University of Dhaka

Department of Computer Science & Engineering

CSE – 2112 Object – Oriented Programming Lab
Session: 2022-23

Project Report: Movie Ticket Booking System

Submitted to
Dr. Chowdhury Farhan Ahmed, Professor, CSEDU
Redwan Ahmed Rizvee, Lecturer, CSEDU

Submitted by
Ibna Afra Roza - 16
Mehedi Hasan - 22

Table of Contents

Movie Ticket Booking System

1.1 Concept :

We aimed to design a ‘ Online Movie Ticket Booking System ‘ website using OOP principles and JavaFX for the GUI . The goal of this website is to create an efficient and user friendly platform that connects the ticket buyers with the ticket sellers and accounting sides . Users can login with their personal credentials', have an idea of the available movies, select their preferable one and book ticket by checking available timeslot and pay via online. This site makes it easier for people to do ticket booking hassle free and without physical presence.

1.2 The Motivation :

The objective for developing this software is to meet the need for ticket booking while also improving the accessibility and efficiency of ticket booking services. Traditional methods of booking tickets are often time-consuming and inefficient. By digitalizing the procedure, the site seeks to save time by allowing the buyers to rapidly discover and connect with the theatres without being present physically during their busy schedule. Furthermore, the site can encourage huge community of local people to connect with the digital world in a more easier way.

1.3 Features :

⍰⍰ User Features

Login-Signup: User can login to his account or create one from user_database.txt.
Browse Movies: View the list of movies available for booking.
View Details: Check movie price, available seats, and timeslots.
Book Tickets: Reserve your seat if available.
Check Profile: Visit profile for look back the data and due(Update not possible)
Check Tickets: Look back to the tickets that user have already booked.
Recover Password: Recover the user id with date of birth in case of forget password

⍰⍰ Admin Features

Add Movies: Add new movies with details like title, price, and timeslots.
Edit Details: Modify seat availability, timeslot information, and more.
Remove Movies: Delete movies directly from the database.
File Management: Data is stored in movie_database.txt.

1.4 Tools

UI/UX Design : SceneBuilder
Programming Language : Java
Framework : JavaFX
Database : Files
IDE : IntelliJ IDEA
Other Tools : CSS,FXML

Resources:

- Thumbnails: AI Image Generator
- Icons: Flaticon

2. Design and Implementation

2.1 UX Flow

The UX flow for the "Online Movie Ticket Booking System" is designed to ensure a smooth, user-friendly experience for both customers and administrators. The system architecture is structured to make navigation intuitive, whether users are booking tickets, managing bookings, or accessing administrative functionalities.
Below is the detailed UX flow:

User Onboarding

1. Account Creation and Login

New users can create accounts by providing essential details such as name, email, password, and date of birth.
Existing users can log in using their credentials.
For password recovery, users can reset their passwords via date of birth.

2. Role Identification

Customer: The majority of users will register as customers, primarily using the app to browse and book tickets.

Administrator: A select group of users will have admin privileges to manage movie schedules, tickets, and payments.

Dashboard Interaction

1. Customer Dashboard

After logging in, users are directed to the Customer Dashboard, which contains the following sections:

Movie Listings: Browse available movies and their schedules.
Profile : Check their own profile and booking history or payment due.
Bookings: View active and past ticket bookings.

2. Admin Dashboard

Admin users access an Admin Dashboard with functionalities such as:

Adding and managing movie schedules.
Editing seat layouts for theaters.
Viewing sales reports and analytics.

Movie Selection and Booking Process

1. Movie Details

Customers can click on any movie to view detailed information, including show times, duration , genre.

2. Showtime Selection

After selecting a movie, users choose a show time based on availability.

3. Seat Selection

An interactive seating chart displays available, occupied seats. Users can select seats and proceed to checkout.

4. Payment

Users are redirected to a payment gateway for secure transactions.

5. Booking Confirmation

After a successful payment, users receive a confirmation message showed on the window.

Admin Functionalities

1. Movie Management

Admins can add new movies, update schedules, and remove outdated listings.

2. Seat Layout Customization

Admins can customize the seating arrangement for specific theaters.

3. Reports and Analytics

Sales reports and customer analytics are generated to help admins make data-driven decisions.

Summary of User Journey

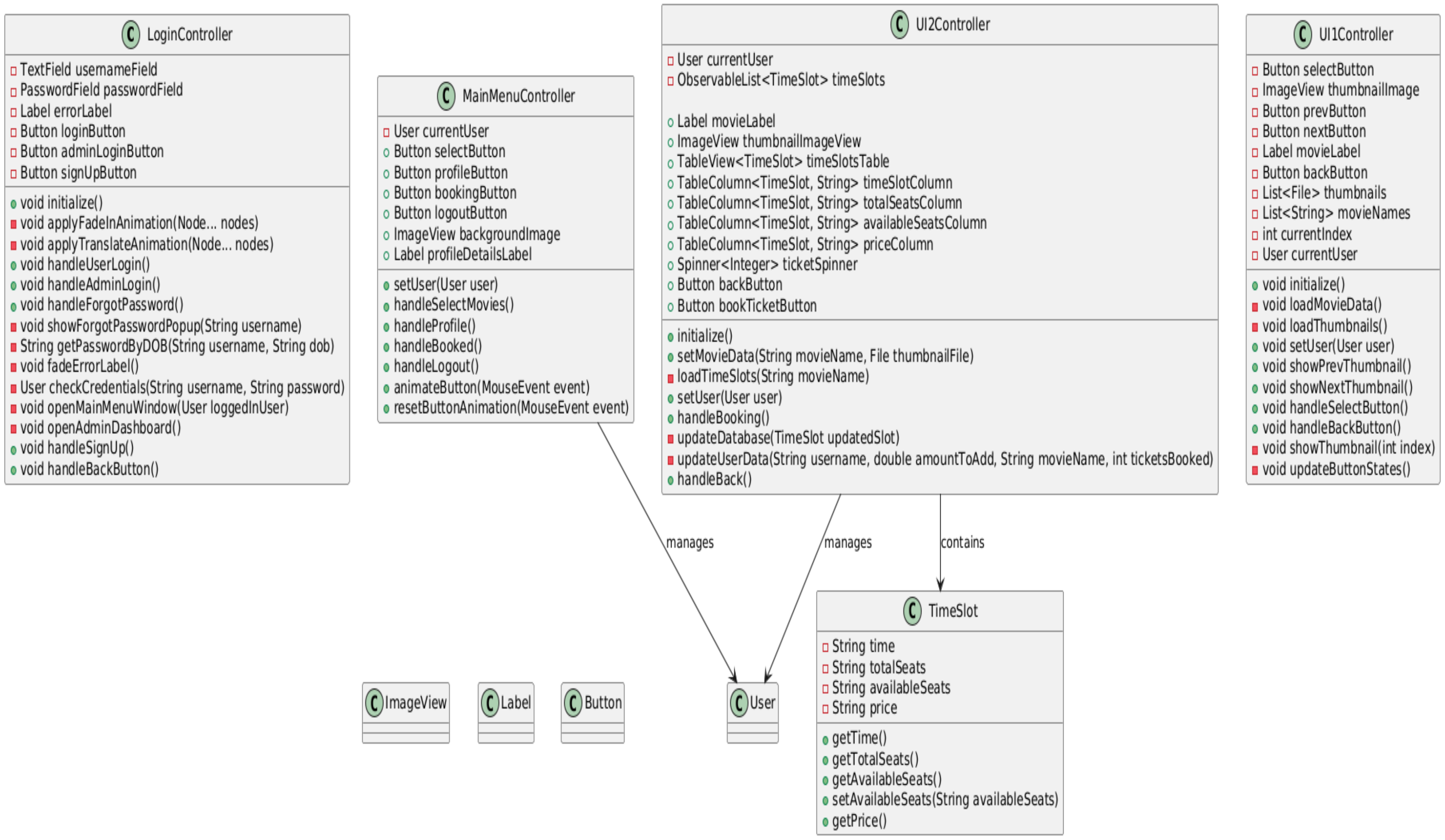
1. Customer

Account creation → Movie browsing → Showtime and seat selection → Payment → Confirmation → Booking history.

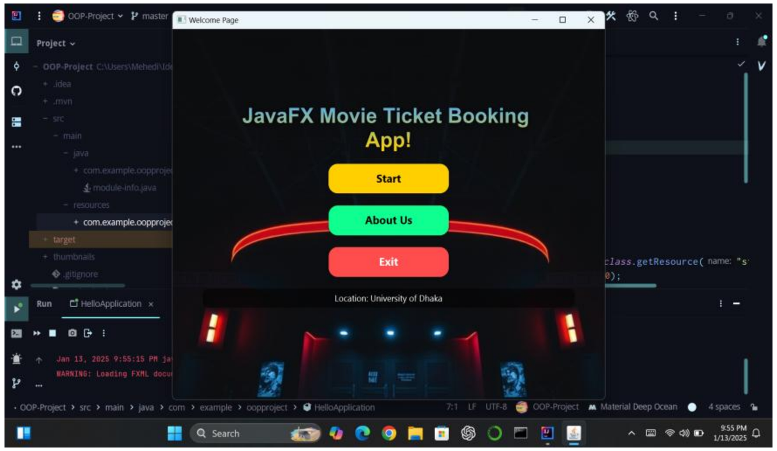
2. Admin

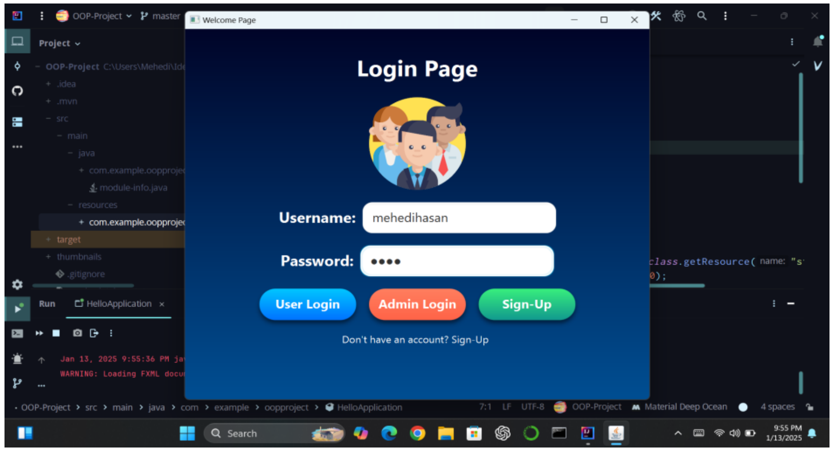
Login → Add or manage movies → Customize seat layouts → View reports → Notify customers of updates or offers.

UML Diagram


LoginController
TextField usernameField
PasswordField passwordField
Label errorLabel
Button loginButton
Button adminLoginButton
Button signUpButton
void initialize()
void applyFadeInAnimation(Node... nodes)
void applyTranslateAnimation(Node... nodes)
void handleUserLogin()
void handleAdminLogin()
void handleForgotPassword()
void showForgotPasswordPopup(String username)
String getPasswordByDOB(String username, String dob)
void fadeErrorLabel()
User checkCredentials(String username, String password)
void openMainMenuWindow(User loggedInUser)
void openAdminDashboard()
void handleSignUp()
void handleBackButton()
MainMenuController
User currentUser
Button selectButton
Button profileButton
Button bookingButton
Button logoutButton
ImageView backgroundImage
Label profileDetailsLabel
setUser(User user)
handleSelectMovies()
handleProfile()
handleBooked()
handleLogout()
animateButton(MouseEvent event)
resetButtonAnimation(MouseEvent event)
UI2Controller
User currentUser
ObservableList<TimeSlot> timeSlots
Label movieLabel
ImageView thumbnailImageView
TableView<TimeSlot> timeSlotsTable
TableColumn<TimeSlot, String> timeSlotColumn
TableColumn<TimeSlot, String> totalSeatsColumn
TableColumn<TimeSlot, String> availableSeatsColumn
TableColumn<TimeSlot, String> priceColumn
Spinner<Integer> ticketSpinner
Button backButton
Button bookTicketButton
initialize()
setMovieData(String movieName, File thumbnailFile)
loadTimeSlots(String movieName)
setUser(User user)
handleBooking()
updateDatabase(TimeSlot updatedSlot)
updateUserData(String username, double amountToAdd, String movieName, int ticketsBooked)
handleBack()
UI1Controller
Button selectButton
ImageView thumbnailImage
Button prevButton
Button nextButton
Label movieLabel
Button backButton
List<File> thumbnails
List<String> movieNames
int currentIndex
User currentUser
void initialize()
void loadMovieData()
void loadThumbnails()
void setUser(User user)
void showPrevThumbnail()
void showNextThumbnail()
void handleSelectButton()
void handleBackButton()
void showThumbnail(int index)
void updateButtonStates()
manages
manages
contains
TimeSlot
String time
String totalSeats
String availableSeats
String price
getTime()
getTotalSeats()
getAvailableSeats()
setAvailableSeats(String availableSeats)
getPrice()
ImageView
Label
Button
User

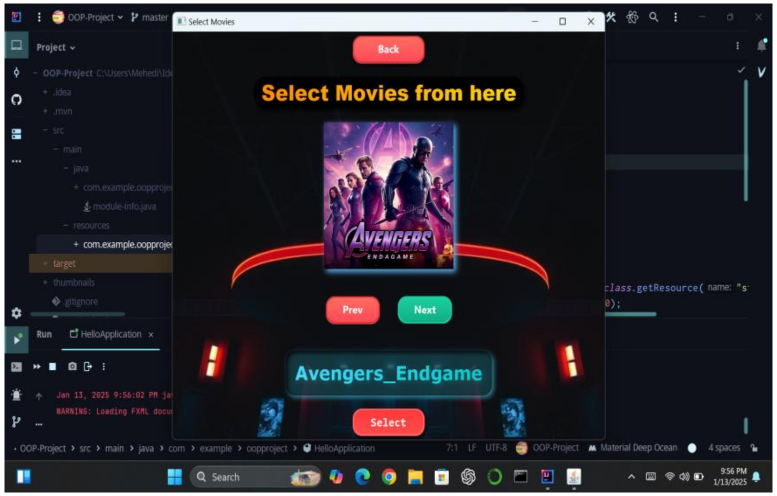
2.2 UI

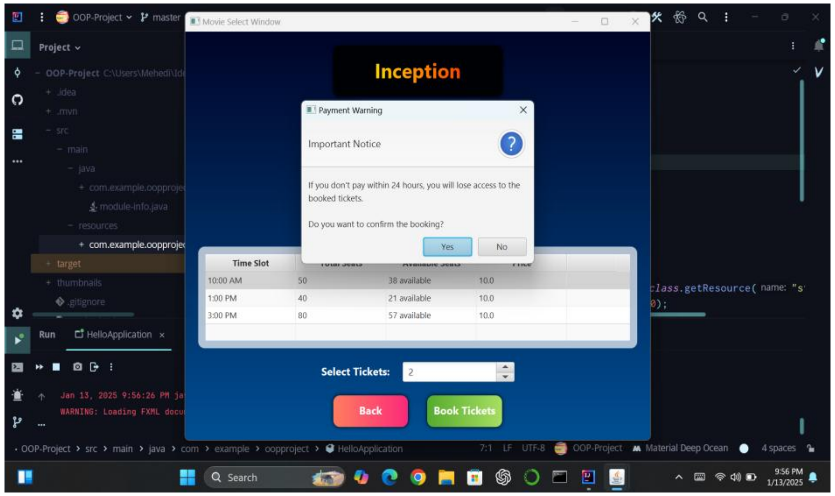

Welcome Page
JavaFX Movie Ticket Booking App!
Start
About Us
Exit
Location: University of Dhaka



Username: mehedihasan

Password: ••••





Important Notice

If you don't pay within 24 hours, you will lose access to the booked tickets.

Do you want to confirm the booking?

Time Slot	Total Seats	Available Seats	Price
10:00 AM	50	38 available	10.0
1:00 PM	40	21 available	10.0
3:00 PM	80	57 available	10.0

Select Tickets: 2

2.3 System Design

The "Online Movie Ticket Booking System" is built using a client-server architecture to ensure scalability, maintainability, and a seamless user experience. The system is divided into three main components: Front-end, Back-end, and Database. Below is a detailed explanation of each:

1. Front-End

Technology : JavaFX
Responsibilities:
The front-end is responsible for managing all user interactions, displaying movie information, and providing a user-friendly interface for booking tickets.

Components:
1. FXML Files: Define the layout and structure of the user interface, such as login pages, movie details pages, and the seat selection screen.
2. Controllers: Manage the behavior and logic for the UI components. For example, handling button clicks for booking tickets or navigating between screens.

2. Back-End

Technology: Java
Responsibilities:
The back-end handles business logic, processes user requests, communicates with the database, and ensures secure payment integration.

Components:

1. Services:

Core business logic resides here, such as managing bookings, seat availability, and user authentication.

2. Controllers:

These route user requests from the front-end to the appropriate services and return responses. For example, when a user selects a movie, the controller fetches the show times from the database.

3. Database

Technology: Files
Responsibilities:
The Files store persistent data, including user profiles, movie details, show times, seat availability, booking information.

Components:

Tables:

Users Table: Stores user information (e.g., user ID, name, email, role).
Movies Table: Contains movie details (e.g., title, genre, description, cast, duration).
Show times Table: Stores available show times for each movie.
Bookings Table: Tracks ticket bookings, including user ID, selected seats, and payment status.
Payments Table: Stores payment transaction details.

Detailed Class Descriptions

User Class

Purpose: Represents a user account and manages login, personal details, and wallet balance.
Attributes:
username, password, firstName, lastName , dateOfBirth, **currentBalance, lastTransactions**
Methods:
getUsername(), **currentBalance()**

Movie Class

Purpose: Represents a movie and manages title, price, timeslots, seat availability, and booking status.
Attributes:
title, price , **timeslot1, timeslot2, timeslot3** , **seats1, seats2, seats3** , **available1, available2, available3**
Methods:
Constructor, getters, and setters for all attributes.

UI1Controller Class

Purpose: Manages movie browsing and navigation.
Attributes:
thumbnails,movieNames,currentIndex,currentUser
Methods:
initialize(),loadMovieData(),loadThumbnails(),setUser(User user), showPrevThumbnail(),showNextThumbnail(),handleSelectButton(), handleBackButton(),showThumbnail(int index),updateButtonStates()

UI2 Controller

Attributes :
movieLabel, thumbnailImageView, timeSlotsTable, timeSlotColumn, totalSeatsColumn, availableSeatsColumn, priceColumn, ticketSpinner, backButton, bookTicketButton, timeSlots, currentUser
Methods:
•initialize(), setMovieData(), loadTimeSlots(), setUser(User user), handleBooking(),
•updateDatabase (TimeSlot updatedSlot), updateUserData(String username,
•double amountToAdd, String movieName, int ticketsBooked), handleBack()
Inner Class:
•TimeSlot (Attributes: time, totalSeats, availableSeats, price;
•Methods: getTime(), getTotalSeats(), getAvailableSeats(), setAvailableSeats(),
•getPrice())

LoginController Class

Attributes:
usernameField, passwordField, errorLabel, loginButton, adminLoginButton, signUpButton
Methods:
initialize(), applyFadeInAnimation (Node... nodes), applyTranslateAnimation (Node... nodes), handleUserLogin(), handleAdminLogin(), handleForgotPassword(), showForgotPasswordPopup(String username), getPasswordByDOB (String username, String dob), fadeErrorLabel(), checkCredentials (String username, String password), openMainMenuWindow (User loggedInUser), openAdminDashboard(), handleSignUp()

SignupController Class
Attributes:
BackButton, usernameField, firstNameField, lastNameField, dobField, passwordField, errorLabel
Methods:
handleBack(), handleSubmit()

AdminController Class
Attributes:
movieTable, titleColumn, priceColumn, timeslot1Column, seats1Column, available1Column, timeslot2Column, seats2Column, available2Column, timeslot3Column, seats3Column, available3Column, movieList, FILE_PATH
Methods:
initialize(), setupEditableColumns(), onAddMovie(), onDeleteMovie(), onSaveChanges(), loadMoviesFromFile(), saveMoviesToFile (List<Movie> movies), showAlert (String title, String message), handleLogout (javafx.event.ActionEvent actionEvent)

2.4 Discussion

The app ensures usability, reliability, and scalability with features like interactive seat selection and secure payment integration. RESTful APIs handle concurrent requests and data security, while JavaFX provides a responsive, user-friendly interface for smooth navigation.

2.5 Use of OOP

The application applies OOP principles:
Encapsulation: Data like users, movies, and bookings are securely managed in classes.
Inheritance: The User class is extended by Customer and Admin for role-specific behavior.
Polymorphism: Methods like displayDashboard are customized in subclasses for different roles.
Abstraction: Interfaces ensure flexible and modular system design.

Conclusion

3.1 Challenges and Solutions

Payment Integration: Secure and seamless payment required extensive testing.
Seat Selection: Real-time updates were achieved with optimized database queries.
Data Security: Encryption and secure authentication ensured user privacy.

3.2 Future Enhancements

Database integration (e.g., MySQL) for scalable data management.
Payment gateway integration for online ticket booking.
Graphical seat selection interface.

3.3 Reference

All the movie thumbnails were generated from the website: AI Image Generator
Icons that has been used, collected from a free website: Flaticon

Reference Book: JavaFX 9 by Example, Carl Dea, Gerrit Grunwald, Jose Pereda, Sean Phillips, Mark Heckler

3.4 Contribution

Mehedi Hasan – 22

Managed back-end development, database integration, login-signup mechanism and admin access for editing file. Worked on ensuring secure transactions, efficient handling of user requests, and scalability of the system.

Ibna Afra Roza – 16

Designed and implemented UI1 (user dashboard) and UI2 (movie selection and seat booking). Focused on creating an intuitive, responsive interface and seamless navigation between features. Ensured real-time updates for seat availability and smooth user interactions.