Movie Ticket Booking System

By :

Maitri Kasodariya (202001197)
Miglani Neha (202001097)

Description:

Using this system, customers can book their seats without any obligation, at any time. This system is an attempt to provide the advantage of systematic ticket management for customers to create a user-friendly environment to facilitate customer services, this system would play an immense role for better customer satisfaction.

The project is simple to understand, and the source code has been presented in an understandable manner. We have used CodeBlocks as an IDE for better code compilation.

We have used a base class named Movie, which inherits two classes String and Ticket. Our program uses multiple if-else and switch statements including for loop. Also, data structures like LinkedLists (to book tickets for different movies with their slots), Arrays (to display and update seat matrix), and Binary Search Trees (so that users can find the nearest theatre by taking their pin codes as input) have been used.

Problem Statement:

A client who is the owner of multiple theatres located in different cities of India wants a system such that any of his customer can book a ticket from his place and the client can manage all the data on his computer itself so that transparency is maintained and money is saved from using paper

ticket formats.Also that their customers get the leverage of booking tickets from home and not travelling long distance to their theatres consuming fuel and time. Also they can increase their customer base by using such a system.

Approach:

- We first planned and listed all the possible features that we can include in our programme.
- Then we drew a rough sketch of the display pattern of the main screen and the paths for our program.
- We decided to design the programme such that we code the different parts and we can compile it and see if there are any errors in the recently designed part of the code and move further.
- This was done to avoid ambiguity at the final run of the code.
- Then we started coding by initialising a base class containing general functions.
- We initialised a few more classes as needed and made them the subclasses of the base class for easy access of the base class content in those classes if needed.
- Further we coded the functions and connected them and completed the code.
- We went through a lot of compile-time and run-time errors at different places and tried to solve them as per our logic.
- Also after completing the code we felt that still few things were missing and so we started working on the code again.
- While working, we realized that our code would have been better if we made some changes in some of the functions, and thus we changed some of the functions again.
- It was a great experience going through this project which improved our coding schemes and gave us an idea of the complications in such a big code/project.

- **<u>Features Included in the program:</u>**

  - ○ Special discounts for users having membership cards.
  - ○ Book tickets for different areas in 6 different cities and 4 different states.
  - ○ Selection for the Premium or Gold seats.
  - ○ Different payment methods (online and offline as well).
  - ○ 5 screens for different movies.
  - ○ 8 different slot timings to watch the movie as per user's ease and time.
  - ○ A user can also cancel the ticket.
  - ○ A person can contact the theatre if he wants to advertise.
  - ○ Know the ticket price from the main window.
  - ○ Know if the appropriate timings are available to watch the movie.
  - ○ Data storage of the ticket transactions with details in xls file for the owner of the theatres.

<u>Things learnt:</u>

- How to include real time in the code.
- Learnt File Handling for data storage.
- A more clarity on pointers.
- Using Inheritance in classes.
- Not to write the whole code at once and keep it's run at last rather keep it testing after completing every part of the code.
- If your code is completed but still you want to make some major changes in it then you should keep a copy of your code saved so that your whole code doesn't go in vain because of some error.
- Also, we need to rely on some sources to perform better while writing the code.

Limitations:

- We don't have a system which books the tickets for the upcoming days (tomorrow, day after tomorrow).
- When the slot is full for a movie, the system does not show automatically that it is housefull.
- Ticket Cancellation is not available on the program.
- The owner of the theatres has no option to decide the slots that he wants to keep for a specific movie.
- Also the owner cannot insert the movie by an option instead he will have to change the code himself.