**C++ Project**

Name: Nick McCaig

Student ID:  N0787115

**SOFT10101 computer science programming coursework 19/20**

**Title: CINEMA MANAGEMENT SYSTEM**

The cinema management system will mimic the functions required to run a cinema, it is designed to be run in a command line, the program does not use any libraries BUT does contain copied code as declared in the declaration file, it is also commented in the code itself.

## Aims and Objectives

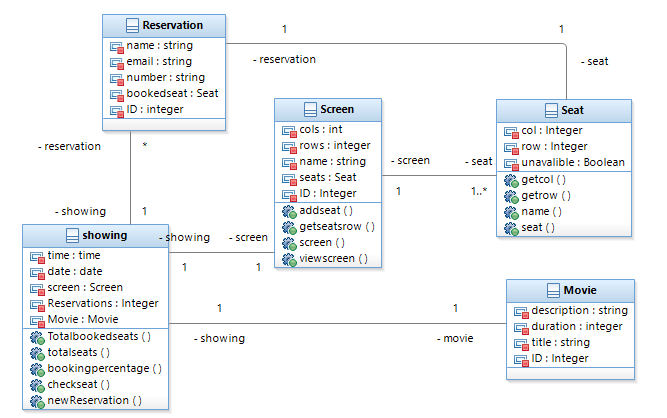
My code will be compared to the following objectives and aims in order to determine if I have been successful.

* To setup and configure a cinema’s theatres.
* To make staff reservations
* To view occupied seats
* To create seats including special seats such as those unavailable
* To create reports for a day showing future showings and booking numbers
* Film and weekly reports
* To add new movies
* To add new showings
* Have checks in place to prevent overbookings and conflicting run times
* Use OOP
* To save data

# DEsign

The program uses a OOP design this helps reduces the amount of code repeated unnecessarily. The system uses pointers to link the class’s while avoiding replicating data, the class diagram below demonstrates how the data is created and stored.

Below are diagrams used to create the program, note most functions are commented explaining their purpose.



Some notable aspects of the design include the use of ID’s which are only used when saving the file in the CSV file when the data is serialised to allow for saving as the object-oriented design makes use of pointers to link data.

## File structure

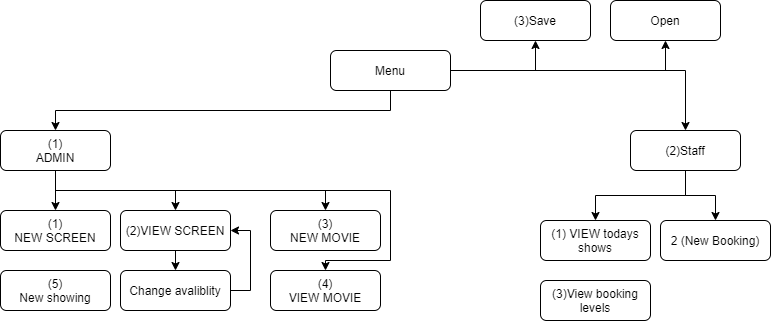
The file structure makes use of 4 files, each saving a different type of object except for the screen class which also contains the seats. Below is an example of the file structure used. A new line is used to represent a new object.

Note that this file is not designed to be edited easily by the end user and they are strongly encouraged to use the program as the opening file function will not check for any conflicts when a user creates new show as it loads as this is done in the initial creation. (in blue is example data which could be stored in these files)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Movie File | |  |  |  |  |  |  |
| ID | Name | Duration | Description | |  |  |  |
| 0 | Movie A | 33 | A movie | |  |  |  |
| Screen File | |  |  |  |  |  |  |
| ID | Name | Rows | Cols | Seat- row | Seat- col | Seat- Unavailable | |
| 0 | Screen 1 -IMAX | 10 | 10 | 0 | 0 | 0 | continues |
| Showing File | |  |  |  |  |  |  |
| Showing | ScreenID | MovieID | Year | Day | Hour(24) | Min |  |
| 0 | 0 | 0 | 2020 | 1 | 12 | 12 |  |
| Booking File | |  |  |  |  |  |  |
| ID | Name | Number | Email | Seat- row | Seat- col |  |  |
| 0 | Jam | 4578923 | aohiw@dawd.com | 0 | 0 |  |  |
| 1 | Nick | 793 | ic@dwajd.com | 0 | 4 |  |  |

# UI design

### Menu system



I used a menu flow diagram to keep track of menu system I had created, this made it easier to ensure the menu system worked as expected. All this code Runs in a Loop, to exit the program or a menu option at any time when selecting a new option, the user would type ‘9’.

My using a OOP design there are some key features of the code which make easy to add and remove features.

# testing

In order to carry out the tests the code must first have some sample data, some tests do not require this testing data but where it is required it is noted. The complete testing data is below. Each test should be completed separately unless it is noted it requires a previous test to be completed successfully.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Bookings |  |  |  |  |  |  |
| 1 | Nick | 89213 | a@mail.com | 0 | 8 |  |
| 1 | Nick | 89213 | a@mail.com | 1 | 8 |  |
| 1 | Nick | 89213 | a@mail.com | 2 | 8 |  |
| 1 | Jam | 4328932 | a@email.com | 7 | 4 |  |
| Movies |  |  |  |  |  |  |
| 0 | A movie | 38 | this is a movie | |  |  |
| 1 | A new movie | 120 | A new movie ... | |  |  |
| screens |  |  |  |  |  |  |
| 0 | A - IMAX | 5 | 5 | 0 | 0 | 0 |
| 1 | B - Reg | 9 | 9 | 0 | 0 | 0 |
| Showings |  |  |  |  |  |  |
| 0 | 0 | 0 | 2020 | 150 | 10 | 15 |
| 1 | 1 | 1 | 2020 | 150 | 11 | 15 |
|  |  |  |  |  |  |  |

|  |  |  |
| --- | --- | --- |
| Test | Input | Success criteria |
| (1)Create a theatre screen | Rows:5 Cols:5 Name: Screen A - IMAX | A new object should be created containing the 5 rows and cols.  View screen function will confirm this being displayed |
| (2)Create a new movie | Name: A movie Duration: 38 Description: this is a movie | A new object should be created with the same inputs.  View movies function will confirm this has been created. |
| (3) Change the availability of a seat | Input as test 1, but using change function, row: 0 col:1 | Seat (0,1) should now be displayed as red in the view screen function. |
| (4) View available seats | Using test data with ID 1. | Using test data view screen with ID 1 should be represented in a grid with the correct colours where green is free, Grey is booked and red is unavailable. |
| (5) To add a new showing | Using testing data Screen ID 0, Movie ID 0, Year 2020, day 150, hour: 10, min:15 | A new object should be created with the same as the inputs.  View all shows should show this data. |
| (6) To create a new showing which conflicts with a booked screen | Using testing data continue from test (5)  Screen ID 0, Movie ID 0, Year 2020,  Day 150, hour: 10 min:30 | As the movie will conflict with the running time of the showing in test 5 it should not allow the creation of the movie. |
| (7)create a new booking | Using testing data, booking showing 0, row 1 :col: 1 with name: nick email: [na@mail.com](mailto:na@mail.com) number: 070770707 | Seat should be free to book. Success should be shown, and it should now show up as grey on the view screen function. |
| (8) to try to book an already booked seat | Using the same details as in test 7, number: aaaa then 023178921. | The program should ask the user to re-enter the number as it is invalid, the user will then attempt to book the seat, which it should say is booked then continue. |
| (9) To check booking levels and details of customers via the showing report menu. | Using testing data checking show ID 1. | Should display a booking level of 4% and show all the booked seats details. |

# DEMO

The demo found at: will use the testing outline as a guide to show all the features of the program.

<https://youtu.be/JedsLGz0Go4>