T03 Project Part 2 Report

ECE 103

Faisal Alsemany (alselem@pdx.edu)
Hoang Du (hkd3@pdx.edu)
Dmitrii P Fotin (dfotin@pdx.edu)
Rekik Mengstu (rekik@pdx.edu)

Introduction

The goal of the project is to build and present a well-programmed and convenient movie ticket application. It would function via compiler console and would walk a user through each step of booking a movie ticket, pre-ordering snacks and drinks and completing the purchase as well as printing out the order details to a TXT file.
The team split the initial project description into standalone sections to distribute the workload among the teammates and ensure modular program development. Such sections were coded into individual functions.
The program has a separate User Interface and password-protected Admin Interface. The User Interface walks a user through the ticket booking process, adding snacks/drinks to the order, adding promo codes, adjusting order details, starting from scratch or cancelling the order.
The Admin Interface allows to review and change all ticket bookings and scheduled viewings as well as add new movies to the list, add new viewings to the schedule and update the blacklist.

Software Design

Description of the problem

The project is to design a movie ticket booking program that allows users to reserve seats for selected viewings and has a separate admin interface to allow adding or adjusting bookings and viewings.

Design approach and assumptions

The program contains multiple arrays that store data from inputs while the program is running. After each confirmed update to the arrays, the program also updates the relevant external TXT files. Upon startup, the program reads the data from the external files and stores in in the arrays to make sure the user is operating with the latest data in mind.
The program was divided into logically separate sections, each of which is represented by a standalone function to make the program easier to read and update.
The program stores all global variables and functions in respectively named header files separate from main.c to make the program easier to read and update.
The program starts with User/Admin selection. User Interface walks the user through the steps to complete the booking: select movie, date, time and seat - as well as adding snacks/drinks to the order. The user is then presented with choices to complete the booking, update the booking details, apply a promo code, start the booking over or cancel the booking altogether. Upon confirmation, the program updates all TXT files on the drive to contain the latest data.
The Admin Interface allows to see all bookings and viewings as well as change/remove them and add new ones. It also allows editing the blacklist that prevents certain users from completing bookings.

Logic flow

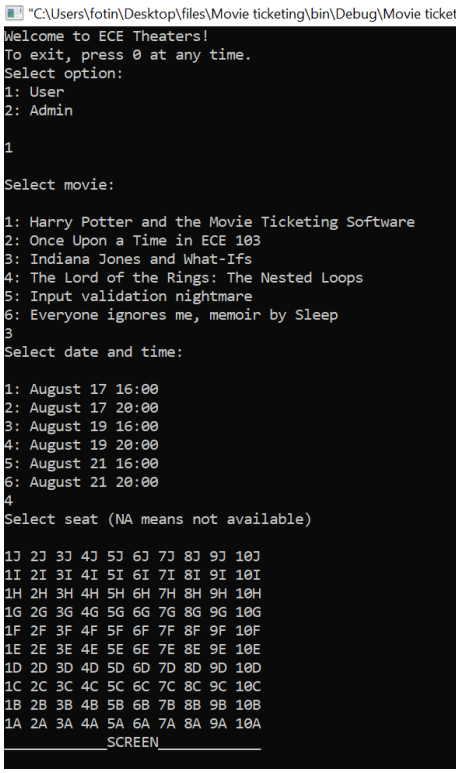
Please reference the **Logic_Flowchart.png** file submitted along with the report.

Function list

1. **void** uploadData - function to upload all data from TXT files at program startup
   Dmitrii Fotin
2. **void** selectMovie - function to guide the user through the booking process while referencing all other existing bookings
   Hoang Du
3. **void** selectFood - function to guide the user through food/drink selection process
   Rekik Mengstu
4. **void** pay - function to confirm the booking, update the system records and data exports
   Hoang Du and Dmitrii Fotin
5. **void** printTicket - function to output the booking details to a TXT ticket file (assumed to be shared with the user)
   Dmitrii Fotin
6. **void** applyPromo - function to apply discount based on user-provided promo code to the user total
   Rekik Mengstu and Dmitrii Fotin
7. **void** changeOrder - function for user to update snack/drink details before completing the booking
   Dmitrii Fotin
8. **void** startOver - function to discard all booking details and restart the booking process (**void** selectMovie)
   Dmitri Fotin
9. **void** endSession - function to terminate the session without updating any data exports
   Dmitrii Fotin
10. **void** printSeats - function to print available seats for user to choose from
    Hoang Du and Dmitrii Fotin
11. **void** reservationConfirmation - function to check user-selected seat against the existing bookings for validity
    Hoang Du and Dmitrii Fotin
12. **void** adminLogin - function that authenticates the user into the admin interface
    Dmitri Fotin
13. **void** viewBookings - function that prints a report of all existing bookings
    Dmitri Fotin
14. **void** addEvent - function that allows the admin to add a new viewing to the schedule
    Dmitri Fotin
15. **void** editEvent - functions that allows the admin to change the existing viewing details
    Dmitri Fotin

16. **void** editBooking - function that allows the admin to update existing order details (viewing and snacks/drinks)
    Dmitri Fotin
17. **void** editBlacklist - function to add new entries to or remove existing entries from the blacklist
    Dmitri Fotin

Photographs



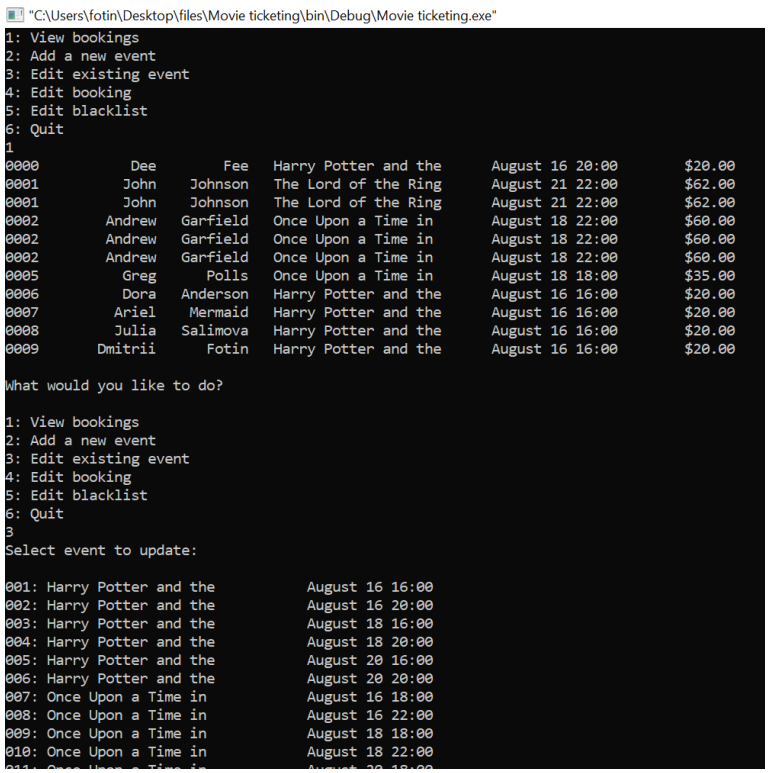


Figure 1: User interface *Figure 2: Admin interface*

Extra credit list:

- Blacklist feature preventing banned customers from placing orders;
- Promo code feature allowing to apply discount to order total based on user-entered promo code;

Problems and solutions

- The code was very repetitive with many steps happening over and over, the team organized as much code into functions as possible to shorten the code by referencing individual sections of the program as needed;

- The program was very long to read, so the team broke functions and global variables out into separate header files;
- The string inputs were causing many errors, fflush(stdin) and fgets functions allowed to establish reliable inputs;

Discussion

The project covered all class material as well as required a lot of external research. The code prepared by the team uses functions and header files to organize code, passes arguments by value and pointer to ensure correct program flow, stores and retrieves data from external files in order to operate outside of hardcoded values and RAM and processes a very diverse range of string, double and int inputs and outputs with quite elaborate input and output formatting.

Signature

In the project each person do well with their contribution as:

Hoang Du: Ticket booking functionality, changing booking details functionality, file IO, project report.
Rekik Mengstu: Food ordering functionality, promo functionality, file IO, project report.
Faisal Alsemany: Version testing.
Dmitrii Fotin: File IO, header file and function organization, Admin UI functionality, project report.

All of the students listed at the top of this report have read it and agree with its content.