VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"JNANA SANGAMA", BELAGAVI - 590 018

A MINI PROJECT REPORT

on

# "MOVIE MANAGEMENT SYSTEM"

***Submitted by***


| | |
|---|---|
| **Sunidhi S Patwardhan** | **4SF19IS109** |
| **Swathi S Nayak** | **4SF19IS115** |


**BACHELOR OF ENGINEERING**

**in**

**INFORMATION SCIENCE & ENGINEERING**

***Under the Guidance of***


**Ms. Jayapadmini Kanchan,**

Assistant Professor,

Department of ISE,


**at**

# SAHYADRI

**College of Engineering and Management**

**Adyar, Mangaluru - 575 007**

**2021 - 22**

# SAHYADRI

## College of Engineering and Management

### Adyar, Mangaluru - 575 007

### Department of Information Science & Engineering

# CERTIFICATE

This is to certify that the mini project entitled **"Movie Management System"** has been carried out by **Sunidhi S Patwardhan (4SF19IS109) and Swathi S Nayak (4SF19IS115)** the bonafide students of Sahyadri College of Engineering and Management, Bachelor of Engineering in Information Science & Engineering of Visvesvaraya Technological University, Belagavi during the year 2021-22. It is certified that all corrections / suggestions indicated for internal assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of mini project work prescribed in File Structures Laboratory with Mini Project(18ISL67) for the said degree in sixth semester.

| | | |
|---|---|---|
| **Signature of the Guide1** | **Signature of the Guide2** | **Signature of the HOD** |
| Ms. Jayapadmini Kanchan | Mrs. Harinakshi C | Dr. Shamanth Rai |

## External Viva:

| Examiner's Name | Signature with Date |
|---|---|
| 1. ..... | ..... |
| 2. ..... | ..... |

# SAHYADRI

## College of Engineering and Management

### Adyar, Mangaluru - 575 007

### Department of Information Science & Engineering

# DECLARATION

We hereby declare that the entire work embodied in this Mini Project Report titled **"Movie Management System"** has been carried out by us at Sahyadri College of Engineering and Management, Mangaluru under the supervision of **Ms. Jayapadmini Kanchan,** for **Bachelor of Engineering** in **Information Science & Engineering**. This report has not been submitted to this or any other University for the award of any other degree.

**Sunidhi S Patwardhan (4SF19IS109)**

**Swathi S Nayak (4SF19IS115)**

Dept. of ISE, SCEM, Mangaluru

# Abstract

This project is an insight into designing and implementing a movie management system with the help of C++ programming language over file handling and file management systems. The project provides features for adding, searching, modifying, deleting, booking, and confirming tickets. In addition, it provides information about the total number of transactions, feedback, and ratings. It also provides the users with the functionality of viewing the movie receipt after payment. By developing a Movie Management System, the aim is to facilitate the booking of tickets more efficiently, simplify the user experience, and enable movies to become more accessible.

# Acknowledgement

It is with great satisfaction and euphoria that we are submitting the Mini Project Report on **"Movie Management System"**. We have completed it as a part of the curriculum of Visvesvaraya Technological University, Belagavi for the award of Bachelor of Engineering in Information Science & Engineering.

We are profoundly indebted to our guide, **Ms. Jayapadmini Kanchan**, Assistant Professor, Department of Information Science & Engineering for innumerable acts of timely advice, encouragement and We sincerely express our gratitude.

We express our sincere gratitude to **Dr. Shamanth Rai**, Head and Associate Professor, Department of Information Science & Engineering for his invaluable support and guidance.

We sincerely thank **Dr. Rajesha S**, Principal, Sahyadri College of Engineering and Management and **Dr. D. L. Prabhakara**, Director, Sahyadri Educational Institutions, who have always been a great source of inspiration.

Finally, yet importantly, we express our heartfelt thanks to our family and friends for their wishes and encouragement throughout the work.

**Sunidhi S Patwardhan (4SF19IS109)**
**Swathi S Nayak (4SF19IS115)**

# Table of Contents

# List of Figures

# Chapter 1

# Introduction

The Movie Management System is robust and integrated technology. It is used for managing the essential operations of cinema theatres. It is system of digital solutions which helps you manage all the operations within your cinema in a simpler manner, reducing the work load within the cinema to a great extent and helping to simplify all the operations. It maximises the operating efficiency by providing reliable, centralised control. It helps to streamline and automate all the operations giving the ultimate management solution, so that the users have centralized control over all the operations that go on within the cinema. It deals with booking tickets with ease. It is a user friendly application for both the users and the manager.

## 1.1 Purpose

The purpose of the Movie Management System is to present a system to increase the ease and efficiency of booking the movie tickets. It aims to provide a facility to book movie tickets anytime and anywhere, thereby reducing human resources. It enhances the user experience and provides an uninterrupted service while booking movie tickets promising consistency and optimisation.

## 1.2 Scope

This project focuses on buying movie tickets according to the customer preferences. Customers can buy tickets 24×7. It is developed keeping in view of the multiple movies running at different showtime in a theatre. It provides the flexibility for the customers to choose a movie with their preferred showtime. Customers can see a

graphical view of the seat availability and choose their desired seat. They can pay ticket amount online via credit card, debit card or even through cash which makes the payment easy.

## 1.3 Overview

This project facilitates customers to book movie tickets with ease. It contains a manager and customers where the manager has the authority to manage the movies. Managing includes the addition, deletion and updating of film from the files. Customers can book tickets for the movie they wish to watch. They even have the authority to cancel their booked tickets anytime with ease. An additional feature to this project is that they have the facility to even buy food and beverages prior. This makes it a plus point to contribute to the revenue along with the tickets being sold.

# Chapter 2

# Requirements Specification

## 2.1 Hardware Specification

- Processor : Intel(R) Core(TM) i5-1035G1 CPU @ 1.00GHz 1.19 GHz
- RAM : 8GB
- Hard Disk : 500GB
- Input Device : Standard keyboard and Mouse
- Output Device : Monitor

## 2.2 Software Specification

- Programming Language : C++
- IDE :Visual Studio Code

# Chapter 3

# System Design

## 3.1 Architecture Diagram

In this system, the two major focuses are on the customers and the manager as shown in the below figure 3.1.

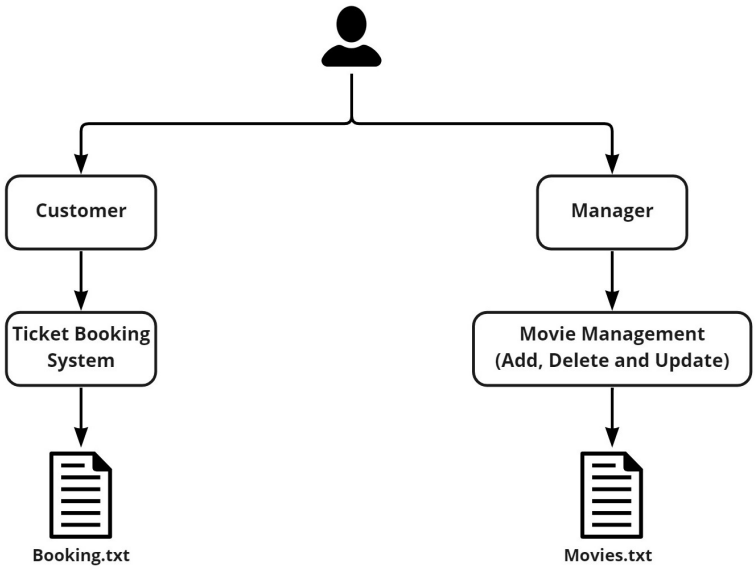


Figure 3.1: System Architecture Diagram

The manager has the control over managing the movies present in the files by adding, deleting or modifying the details of movie. Customer will interact with the system by choosing one of the options available like movie booking, cancelling a ticket and ordering food or beverages.

# Chapter 4

# Implementation

## Base Code

In figure 4.1, the base class of movie management system comprising of it's methods/functions as shown.

```
void title();
bool CompareNames(Movie movie1, Movie movie2);
int ReadFile(MOVIE &movies, int &movieCount);
void showmovielist(MOVIE &movies, const int movieCount);
void time_checking(int time[], int i);
void showTicket();
void payment(float pay);
void movie_booking();
void ticket_cancel();
void food_drinks();
void displayMenu();
void add_movie();
void delete_movie();
void update_movie();
void management();
string member_login();
```

Figure 4.1: Snippet of base class

## Main Menu

In figure 4.2, the main() is used to provide menu driven style to process the operations that one wishes to produce.

```
int main()
{
    time_t now = time(0);
    char *dt = ctime(&now);
    char selection_role;
    system("CLS");
    title();
    cout << "                  The local date and time is: " << dt << endl;
    cout << "                  ========================        ============================        =======================\n"
            "                  * (A)  MOVIE BOOKING   *        * (B) TICKET CANCELLATION *        * (C) SNACKS & DRINKS *\n"
            "                  ========================        ============================        =======================\n\n"
            "                  ========================        ========================\n"
            "                  * (D) MANAGE MOVIES    *         * (E)     QUIT       *\n"
            "                  =======================         ========================\n\n";

    do
    {
        cout << "                                                                SELECT AN OPTION: ";
        cin >> selection_role;
        cin.ignore(10000, '\n');
        selection_role = tolower(selection_role);
        if (selection_role < 'a' || selection_role > 'f')
            cout << "                                            You have entered a wrong selection. Try again.\n\n";
    } while (selection_role < 'a' || selection_role > 'f');

    if (selection_role == 'a')
    {
        movie_booking();
    }
    else if (selection_role == 'b')
    {
        ticket_cancel();
    }
    else if (selection_role == 'c')
    {
        food_drinks();
    }
    else if (selection_role == 'd')
    {
        management();
    }
    else
    {
        exit(1);
    }
}
```

Figure 4.2: Snippet of main menu

## Movie ticket booking

In figure 4.3, the movie_booking() is used to book the tickets for the movies.

```
void movie_booking()
{
    Customer *c1 = new Customer;
    fstream movie;
    movie.open("Movies.txt", ios::in);
    string showtime;
    int seat_price, lines, select_movie, selection_show;
    char response, r, r2, pay_res, reuse_response, seat_type, temp, seat_row[10], seat_column[10], response1;
    float remaining = 0, cash = 0;
    srand(time(0));
    int ticket_num;

login:
    ticket_num = (rand() % 99999) + 10000;
    system("CLS");
    title();
    cout << "\n\t\t\t\t*************************\n"
         << "\t\t\t\t   *    LOGIN PAGE   * \n"
         << "\t\t\t\t************************* \n\n\n";

    cout << "\t\t\tPRESS ANY KEY TO CONTINUE: ";
    cin >> response;
    response = tolower(response);
    system("CLS");
    title();

    c1->setName();
    c1->setEmail();
    c1->setPhonenumber();

movies:
    int movieCount{}, readStatus{};
    MOVIE movies;
    if (readStatus = ReadFile(movies, movieCount))
        showmovielist(movies, movieCount);
    cout << endl
         << endl;
    int p = sizeof(movie_name) / sizeof(movie_name[0]);

    cout << " Please select a movie [1,2,3....]: ";
    cin >> select_movie;
    cout << endl;

    while ((!cin >> select_movie) || (select_movie > counter - 1) || (select_movie < 1))
    {
        cout << " TRY AGAIN [1,2,3....]: ";
        cin.clear();
        cin.ignore(25, '\n');
        cin >> select_movie;
    }
```

Figure 4.3: Snippet of movie ticket booking

## Ticket receipt

In figure 4.4, movie ticket receipt is produced once the booking is confirmed.

```
receipt:
    system("CLS");
    cout << "=========================================================\n"
         << "=                         RECEIPT                       =\n"
         << "---------------------------------------------------------\n"
         << "=                       MOVIE TICKET                    =\n"
         << "=========================================================\n\n"
         << "\nTicket No. : " << ticket_num << endl
         << "\nTo Mr/Ms " << c1->getName() << ",\n"
         << "\nYou have booked " << movie_name[select_movie] << " at " << showtime << ", " << movie_day[select_movie];

    cout << "\nYour selection of seat(s) is : ";
    for (int i = 0; i < seat_no; i++)
    {
        cout << "(" << seat_row[i] << "," << seat_column[i] << ")";
        // The comma will not be print out after last seat
        if (i < (seat_no - 1))
        {
            cout << ", ";
        }
    }

    cout << "\nPrice of seat(s) : "
         << "RM " << seat_price << endl
         << endl;
    cout << "---------------------------------------------------------" << endl
         << left << setw(15) << "\nAmount to PAY : RM " << seat_price << endl;

    payment(seat_price);
```

Figure 4.4: Snippet of ticket receipt

## Payment

In figure 4.5, the payment() is used to make the payment for the ticket bought.

```
void payment(float pay)
{
    char resp;
    string card_type, card_num, ccv;
    cout << " Select the payment method: " << endl;
    cout << "                  (A) CASH  " << endl;
    cout << "                  (B) DEBIT CARD " << endl;
    cout << "                  (C) CREDIT CARD " << endl
         << endl;
    cout << " Your option [A, B, C] :  ";
    cin >> resp;
    while ((resp != 'A') && (resp != 'a') && (resp != 'B') && (resp != 'b') && (resp != 'C') && (resp != 'c'))
    {
        cout << " Please select a correct option [ a, b, c] : ";
        cin >> resp;
    }
```

Figure 4.5: Snippet of payment

## Ticket cancellation

In figure 4.6, the ticket_cancel() is used to cancel the booked tickets.

```
void ticket_cancel()
{

    fstream booking, cancelled_tickets;

    string search_receipt;
    char ticket_num[25], name[25], phone[25], email[40], seat_type[25], movie[25], day[25], showtime[25], seat_booked[60];
    bool isExist = 0, isFound = 0;
    char response;

    do
    {
        int lines = 0;
        system("CLS");
        title();

        cout << "\t\t\t\t\t\tTICKET CANCELLATION\n\n";
        cout << "\n\t\tEnter your ticket no. : ";
        cin >> search_receipt;

        booking.open("Booking.txt", ios::in);
        while (!booking.eof())
        {
            booking.getline(ticket_num, 25, '|');
            booking.getline(name, 25, '|');
            booking.getline(phone, 25, '|');
            booking.getline(email, 40, '|');
            booking.getline(movie, 25, '|');
            booking.getline(day, 25, '|');
            booking.getline(showtime, 25, '|');
            booking.getline(seat_type, 25, '|');
            booking.getline(seat_booked, 60);

            if (ticket_num == search_receipt)
            {
                isExist = 1;
                break;
            }
            else
            {
                isExist = 0;
            }
        }
        booking.close();
```

Figure 4.6: Snippet of ticket cancellation

## Management operations - Adding a movie

In figure 4.7, the manager can add a particular movie into the file based on movie release.

```
void add_movie()
{
    system("CLS");
    char name[25], day[25];
    int time[3] = {}, i = 0;
    fstream movie;
    movie.open("Movies.txt", ios::app);

    title();
    cout << "Adding new movie!!" << endl
         << endl;

    cin.ignore();
    cout << " Enter the movie name: ";
    cin.getline(name, 25);
    cout << endl;
    cout << " Enter the day of the showtime: ";
    cin.getline(day, 25);
    cout << endl;
    cout << " Enter the time for First show  (HHMM): ";
    cin >> time[0];
    cout << endl;
    time_checking(time, i);
    i = 1;

    cout << " Enter the time for Second show (HHMM): ";
    cin >> time[1];
    cout << endl;
    time_checking(time, i);
    i = 2;

    cout << " Enter the time for Third show  (HHMM): ";
    cin >> time[2];
    cout << endl;
    time_checking(time, i);

    movie << name << '|' << day << '|' << std::setfill('0') << std::setw(4) << time[0] << '|' << std::setfill('0')
     << std::setw(4) << time[1] << '|' << std::setfill('0') << std::setw(4) << time[2] << '\n';
    cout << endl;
    cout << "You are done, Manager!";
    movie.close();
}
```

Figure 4.7: Snippet of management operations - Addition

## Management operations - Deleting a movie

In figure 4.8, the manager can delete a particular movie from the file.

```
cout << "============= You have selected to delete " << movie_name[select_movie] << " ("
   << movie_day[select_movie] << ") "
        << "====================\n";

   int i = 0;

   for (int i = 0; i < lines; i++)
   {
       movie.getline(name, 25, '|');
       movie.getline(day, 25, '|');
       movie.getline(time1, 25, '|');
       movie.getline(time2, 25, '|');
       movie.getline(time3, 25);

       if (name == movie_name[select_movie] && day == movie_day[select_movie] &&
       time1 == showtime1[select_movie] && time2 == showtime2[select_movie] && time3 == showtime3[select_movie])
       {
           deleted_movies << movie_name[select_movie] << " (" << movie_day[select_movie] << ") " << showtime1[select_movie]
           << " " << showtime2[select_movie] << " " << showtime1[select_movie] << '\n';
           continue;
       }
       else
       {
           extra << name << '|' << day << '|' << time1 << '|' << time2 << '|' << time3 << '\n';
       }
   }

   extra.close();
   movie.close();
   movie.open("Movies.txt", ios::out);
   extra.open("extra.txt", ios::in);

   for (int y = 0; y < (lines - 2); y++)
   {
       extra.getline(name, 25, '|');
       extra.getline(day, 25, '|');
       extra.getline(time1, 25, '|');
       extra.getline(time2, 25, '|');
       extra.getline(time3, 25);
       movie << name << '|' << day << '|' << time1 << '|' << time2 << '|' << time3 << '\n';
   }

   extra.close();
   movie.close();
   deleted_movies.close();

   cout << "\n You have done deleting the showtime manager!!! \n";
   remove("Extra.txt");
}
```

Figure 4.8: Snippet of management operations - Deletion

## Management operations - Updating a movie

In figure 4.9, the manager can update a particular movie in the file.

```
while (!cin >> select_movie)
    {
        cout << " TRY AGAIN [1,2,3....]: ";
        cin.clear();
        cin.ignore(25, '\n');
        cin >> select_movie;
    }

    while ((select_movie > counter - 1) || (select_movie < 1))
    {
        cout << " TRY AGAIN [1,2,3....]: ";
        cin >> select_movie;
    }

    cout << "============= You have selected to update " << movie_name[select_movie] << " ("
    << movie_day[select_movie] << ") "
         << "====================\n";
    cout << endl;

    for (int i = 0; i <= lines; i++)
    {
        movie.getline(name, 25, '|');
        movie.getline(day, 25, '|');
        movie.getline(time1, 25, '|');
        movie.getline(time2, 25, '|');
        movie.getline(time3, 25);

        if (name == movie_name[select_movie] && day == movie_day[select_movie] && time1 == showtime1[select_movie]
        && time2 == showtime2[select_movie] && time3 == showtime3[select_movie])
        {
            cout << "Enter the day of the showtime: ";
            cin.ignore();
            string day1;
            cin >> day1;

            int time[3] = {}, x = 0;
            cout << " \nEnter the time for First show  (HHMM): ";
            cin >> time[0];
            time_checking(time, x);
            x = 1;

            cout << " \nEnter the time for Second show (HHMM): ";
            cin >> time[1];
            time_checking(time, x);
            x = 2;

            cout << " \nEnter the time for Third show  (HHMM): ";
            cin >> time[2];
            time_checking(time, x);
        }
    }
    movie.close();

    cout << "\n You have done editing the showtime manager!!! \n";
}
```

Figure 4.9: Snippet of management operations - Updating

# Chapter 5

# Results and Discussion

Below figure 5.1 shows the welcome page of the system.

```
*********************************************************************************

                                    MOVIE TICKET RESERVATION SYSTEM (MTRS)

                 ***********************************************************************************

                The local date and time is: Wed Jul 13 14:52:33 2022

                ========================        ============================        =======================
                * (A)  MOVIE BOOKING   *        * (B) TICKET CANCELLATION *        * (C) SNACKS & DRINKS *
                ========================        ============================        =======================

                ========================        ========================
                * (D) MANAGE MOVIES    *         * (E)     QUIT       *
                =======================         ========================

                                                    SELECT AN OPTION: A
```

Figure 5.1: Snippet of welcome page

Below figure 5.2 shows the customer registration page of the system.

```
*********************************************************************************

                                    MOVIE TICKET RESERVATION SYSTEM (MTRS)

                 ***********************************************************************************

                Enter your name : Trisha

                Enter your email address: trisha12@gmail.com

                Enter your phone number : 7894641236
```

Figure 5.2: Snippet of customer registration

Below figure 5.3 shows the number of movies available for screening in a theatre.

```
 ********************************************************************************

                                        MOVIE TICKET RESERVATION SYSTEM (MTRS)

                    ******************************************************************************************


 ==========================================================================
| SERIAL |=========================================================|  SHOWTIME  |
| NUMBER |                 MOVIE NAME                    |  DAY  |  AVAILABLE |
 ==========================================================================
   (1)     Avengers                                        MON    (1) 0900
                                                                  (2) 1000
                                                                  (3) 1100
   (2)     Avengers                                        WED    (1) 0900
                                                                  (2) 1000
                                                                  (3) 1100
   (3)     Avengers                                        FRI    (1) 0900
                                                                  (2) 1000
                                                                  (3) 1100
 ==========================================================================
   (4)     Dune                                            SAT    (1) 0900
                                                                  (2) 1100
                                                                  (3) 1330
 ==========================================================================
   (5)     Judge Dred                                      SUN    (1) 0930
                                                                  (2) 1030
                                                                  (3) 1530
 ==========================================================================
   (6)     Spaced Invaders                                 FRI    (1) 0200
                                                                  (2) 0400
                                                                  (3) 0600
   (7)     Spaced Invaders                                 SAT    (1) 0200
                                                                  (2) 0400
                                                                  (3) 0600
 ==========================================================================


 Please select a movie [1,2,3....]: 1

============== You have selected Avengers.=====================

Which show you want to select for [SHOW(1) / SHOW(2) / SHOW(3)]: 1
============== You have selected 0900.====================

Would you wish to [B]ack, [P]roceed or [R]eselect? :P
```

Figure 5.3: Snippet of displaying of the movies in the theatre

Below figure 5.4 shows the booked ticket receipt.

```
==========================================================
=                          RECEIPT                       =
----------------------------------------------------------
=                        MOVIE TICKET                    =
==========================================================


Ticket No. : 39992

To Mr/Ms Trisha,

You have booked Avengers at 0900, MON
Your selection of seat(s) is : (d,2), (d,3)
Price of seat(s) : RM 26

----------------------------------------------------------

Amount to PAY : RM 26
 Select the payment method:
                  (A) CASH
                  (B) DEBIT CARD
                  (C) CREDIT CARD

 Your option [A, B, C] :  A
```

Figure 5.4: Snippet of ticket receipt

Below figure 5.5 shows the ticket cancellation done by the customer.

```
TICKET CANCELLATION


                Enter your ticket no. : 39137
                Record is found!

                ==========================================================
                =                   BOOKING RECORD                       =
                ==========================================================

                Ticket No.     : 39137
                Name           : Trisha
                Phone no.      : 7894641236
                E-mail         : trisha12@gmail.com
                Movie selected : Avengers
                Day            : MON
                Showtime       : 0900
                Seat selected  : (d,2)(d,3)
                Seat type      : Normal seat

                ----------------------------------------------------------
                Are you sure you want to cancel your ticket booking?[y/n]: y

                Ticket cancel successfully !!!

                Do you want to use it again?[Y/n]: n
```

Figure 5.5: Snippet of ticket cancellation

# Chapter 6

# Conclusion

This project is developed in favour of Movie Management which helps in the easy execution and management of essential data. It helps in the effective management of movie ticket booking and the cancellation process. It provides the statistics about the movie tickets booked for the current movies running in the theatre with the additional feature of updating and deleting the movie. It works as per the user requirement and has the options accordingly. It also has the ability to print the receipt of the movie ticket booked. The main purpose is effective and easy handling of the movie ticket booking data and the cancellation. It is recommended that the system should be used with the necessary specifications of the system requirement and provision for the booking and cancelling of tickets. It should be made available throughout the hours of operation to make it relevant and useful.

# References


[1] Michael J. Folk, Bill Zoellick, Greg Riccardi: File Structures-An Object Oriented Approach with C++, 3rd Edition, Pearson Education, 1998.

[2] K.R. Venugopal, K.G. Srinivas, P.M. Krishnaraj: File Structures Using C++, Tata McGraw-Hill, 2008.

[3] Scot Robert Ladd: C++ Components and Algorithms, BPB Publications, 1993.

[4] Raghu Ramakrishan and Johannes Gehrke: Database Management Systems, 3rd Edition, McGraw Hill, 2003.