
2. Security:
    - Be secure and protect the customer's personal and payment information.
    - Reliable and able to handle errors and exceptions gracefully.
3. Usability:
    - User-friendly and easy to navigate.
    - Compatible with different operating systems.
    - Support accessibility features.
4. Scalability:
    - Scalable and able to handle high volumes of traffic and transactions.
5. Data Backup and Recovery:
    - Implement a disaster recovery plan.

Design of the System (Class Diagram)

This section formally introduces the class diagram that represents the structure and functionality of our movie ticket booking system. This diagram provides a comprehensive overview of the system's key components and relationships, facilitating understanding and future development efforts.

Class	Description
User	Represents a registered user of the system and holds information such as first name, last name, credit card number, username, and password.
Movie	Represents a movie available for booking, including its title, rating, genre, showtime, and synopsis.
Booking	Represents a booked ticket, linking a specific user to a particular movie and their chosen seat.
Room	Represents a cinema room with a name, the movie it shows, and a list of available seats.
BookingSystem	Manages all bookings, users, and movies within the system, providing functionalities like user registration, movie booking, and booking history.
Account	Handles all the payment-related functions
Main	Serves as the main entry point for interacting with the system, facilitating user actions, and utilizing the BookingSystem's functions.

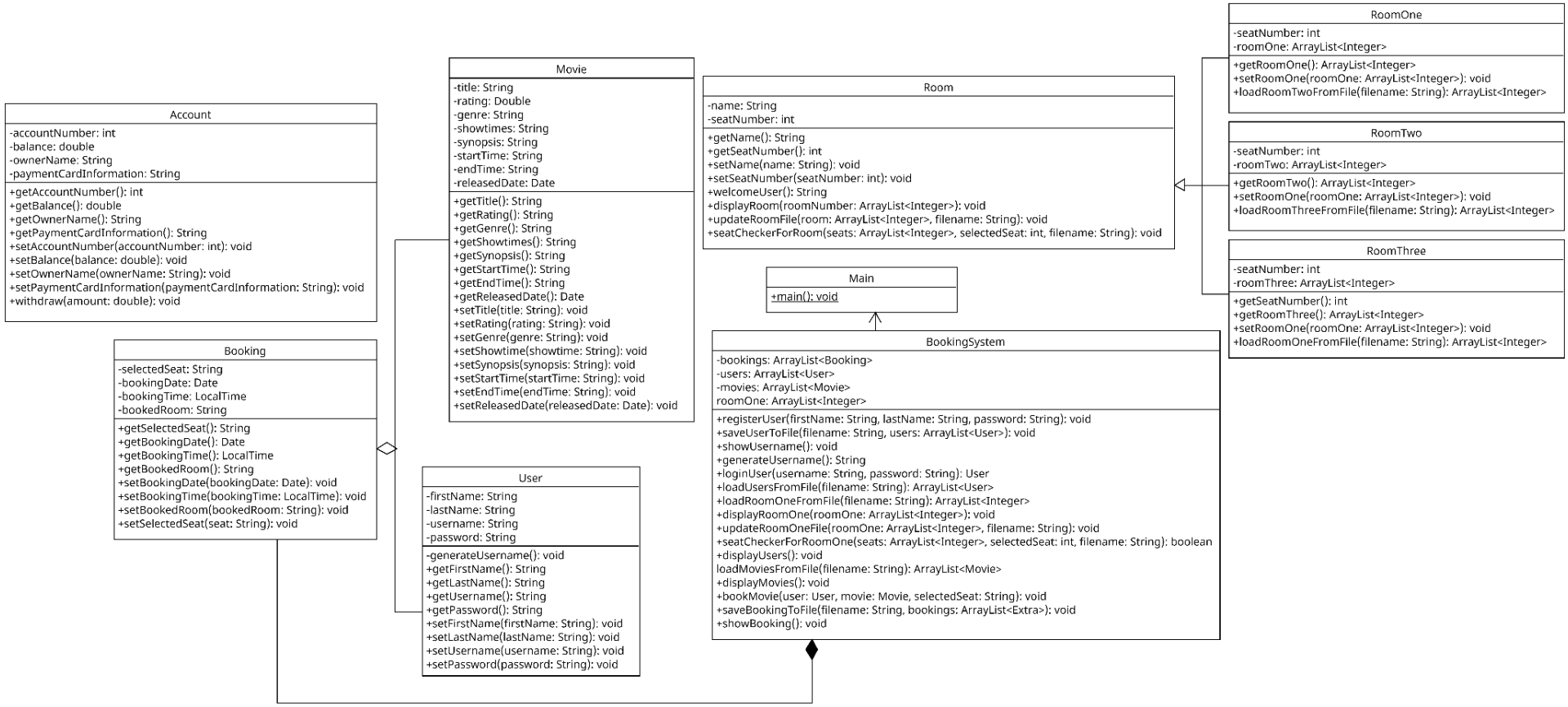
Relationships

- A User can have many Bookings.
- A Booking belongs to one User and one Movie.
- A Movie can be in many Bookings.
- A Room has one Movie and many Seats.

Constraints

- All private variables are accessed through setter and getter methods for security.
- The Main class manipulates the functions and processes within the BookingSystem.
- The system uses file-based databases to store information about Rooms, Movies, Users, and Bookings.

- Users cannot cancel or modify their bookings after booking due to the following reasons:
    - Reduced Administrative Overhead: Managing cancellations and modifications can be time-consuming and resource-intensive for the system.
    - Guaranteed Availability: By eliminating cancellations, users can be confident that their booked seats will be available when they arrive at the cinema.
    - Enhanced Planning and Scheduling: The system can accurately predict attendance and plan resources accordingly.
    - Increased Revenue: Guaranteed bookings minimize the risk of empty seats and lead to greater revenue for the cinema.
    - Fairness and Equality: Disallowing cancellations ensures fair and equal ticket access for all users.


Account
-accountNumber: int
-balance: double
-ownerName: String
-paymentCardInformation: String
+getAccountNumber(): int
+getBalance(): double
+getOwnerName(): String
+getPaymentCardInformation(): String
+setAccountNumber(accountNumber: int): void
+setBalance(balance: double): void
+setOwnerName(ownerName: String): void
+setPaymentCardInformation(paymentCardInformation: String): void
+withdraw(amount: double): void
Movie
-title: String
-rating: Double
-genre: String
-showtimes: String
-synopsis: String
-startTime: String
-endTime: String
-releasedDate: Date
+getTitle(): String
+getRating(): String
+getGenre(): String
+getShowtimes(): String
+getSynopsis(): String
+getStartTime(): String
+getEndTime(): String
+getReleasedDate(): Date
+setTitle(title: String): void
+setRating(rating: String): void
+setGenre(genre: String): void
+setShowtime(showtime: String): void
+setSynopsis(synopsis: String): void
+setStartTime(startTime: String): void
+setEndTime(endTime: String): void
+setReleasedDate(releasedDate: Date): void
Room
-name: String
-seatNumber: int
+getName(): String
+getSeatNumber(): int
+setName(name: String): void
+setSeatNumber(seatNumber: int): void
+welcomeUser(): String
+displayRoom(roomNumber: ArrayList<Integer>): void
+updateRoomFile(room: ArrayList<Integer>, filename: String): void
+seatCheckerForRoom(seats: ArrayList<Integer>, selectedSeat: int, filename: String): void
RoomOne
-seatNumber: int
-roomOne: ArrayList<Integer>
+getRoomOne(): ArrayList<Integer>
+setRoomOne(roomOne: ArrayList<Integer>): void
+loadRoomTwoFromFile(filename: String): ArrayList<Integer>
RoomTwo
-seatNumber: int
-roomTwo: ArrayList<Integer>
+getRoomTwo(): ArrayList<Integer>
+setRoomOne(roomOne: ArrayList<Integer>): void
+loadRoomThreeFromFile(filename: String): ArrayList<Integer>
RoomThree
-seatNumber: int
-roomThree: ArrayList<Integer>
+getSeatNumber(): int
+getRoomThree(): ArrayList<Integer>
+setRoomOne(roomOne: ArrayList<Integer>): void
+loadRoomOneFromFile(filename: String): ArrayList<Integer>
Booking
-selectedSeat: String
-bookingDate: Date
-bookingTime: LocalTime
-bookedRoom: String
+getSelectedSeat(): String
+getBookingDate(): Date
+getBookingTime(): LocalTime
+getBookedRoom(): String
+setBookingDate(bookingDate: Date): void
+setBookingTime(bookingTime: LocalTime): void
+setBookedRoom(bookedRoom: String): void
+setSelectedSeat(seat: String): void
User
-firstName: String
-lastName: String
-username: String
-password: String
-generateUsername(): void
+getFirstName(): String
+getLastName(): String
+getUsername(): String
+getPassword(): String
+setFirstName(firstName: String): void
+setLastName(lastName: String): void
+setUsername(username: String): void
+setPassword(password: String): void
Main
+main(): void
BookingSystem
-bookings: ArrayList<Booking>
-users: ArrayList<User>
-movies: ArrayList<Movie>
roomOne: ArrayList<Integer>
+registerUser(firstName: String, lastName: String, password: String): void
+saveUserToFile(filename: String, users: ArrayList<User>): void
+showUsername(): void
+generateUsername(): String
+loginUser(username: String, password: String): User
+loadUsersFromFile(filename: String): ArrayList<User>
+loadRoomOneFromFile(filename: String): ArrayList<Integer>
+displayRoomOne(roomOne: ArrayList<Integer>): void
+updateRoomOneFile(roomOne: ArrayList<Integer>, filename: String): void
+seatCheckerForRoomOne(seats: ArrayList<Integer>, selectedSeat: int, filename: String): boolean
+displayUsers(): void
loadMoviesFromFile(filename: String): ArrayList<Movie>
+displayMovies(): void
+bookMovie(user: User, movie: Movie, selectedSeat: String): void
+saveBookingToFile(filename: String, bookings: ArrayList<Extra>): void
+showBooking(): void