**Report**

First of all with this assignment I started by reading through the case study and making a ‘to do list’. Within this to do list I listed all the documents needed to be completed and a list of everything the system needs. After I moved onto making a requirements list, which overall in this assignment has helped me because it gave me a good idea of all the functions needed within in the system and gave me a good scope on making the system. From the requirements list I was able to create a class diagram which opened my eyes to where I needed inheritance, polymorphism and encapsulation.

After making the class diagram I was then ready to write my pseudo code. I did encounter an issue when making this document which was that I didn’t entirely know what methods I was going to implement in some areas, so I had to not finish the document in some areas to then move onto coding the system. The way I coded the system first of all was by laying out the header files, main, classes, functions, variables and any includes needed. Then I started working through my requirement list on each function in order. I found quite a few issues with my code, one issue was that whenever a variable is of an integer type and required a user input, if the user entered a letter it would break the program, unfortunately I was never able to overcome this issue. Another problem I had to overcome was making the seating floor plan, the first issue was that I attempted to do this without initialising which I now realise was never going to work because I needed data to pass across functions instead of doing it all within one function. A problem I encountered as well was that in this project was that it required a lot of error handling in the sense of making sure the user doesn’t break the program by entering a wrong input. In some areas of my code I have demonstrated error handling and how I could deal with such situations, but again I didn’t have enough time to implement error handling in all of the code.

After coding I moved onto testing, testing in general was very revealing to me at where my code could be improved. So after one overall test I had to go back and edit my system in order to overcome some unexpected outcomes, an example of this was that in some cases it wasn’t letting the user input values with a cin, so instead I had to getline and within the parameters then get cin.

Overall with this assignment and module I think it has taught me a lot about C++ showing me a different way of programming and teaching me new techniques of inheriting data, passing data and ensuring that the code doesn’t break. Also I now have a different perspective on how I can approach presenting data to the user and realise now that it’s not as easy as it sounds.

**Code**

# **Main**

#include "Customer.h"

#include "show.h"

#include "showSeat.h"

#include "ticket.h"

int main()

{

int TicketCounter;

customer c;

c.GetLogin();

c.GetProfile();

show s;

s.DisplayShow();

s.SelectShow();

showSeat ss;

ss.Initialise();

ss.DisplaySeats();

TicketCounter = ss.GetSeats();

ss.DisplaySeats();

ticket t;

t.SetPrice(TicketCounter);

t.Deselect();

t.Pay();

t.PrintTicket();

system("PAUSE");

return EXIT\_SUCCESS;

}

# **Customer**

#pragma once

#include <iostream>

#include <iomanip>

#include <string>

using namespace std;

class customer

{

public:

void GetLogin();

void GetProfile();

private:

string email, password, e[10], p[10];

bool check = false, first = true;

int num = 0;

};

void customer::GetLogin()

{

cout << "Enter Email: ";

getline(cin, email);

e[num] = email;

cout << endl;

cout << "Enter Password: ";

getline(cin, password);

p[num] = password;

cout << endl;

num++;

if(!first)

{

for (int i = 0; i <= num; i++)

{

if (email == e[i] && password == p[i])

{

check = true;

}

}

}

first = false;

}

void customer::GetProfile()

{

string firstname, surname, address1, address2, postcode, vcode;

int mobile = 0;

if (check == true)

{

cout << "Welcome Back, please enter verification code: ";

getline(cin, vcode);

cout << endl;

}

else

{

cout << "Create new Profile" << endl;

cout << "First Name: ";

getline(cin, firstname);

cout << endl;

cout << "Surname: ";

getline(cin, surname);

cout << endl;

cout << "Address Line 1: ";

getline(cin, address1);

cout << endl;

cout << "Address Line 2: ";

getline(cin, address2);

cout << endl;

cout << "Postcode: ";

getline(cin, postcode);

cout << endl;

cout << "Mobile Number: ";

cin >> mobile;

cout << endl;

}

}

# **Show**

#pragma once

#include <iostream>

#include <iomanip>

#include <string>

using namespace std;

class show

{

public:

void DisplayShow();

void SelectShow();

protected:

bool initial = true;

int pickShow = 0;

};

void show::DisplayShow()

{

if (initial)

{

cout << endl;

cout << endl;

cout << "List of upcoming events..." << endl;

cout << "Select a show: " << endl;

cout << endl;

cout << "1: High School Musical 1 Our Fan Base is Under 14 Years Old 28/01/17" << endl;

cout << "2: High School Musical 2 We Can't Even Believe The First One Sold 15/02/17" << endl;

cout << "3: High School Musical 3 Guess What We Made Another One 28/02/17" << endl;

cout << "4: High School Musical 4 These Actors Are Definitely Too Old For High School 12/03/17" << endl;

cout << endl;

}

cout << "Select a show (Enter 1/2/3/4): " << endl;

cin >> pickShow;

cout << endl;

if (!initial)

SelectShow();

initial = false;

}

void show::SelectShow()

{

switch (pickShow)

{

case 1: cout << "You have chosen High School Musical 1" << endl;

break;

case 2: cout << "You have chosen High School Musical 2" << endl;

break;

case 3: cout << "You have chosen High School Musical 3" << endl;

break;

case 4: cout << "You have chosen High School Musical 4" << endl;

break;

default: cout << "Invalid Input" << endl, DisplayShow();

}

}

# **Show Seat**

#pragma once

#include <iostream>

#include <iomanip>

using namespace std;

class showSeat

{

public:

void Initialise();

void DisplaySeats();

int GetSeats();

protected:

char Seat[10][10];

int ArrayRows[10], ArrayColumns[10], ticketcount = 0;

};

void showSeat::Initialise()

{

cout << endl;

for (int Rows = 0; Rows < 10; Rows++)

{

for (int Columns = 0; Columns < 10; Columns++)

{

Seat[Rows][Columns] = 'A';

}

}

}

void showSeat::DisplaySeats()

{

cout << "Column: 0 1 2 3 4 5 6 7 8 9" << endl;

for (int Rows = 0; Rows < 10; Rows++)

{

cout << "Row: " << Rows << " ";

for (int Columns = 0; Columns < 10; Columns++)

{

cout << Seat[Rows][Columns] << " ";

}

cout << endl;

}

}

int showSeat::GetSeats()

{

cout << endl;

cout << "How many tickets would you like?" << endl;

cin >> ticketcount;

cout << endl;

for (int i = 1; i < (ticketcount + 1); i++)

{

cout << "Ticket: " << i << endl;

cout << "Pick your desired seat row: " << endl;

cin >> ArrayRows[i];

cout << endl;

cout << "Pick your desired seat column: " << endl;

cin >> ArrayColumns[i];

cout << endl;

for (int Rows = 0; Rows < 10; Rows++)

{

for (int Columns = 0; Columns < 10; Columns++)

{

if (Seat[Rows][Columns] == 'A')

{

Seat[ArrayRows[i]][ArrayColumns[i]] = 'H';

}

}

}

}

return ticketcount;

}

# **Ticket**

#pragma once

#include <iostream>

#include <iomanip>

#include <string>

#include <cstdlib>

#include "Show.h"

using namespace std;

class ticket

{

public :

void SetPrice(int counter);

void Deselect();

void Pay();

void PrintTicket();

protected :

int ticket\_number = 0, adult = 0, student = 0, child = 0, senior = 0, response = 0;

float ticket\_total = 0, adultPrice = 14.99, studentPrice = 11.99, childPrice = 7.99, seniorPrice = 9.99;

char CardNumber[16], ExpiryDate[5], CSC[3];

string name;

show x;

};

void ticket::SetPrice(int counter)

{

ticket\_number = counter;

cout << endl;

cout << "You have chosen " << ticket\_number << " ticket(s) to buy..." << endl;

cout << endl;

cout << "How many child tickets? 7.99" << endl;

cin >> child;

cout << endl;

cout << "How many student tickets? 11.99" << endl;

cin >> student;

cout << endl;

cout << "How many adult tickets? 14.99" << endl;

cin >> adult;

cout << endl;

cout << "How many senior citizen tickets? 9.99" << endl;

cin >> senior;

if (ticket\_number == (child + student + adult + senior))

{

ticket\_total = (adult \* adultPrice) + (student \* studentPrice) + (child \* childPrice) + (senior \* seniorPrice);

cout << endl;

cout << "The total for your " << ticket\_number << " ticket(s) comes to " << ticket\_total << endl;

}

else if (ticket\_number < (child + student + adult + senior))

{

cout << endl;

cout << "You've selected too many tickets.." << endl;

SetPrice(counter);

}

else if (ticket\_number > (child + student + adult + senior))

{

cout << endl;

cout << "You've not selected enough tickets.." << endl;

SetPrice(counter);

}

else

{

cout << endl;

cout << "Invalid Input" << endl;

SetPrice(counter);

}

}

void ticket::Deselect()

{

cout << endl;

cout << "Would you like to buy these tickets? (1 = yes or 2 = no)" << endl;

cin >> response;

cout << endl;

switch (response)

{

case 1: cout << "You have chosen to buy the ticket(s)." << endl, cout << endl, system("PAUSE");

break;

case 2: cout << "You have chosen to NOT buy the ticket(s)." << endl, exit(0);

break;

default: cout << "Invalid Input" << endl, Deselect();

}

}

void ticket::Pay()

{

system("cls");

cout << "Total: " << ticket\_total << " pounds" << endl;

cout << endl;

cout << "Enter Name Printed on Card: ";

cin >> name;

cout << endl;

cout << "Enter Card Number: ";

cin >> CardNumber;

cout << endl;

cout << "Enter Expiry Date: ";

cin >> ExpiryDate;

cout << endl;

cout << "Enter Card Security Code: ";

cin >> CSC;

cout << endl;

cout << endl;

cout << endl;

cout << "Your card has been approved" << endl;

cout << endl;

system("PAUSE");

}

void ticket::PrintTicket()

{

system("cls");

cout << "Ticket for High School Musical" << endl;

}