

Ali Hammad 900231044

Karim Elnazir 900231041

Zeyad Maher 900231975

Karim Elhenawy 900232155

Mohamed El-Sayed 900231065

Cinema Booking System - README

Project Overview

Cinema Booking System is an online platform designed to streamline cinema ticket booking. The system provides a graphical user interface (GUI) for both users and admins. Our online platform supports key functionalities, such as seat selection, event management, and wallet system. It is designed to ensure flexibility and convenience. Advanced data structures including splay trees, custom vectors, and 3D arrays are used to ensure efficient performance and real-time updates.

Features

User Functionalities:

- Browse and view upcoming movies and event details.
- Dynamic seat selection with real-time updates.
- Wallet system for managing balances and secure transactions.
- Two-factor authentication for enhanced security.
- Booking history and cancellation with refund.

Admin Features:

- Add or remove movies/events.
- Manage event details, time slots, and pricing.
- Monitor seat availability and bookings.

Key Design Elements:

- Responsive and visually appealing GUI.
- Differentiation between VIP and regular seat types.
- Real-time ticket generation and detailed booking confirmations.

Technical Highlights:

- **Programming Language: C++**
- Developed using **Qt** for an interactive GUI.
- **Data Structures:** Splay Trees, 3D Matrix, Custom Vector (vectorC)
- **Splay Trees:** Efficiently manages customer profiles and data handling.
- **Custom vectorC Class:** Manages dynamic event listings and is used in most parts of the program.
- **3D Matrix:** Handles tickets by tracking movies, time slots, and seat allocations.
- Includes components like login/signup pages, event browsing, seat selection, and admin functionalities.
- **Database:** Stores user profiles, tickets, and event data.

Usage

1. **Run the Application:** Run the application from Qt Creator.
2. **User Actions:**

Login/Register: Access your account or create a new one.

Browse Events: Explore available movies and select a showtime.

Book Seats: Choose seats and confirm the booking with wallet balance.

Manage Wallet: Add funds or check balance from the account page.

3. **Admin Actions:** Use the admin panel to add new movies, update schedules, or manage bookings.

Code Structure

Below is a detailed explanation of the purpose and functionality of each file in the project:

- **Accountpage:** it contains the constructor with the defined parameters passing Customer as a pointer and defined functions.
- **Admin:** declares the push buttons and box layouts of the UI.
- **Confirmbook:** contains the logic for booking confirmations, such as generating ticket details and payment handling. It declares a double T, a vector of int for seat numbers, double RBalance, and passes in the constructor: customer, seats, and events.
- **Creditcardwidget:** passes customer in the constructor and declares the pushbuttons and line edits for the UI.

- **Customer:** it creates a class wallet that declares the attributes and functions, and the class customer declares the attributes and functions where one of them includes a vector of reserved tickets.
- **Customerlogin:** handles the login process, validating user credentials using the splay tree structure and has the UI functionalities for logging in or registering.
- **Events.h:** this initializes attributes such as event Id, title, duration, date, price, and image. It also has functions such as getters and setters, the constructor and the move operations.
- **Eventwindow1:** creates the graphical interface to show the list of events available and their details, login or register push buttons that open their windows.
- **Eventwindow2:** creates the graphical interface to show the list of events available with available time shows, my account and my tickets push buttons that open their windows.
- **Halls:** defines the layout and configuration of cinema halls.
- **Registerwindow:** shows the registration process that initialize data like email, ID, name, and password, where it creates a new customer object that is inserted inside the splay tree
- **Reserved:** class is made for the UI window to show the Tickets reserved from the vector that was implemented.
- **Seats:** creates the UI for selecting seats and having a cinema screen to show the floor planning seating to allow users to choose their seats in an interactive way, and it calculates the total price and has a confirm book push button that will open the tickets details. Where it automatically updates the tickets status in the 3D matrix implemented.
- **Splaytree:** This class implements the splay tree where customers are dynamically rearranged. It includes the main functionalities for a self adjusting binary tree, these functions ensure that the tree will always be balanced.
- **System:** This class manages the tickets for events using the 3D matrix, where each ticket is dynamically allocated Where the i rows are movie names, j columns are time slots and k cells are seat numbers.
- **Tickets:** The tickets class contains the ticket attributes and functionalities of tickets such as generate price.
- **VectorC:** This class creates the vector data structure with its functions from scratch.

UI Files

- **accountpage.ui**: the UI design for the account page, displaying wallet information.
- **confirmbook.ui**: design for the booking confirmation page.
- **creditcardwidget.ui**: UI design for the credit card payment interface.
- **customerlogin.ui**: UI for customer login, including fields for username and password.
- **eventwindow1.ui / eventwindow2.ui**: graphical interface for event browsing and detailed event information.
- **loginadmin.ui**: UI for admin login.
- **mainwindow.ui**: main application interface, serving as a hub for navigation.
- **registerwindow.ui**: user registration form design.
- **reserved.ui**: displays reserved seat information.
- **seats.ui**: visual layout for seat selection, including VIP and regular sections.

Contributors

- Ali Hammad, Karim Elnazir , Zeyad Maher, Karim Elhenawy, Mohamed El-Sayed