EAST WEST UNIVERSITY

CSE110: Object Oriented Programming
Section: 11
Spring 24 Semester

Project Report
CineSphere: A Movie Ticket Booking System

Course Code	: CSE110
Course Title	: Object Oriented Programming
Section	: 11
Group No	: 01
Group Name	: ByteBuilders

Submitted by:

Student ID	Student Name	Contribution Percentage
2023-3-60-222	Sudeepta Mandal	20%
2023-3-60-223	Md. Abir Rahman	50%
2023-3-60-214	Sadman Haque	15%
2023-3-60-221	Md. Asif Khan	15%

Submitted To:

Ahmed Abdal Shafi Rasel,
Lecturer, Department of CSE
East West University

Submission Date: January 19, 2025

Table of Contents

Introduction

The "*CineSphere: A Movie Ticket Booking System"* is a user-friendly and efficient Java-based application developed to streamline the movie ticket booking process. In today's fast-paced world, traditional methods of purchasing movie tickets are becoming increasingly obsolete. Customers seek convenience, speed, and flexibility when booking their movie tickets. This project addresses these demands by providing a digital platform for selecting movies, viewing schedules, and reserving tickets, all from a single interface.

The system incorporates a wide range of functionalities, including viewing movie details, selecting seats, and processing bookings. It also features administrative controls for managing movies, schedules, and ticket availability. Built using JavaFX for the graphical user interface and a robust backend database for data management, CineSphere ensures seamless operation and a smooth user experience. The system aims to enhance user satisfaction by reducing the hassle of in-person bookings and minimizing errors associated with manual systems.

CineSphere is designed to cater to both users and administrators, making it a versatile solution for theaters and moviegoers alike. The project's development focused on simplicity, scalability, and adaptability, ensuring it can meet the needs of diverse audiences and environments.

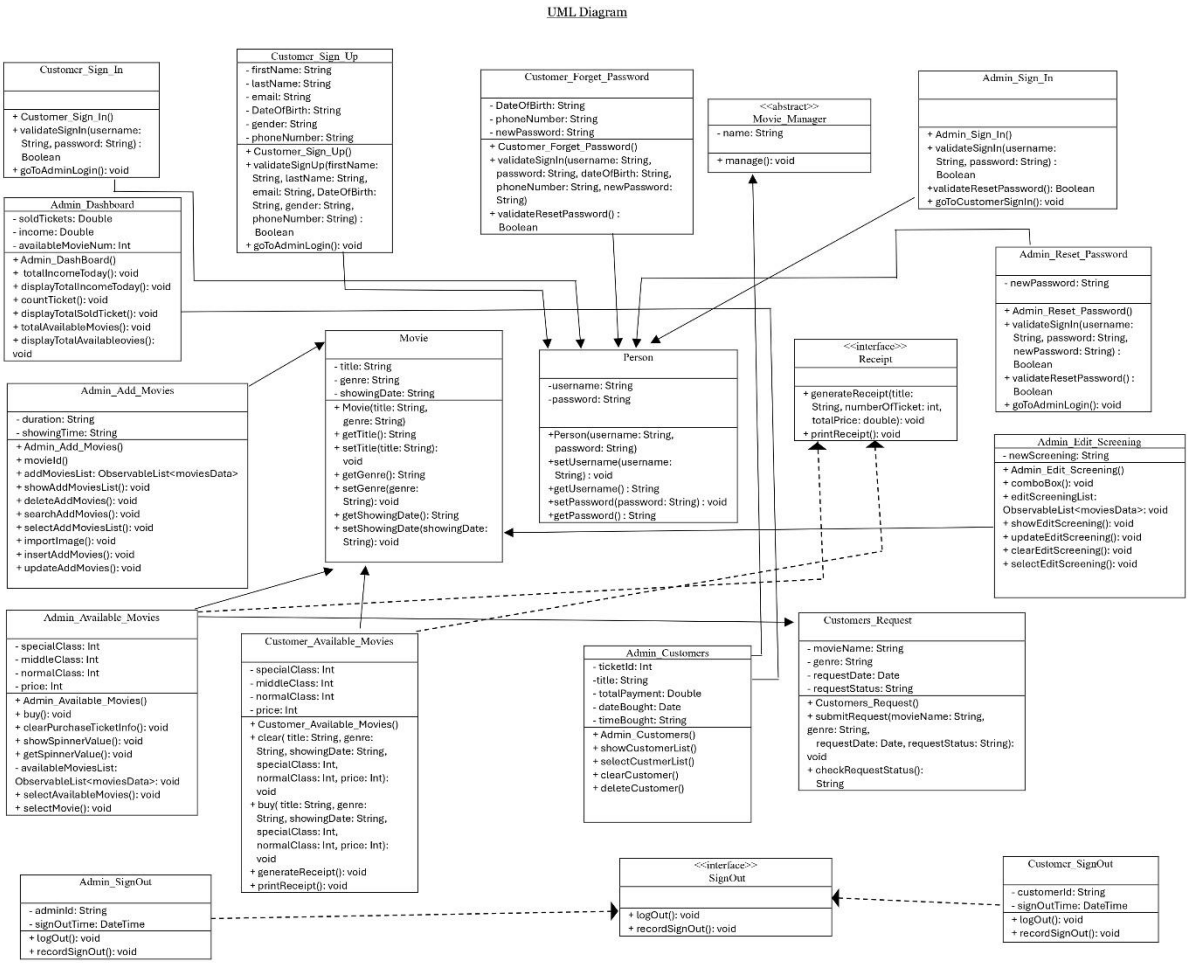
Objective

The primary objective of "*CineSphere: A Movie Ticket Booking System"* is to develop a reliable, efficient, and user-friendly platform for managing and automating the process of movie ticket booking. This project aims to provide moviegoers with a seamless experience for browsing movies, selecting showtimes, reserving seats, and making bookings, while also offering theater administrators tools to manage movie schedules, ticket availability, and customer data effectively.

The system is designed to reduce manual errors, save time, and enhance the overall efficiency of movie ticket management, ultimately catering to the growing demand for digital solutions in the entertainment industry.

Methodology

UML Diagram



Implementation

*Customer_Sign_In
-Validate the customer's login using a username and password.
-Provide a way to navigate to the admin login interface.

*Customer_Sign_Up
-Register a new customer with their personal details like name, email, phone number, gender, and date of birth.
-Validate the customer signup data.

*Customer_Forget_Password
-Validate a customer's request to reset their password.
-Allow customers to set a new password using security steps.

*Admin_Sign_In
-Validate the admin's login using a fixed password.
-Provide a way to navigate to the customer sign-in interface.

*Admin_Reset_Password
-Validate the admin's request to reset their password.
-Allow the admin to set a new password using security steps.

*Admin_Dashboard
-Display the total number of tickets sold, income generated, and available movies.
-Allow viewing and managing total tickets and available movies.

*Admin_Add_Movies
-Insert new movie details such as title, genre, duration, showing time, and screening date.
-Update or delete movie details.
-Provide a list of all movies.

*Admin_Available_Movies
-Display the movies available for booking.
-Allow admins to add or edit movies.
-Generate and print receipts for tickets sold.

***Customer_Available_Movies**
-Display available movies for the customer to view and book.
-Allow customers to select movies and purchase tickets.
-Generate and print receipts for ticket purchases.

***Admin_Edit_Screening**
-Edit the screening details for a movie, such as time or date.
-Update or clear any changes made to the screening.

***Admin_Customers**
-Manage customer details.
-View specific details about a customer.

***Customers_Request**
-Submit a movie request (e.g., for unavailable movies or changes).
-Allow customers to check the status of their submitted requests.

***Admin_Sign_Out and Customer_Sign_Out**
-Log out the respective user.
-Record the logout time for auditing purposes.

***Person**
-Provide basic attributes and functionality for both admin and customer users, such as username, password, and methods to manage these attributes.

***Abstract Class: Movie_Manager**
-Serve as a base class for managing movie-related tasks.
-Define an abstract manage() method to be implemented by subclasses like Admin_Customers.

***Interface: Receipt**
-Provide methods to generate a receipt and print it, ensuring consistent receipt generation across admin and customer operations.

***Interface: SignOut**
-Define methods to handle logout functionality for admins and customers.

***Movie**
-Store movie details such as title, genre, and screening information.
-Provide methods to access and modify movie details.

Reflection on Individual and Team work

Sudeepta Mandal: 20% (Customer & Admin login and signup page, Splash Screen page)

Md. Abir Rahman: 50% (Customer & Admin login and signup page, Splash Screen page, Admin Dashboard page)

Sadman Haque: 15% (Customer & Admin login and signup page, Splash Screen page)

Md, Asif Khan: 15% (Customer & Admin login and signup page, Splash Screen page)

Limitations & FutureWorks

Limitations: There are so many bugs & limitations in our movie ticket booking system desktop application. We didn't completed the Customer sign in checking page, Customer sign up page, Customer forget password page, Admin reset password page. Since our application is not only for Admin but also for Customers, we didn't touched the Customer part.

We didn't implemented the Delete button and Search bar in the Admin add movies class. We also didn't implemented the Receipt button in the Admin available movies class. Again, we didn't implemented the Search bar both in the Admin manage screening class and Admin customers class.

When we buy or add movies , our Admin dashboard can’t update the total sold ticket, total earned today and total available movies automatically untill we re-run the project. Sometimes many bugs ar found when we run our project, like-UI color mismatch. Sometimes the id numbers didn’t change, that’s why all the movie or customer id remains same.

We couldn’t used MYSQL because we didn’t complted Database Management System course yet.

Future Works:

Complete Customer Functionality:

- Implement the Customer Sign In page with proper validation.
- Develop the Customer Sign Up page to allow new users to register.
- Create the Customer Forget Password page for password recovery.

Enhance Admin Functionality:

- Add the Delete Button and Search Bar in the Admin Add Movies class.
- Implement the Receipt Button in the Admin Available Movies class to generate and display receipts for transactions.
- Integrate the Search Bar in the Admin Manage Screening class and Admin Customers class to filter and find specific records efficiently.

Dashboard Auto-Update:

- Ensure the Admin Dashboard updates the following in real-time:
    - Total Sold Tickets
    - Total Earned Today
    - Total Available Movies

Bug Fixes and Optimization:

- Resolve UI Color Mismatches to maintain a consistent design and improve the user experience.
- Fix the ID Number Issue, ensuring that each movie or customer has a unique ID to prevent data conflicts.
- Address other runtime bugs and ensure smoother operation during various user interactions.

Improve User Experience:

- Implement validation for form inputs to prevent errors during user interactions.
- Provide meaningful error messages for any exceptions or invalid actions.
- Enhance the overall design and responsiveness of the UI.

Receipt and Reporting Features:

- Develop a detailed receipt generation system for customers.
- Include reporting features for Admin, such as:
    - Monthly or daily revenue reports.
    - Ticket sales trends.

Testing and Debugging:

- Conduct thorough unit testing, integration testing, and system testing to ensure all features work as expected.
- Perform stress testing to handle edge cases like concurrent bookings or a large number of records.

Database Connection:

- In future, we will connect MYSQL workbech with our project.

Additional Features:

- Introduce a movie recommendation system for customers based on their booking history.
- Add a notifications feature for customers regarding upcoming screenings or offers.
- Implement a movie rating and review system for customers to share feedback.

By addressing these future works, the application will become more robust, user-friendly, and feature-complete, catering to both Admin and Customer needs effectively.

Reference

We used OpenAI(ChatGPT) for some hard method implementation in our project codes. Since we didn’t completed our Database Management System Course, we used google and ChatGPT to connect our project with database.