

**SILVER SCREEN**

**A MINI PROJECT**

**REPORT**

**Submitted by**

**HARSH ANKIT**

***In partial fulfilment for the award of  
the degree of***

***Bachelor of Engineering***

***In***

**INFORMATION SCIENCE AND ENGINEERING**

## *Certificate*

*This is to certify that the mini project work titled*

**SILVER SCREEN**

*Submitted in partial fulfilment of the degree of Bachelor of  
Engineering*

*In Computer Science and Engineering*

*Submitted by*

**HARSH ANKIT**

**1NH18IS143**

*DURING*

*EVEN SEMESTER 2019-2020*

*For*

*19ISE49*

Signature of Reviewer

Signature of HOD

**SEMESTER END EXAMINATION**

*Name of the Examiner*

*Signature with date*

1. \_\_\_\_\_

2. \_\_\_\_\_

## **ABSTRACT**

We all love holidays and on holidays majority of us decide to watch movies. Nowadays, going to multiplex have been the culture of almost all the families of today's generation as it is a time when the family can spend some time together by stealing some hours from their busy schedule. When holidays approach, everyone loves to make a plan of watching movies but then some reconsider their decision thinking about the long queues at the ticket counter and eventually drop their plan. In today's speedy ever-changing world patience is a hard attribute to acquire. However if one stands in a queue there are several downsides other than just wasting of time most of the times people will have to settle for seat at less satisfactory positions than they would have wanted. One make a lot of plan when a new movie is going to be released, they get excited for that movie but they are not the only ones who wants to watch the same movie at the same time and same place. Few lucky people get the ticket, few less lucky get the ticket but not an appropriate seat and the unluckiest ones don't even get a ticket and end up watching some other movies about which they are necessarily not interested in. Not only the customers, due to long queue and fewer employees, the employees often have to deal with mammoth tasks of dealing with huge number of people who are constantly complaining which is highly demotivating and leads to low service quality.

Considering these problems, this project is generated by using java application GUI program that allows the employees and the customers to solve these petty issues. Using this software, the data are arranged in proper manner which reduces the time of involvement and increase the production rate. It is beneficial for everyone. People can book movies anytime and anywhere without standing in a long queue. Even the admin or employ can add, delete, or update the movies and retrieve the data of people booking the seats. This software is made by applying OOPS concept in Java and for storing data in database MySQL is used.

**Keywords:** queue, messing of data, solving problem by this application, increase customers, easy booking, using OOPS concept and MySQL database storing.

## **ACKNOWLEDGEMENT**

The satisfaction and euphoria that accompany the successful completion of any task would be impossible without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

I have great pleasure in expressing gratitude to **Dr. Mohan Manghnani**, Chairman of New Horizon Educational Institutions for providing necessary infrastructure and creating good environment.

I take this opportunity to express my profound gratitude to **Dr. Manjunatha**, Principal NHCE, for his constant support and encouragement.

I am grateful to **Dr. C.S.R Prashanth**, Dean Academics, for his unfailing encouragement and suggestions, given to me in the course of my project work.

I would also like to thank **Dr. R.J. Anandhi**, Professor and Head, Department of Information Science and Engineering, for her constant support.

I express my gratitude to **Mr. Saravanan K.**, Assistant Professor, my project guide, for constantly monitoring the development of the project and setting up precise deadlines. His valuable suggestions were the motivating factors in completing the work.

**HARSH ANKIT**

**(1NH18IS143)**

## **CONTENTS**

<b>ABSTRACT</b>	<b>I</b>
<b>ACKNOWLEDGEMENT</b>	<b>II</b>
<b>LIST OF FIGURES</b>	<b>V</b>
<b>LIST OF TABLES</b>	<b>VI</b>
<b>1. INTRODUCTION</b>	
<b>1.1 PROBLEM DEFINITION</b>	<b>1</b>
<b>1.2 OBJECTIVES</b>	<b>2</b>
<b>1.3 METHODOLOGY TO BE FOLLOWED</b>	<b>2</b>
<b>1.4 EXPECTED OUTCOMES</b>	<b>2</b>
<b>1.5 HARDWARE AND SOFTWARE REQUIREMENTS</b>	
<b>1.5.1 HARDWARE SYSTEM CONFIGURATION</b>	<b>3</b>
<b>1.5.2 SOFTWARE SYSTEM CONFIGURATION</b>	<b>3</b>
<b>2. OBJECT ORIENTED CONCEPTS</b>	<b>4</b>
<b>2.1 CLASS</b>	<b>5</b>
<b>2.2 OBJECT</b>	<b>7</b>
<b>2.3 INHERITANCE</b>	<b>8</b>
<b>2.4 POLYMORPHISM</b>	<b>8</b>
<b>2.5 ABSTRACT CLASS</b>	<b>9</b>
<b>2.6 MULTITHREADING</b>	<b>10</b>
<b>2.7 I/O FUNCTIONS</b>	<b>10</b>
<b>2.8 JAVA PACKAGES</b>	<b>11</b>

<b>2.9 EXCEPTION HANDLING</b>	<b>12</b>
<b>2.10 ENCAPSULATION</b>	<b>12</b>
<b>3. DESIGN</b>	
<b>3.1 DESIGN GOALS</b>	<b>14</b>
<b>3.2 ALGORITHM</b>	<b>15</b>
<b>4. IMPLEMENATION</b>	
<b>4.1 ADMIN MODULE</b>	<b>17</b>
<b>4.2 SIGNUP MODULE</b>	<b>17</b>
<b>4.3 TICKET MODULE</b>	<b>18</b>
<b>4.4 BOOKING MODULE</b>	<b>18</b>
<b>4.5 DATABASE</b>	<b>18</b>
<b>5. RESULTS</b>	
<b>5.1 RESULT VERIFICATION FOR INHERITANCE</b>	<b>19</b>
<b>5.2 RESULT VERIFICATION FOR POLYMORPHISM</b>	<b>20</b>
<b>5.3 RESULT VERIFICATION FOR PACKAGES AND CLASS</b>	<b>20</b>
<b>5.4 RESULT VERIFICATION FOR EXCEPTION HANDLING</b>	<b>21</b>
<b>5.5 RESULT VERIFICATION FOR I/O FUNCTIONS</b>	<b>22</b>
<b>5.6 RESULT VERIFICATION FOR MULTITHREADING</b>	<b>23</b>
<b>6. CONCLUSION</b>	<b>29</b>
<b>7. REFERENCES</b>	<b>30</b>

## **LIST OF FIGURES**

<b><u>Fig. No.</u></b>	<b><u>Figure Description</u></b>	<b><u>Page No.</u></b>
1	OBJECT ORIENTED CONCEPTS IN JAVA	4
2	COMMONLY USED INHERITANCE TYPES	5
3	EXAMPLE FOR OBJECT IMPLEMENTATION	7
4	COMMONLY USED INHERITANCE TYPES	8
5	TYPES OF POLYMORPHISM	9
6	EXAMPLE OF ABSTRACT CLASS	9
7	MULTITHREADED PROGRAM GRAPH	10
8	OVERVIEW OF I/O STREAMS	11
9	IMPORTING ONE PACKAGE TO ANOTHER	11
10	GENERAL FORM OF EXCEPTION HANDLING	12
11	ENCAPSULATION	13
12	DIAGRAM FOR DATA FLOW	14
13	DIAGRAM SHOWING CONCEPTUAL SCENARIO	14
14	FRAME FOR LOGIN CLASS	24
15	FRAME FOR ADMIN CLASS	24
16	FRAME FOR INFO CLASS	25
17	FRAME FOR SIGNIN CLASS	25
18	FRAME FOR BOOKING CLASS	26

<b>19</b>	<b>FRAME FOR DETAIL CLASS</b>	<b>26</b>
<b>20</b>	<b>FRAME FOR MOVIE CLASS</b>	<b>27</b>
<b>21</b>	<b>FRAME FOR TICKET CLASS</b>	<b>27</b>

## **LIST OF TABLES**

<b><u>Table No</u></b>	<b><u>Table Description</u></b>	<b><u>Page No</u></b>
<b>1</b>	<b>ADMIN TABLE</b>	<b>28</b>
<b>2</b>	<b>SIGNUP TABLE</b>	<b>28</b>
<b>3</b>	<b>MOVIE DETAIL TABLE</b>	<b>28</b>
<b>4</b>	<b>BOOKING TABLE</b>	<b>28</b>
<b>5</b>	<b>TRANSACTION TABLE</b>	<b>28</b>



## CHAPTER 1

### **1. INTRODUCTION:**

Nowadays, going to cinemas has been the culture of almost all the families of today's generation as it is a time when the family can spend some time altogether by stealing some hours from their busy schedule. The project silver screen is a java application GUI program that allows the movie theatre owner and the viewers to handle all the cinema activities online quickly and safely. Using this project the customers can book tickets for cinema halls online, through which they can book tickets anytime and anywhere and also can gather information about the movies online. User can easily be able to know about the movies released and then make choice. Admin can use this project to insert and delete data which is updated in database. This project provides ticket booking for the movie, seat management and ticket cancellation. Customers can know when the seats are full or which seat is empty so there is no waste of time. There is no mess in arranging data and is convenient to find where there is fault. Payments are done with the help of different online transaction methods.

#### **1.1 PROBLEM DEFINITION:**

On holidays mostly people decide watching movies at halls. For booking tickets, there is a long queue and when we don't know about the movie we sometime end in watching movies which you are not interested to watch. Due to long queue and less employ, the cinema hall employs also feel difficulties in booking. Sometimes we do not get the best seat so we need to cancel the plan. In order to solve all these problems, silver screen project is made where customers can book their interest movies and can book best seats. They can pay online so they don't need to stand in a long queue. Also the admin can add, delete movies which are showing in hall. It is

convenient for them to help the customers. Customers do not need to travel separately in order to book tickets, can book anywhere and anytime.

## **1.2 OBJECTIVES:**

In this project, the main aim is to provide another way for customers to buy cinema ticket. This system is basically aimed to provide the customer the complete information of the movie, according to which the customer can book the tickets anytime and from anywhere. They do not need to travel separately to book tickets and stand in queue. The program allows the owner to keep track of available seats for a particular movie and even maintain various details of the audiences.

## **1.3 METHODOLOGY TO BE FOLLOWED:**

- a. This silver screen ticket booking system is made by using eclipse and MySQL database for booking the movie tickets
- b. The Admin can login into the system and edit the details for example adding movies with its releasing date, rating, description
- c. Admin not only can edit but also delete and update the movie details
- d. The user can login into the system and book the movie seat by analysing the movie details
- e. Admin can inspect any booking details easily.

## **1.4 EXPECTED OUTCOMES:**

- a. It will successfully establish connection between admin and customers
- b. Admin can add movies which are showing and delete unwanted movies easily
- c. There is increase in profit as it is very convenient to book a ticket and so many people get enrolled in booking

- d. Booking becomes easy, informative, time saving, effective
- e. Look for the desire seat categories and book it with online transaction
- f. Admin gets the details of the customer which will be updated in database of the hall.

## **1.5 HARDWARE AND SOFTWARE REQUIREMENTS:**

### **1.5.1 HARDWARE SYSTEM CONFIGURATION:**

Processor	- Intel Core i5
Speed	- 1.8 GHz
RAM	- 256 MB (min)
Hard Disk	- 10 GB

### **1.5.2 SOFTWARE SYSTEM CONFIGURATION:**

Operating System	- Windows 7/8.1/10
Programming Language	- OOPS with JAVA
Compiler	- Eclipse swing , MySQL database

## Chapter 2

### **2. OBJECT ORIENTED CONCEPTS**

OOPS concepts in Java are the main ideas behind Java's Object Oriented Programming. They are an abstraction, encapsulation, inheritance and polymorphism. Grasping them is key to understanding how Java works. Basically, Java OOP concepts let us create working methods and variables, then re-use all or part of them without compromising security. In Java, abstraction means simple things like objects, classes and variables represent more complex underlying code and data. It prevent in repeating the same work multiple times. Inheritance let programmers to create new classes that share some of the attributes of existing classes. Polymorphism is used to let programmers use the same word to mean different things in different contexts which may be in method overloading form or method overriding form.

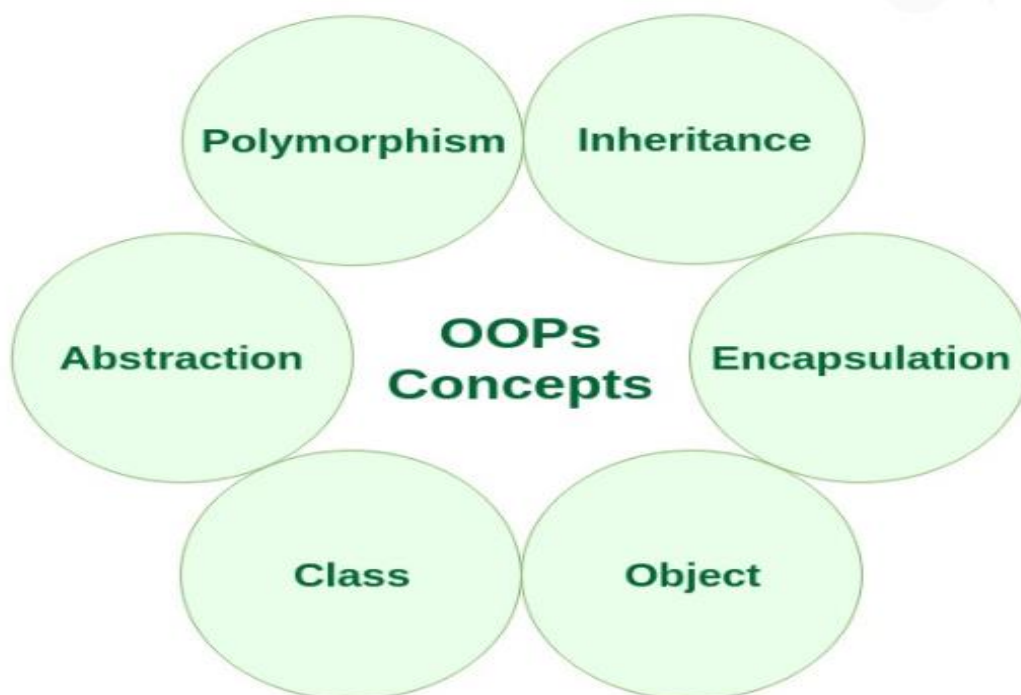


Figure 1: OBJECT ORIENTED CONCEPTS IN JAVA

## 2.1 CLASS :

A class can be defined as a blueprint (template) that describes the behaviors or state that object of its type support and includes or represents the set of the properties or methods that are common to all object of one type. It includes modifiers, class name, super class, and body. We declare its exact form and nature, by specifying the data that it contains and the code that operates on that data while defining a class. A class is declared by the use of the class keyword. The methods and variables defined within the class are called members of the class. The class is at the core of JAVA. All the concepts to be implemented in Java must be encapsulated within a class.

```
class classname {  
    type instance-variable1;  
    type instance-variable2;  
    // ...  
    type instance-variableN;  
  
    type methodname1(parameter-list) {  
        // body of method  
    }  
    type methodname2(parameter-list) {  
        // body of method  
    }  
    // ...  
    type methodnameN(parameter-list) {  
        // body of method  
    }  
}
```

Figure 2: GENERAL FORM OF A CLASS

Classes used in this project:

### 1. Login:

This class is used for the login purpose to check whether the user is customer or admin. The user must first register before logging in. The admin cannot be added at the register page, it is only possible at the database. But admin can change its password at admin class.

### 2. Signup:

This class is used for registration purpose to check who used the software. The user must be registered in order to enter into the page. The user must type the details like name, email, address, phone number, password and a unique username. All these details are entered into the database when registered. Hence he/she can login at the login page.

### 3. Admin:

This class is for admin to edit the software details. Admin can change his password here. Admin can add a movie with its details like its release date, rating, director name and its description. He can delete the movie by typing only the name of the movie. He can even update details of a movie. Admin can even look who have booked through this page.

### 4. Booking:

This is for the customers to look for the available movies. Here three button are given which when clicked leads to three different frames. The frames name are movies, ticket and details.

### 5. Ticket:

This is for actual booking process where data are entered. Here the user is asked to fill the movie name, time, date, account number, amount of money he will transfer and even his name and phone number.

### 6. Movies:

This is for viewing available movies. Here the posters of now showing movies is shown where the name of the movies are given. User can see those picture and click book button in order to go to the booking page that is ticket class.

### 7. Info:

This is for admin to get the booking details. Admin can view who has book the seats from this page. It is directly connected to database. When a user book its ticket, the data are automatically shown here.

### 8. Details:

This is for knowing the movies information. Here the user can view the information that the admin has entered about the movies. It is also connected to database and the results are fetched from the database directly.

## 2.2 OBJECT:

Objects have states and behaviors. For example: a dog is an object having states color, name, breed and behaviors wagging, barking, eating. Object is an instance of a class. For obtaining objects of a class first we need to declare a variable of the class type then, we must acquire an actual, physical copy of the object and assign it to that variable using new operator which at run time dynamically allocates memory for an object and return a reference to it .

```
class Lamp {
    boolean isOn;

    void turnOn() {
        isOn = true;
    }

    void turnOff() {
        isOn = false;
    }
}

class ClassObjectsExample {
    public static void main(String[] args) {
        Lamp l1 = new Lamp(); // create l1 object of Lamp class
        Lamp l2 = new Lamp(); // create l2 object of Lamp class
    }
}
```

Figure 3: EXAMPLE FOR OBJECT IMPLEMENTATION

## 2.3 INHERITENCE:

When one class acquires the properties of another class it is called inheritance. A class that is inherited is called superclass and the class that does the inheriting is called a subclass. In-order to inherit a class `extends` keyword is used. Inheritance is one of the cornerstones of object-oriented programming because it allows the creation of hierarchical classifications. There are five types of inheritance – single level inheritance, multi-level inheritance, hierarchical inheritance, multiple inheritance and hybrid inheritance. Among them hybrid inheritance is not possible in Java and multiple inheritance is only possible in interface concept.

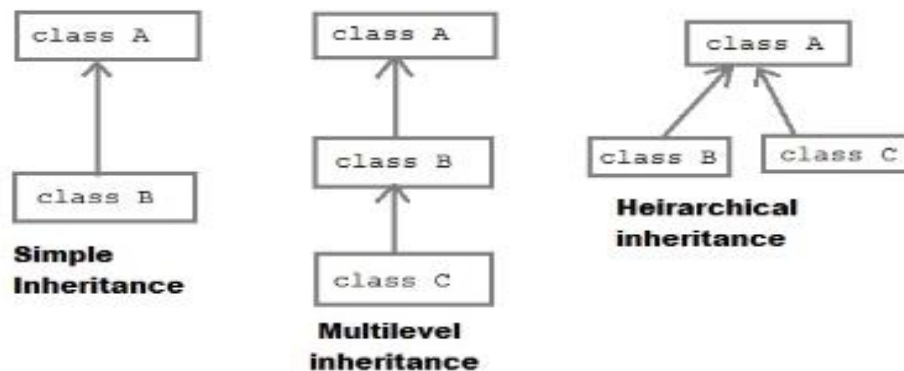


Figure 4: COMMONLY USED INHERITANCE TYPES

## 2.4 POLYMORPHISM:

Polymorphism in Java is a concept by which we can perform a single action in different ways. Polymorphism is derived from two Greek words: *poly*(many) and *morphs*(forms). There are two types of polymorphism in Java: compile time polymorphism and runtime polymorphism. Polymorphism can be performed by method overloading and method overriding. Dynamic method dispatch is runtime polymorphism where a call to an overridden method is resolved at runtime rather than compile time. Method overloading is said to be done when a class has more than one method with the same name but different argument lists. Method overriding is said to be done when a method which is declared in a parent class is also declared in a child class.



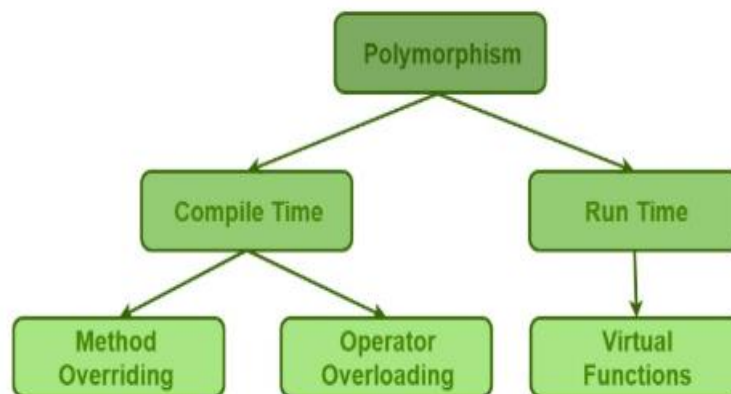


Figure 5: TYPES OF POLYMORPHISM

## 2.5 ABSTRACT CLASS:

When one class is declared abstract which may or may not include abstract methods is called abstract class. Abstract class can be sub classed but cannot be instantiate. Abstract class are similar to interfaces. Here we can declare fields that are not static and final, and define public, protected, and private concrete methods. Java abstract class can implement interfaces without even providing the implementation of interface methods. It is used to provide common method implementation to all the subclasses or to provide default implementation.

```
public abstract class Shape {  
    private String color;  
  
    public Shape() {}  
  
    public String getColor() {  
        return color;  
    }  
    public void setColor(String color) {  
        this.color = color;  
    }  
    public abstract double getArea();  
    public abstract double getPerimeter();  
}
```

Figure 6: EXAMPLE OF ABSTRACT CLASS

## 2.6 MULTITHREADING:

Multithreading is the method of executing two or more threads simultaneously to maximum utilization of CPU. Here two threads run concurrently hence also known as concurrency in Java. Multiple threads don't allocate separate memory area, hence they save memory. There is asynchronous behaviour of the programs in multithreading but when there is a need to access the shared resources by two or more threads then synchronization approach is utilized. It is the specialized form of multitasking.

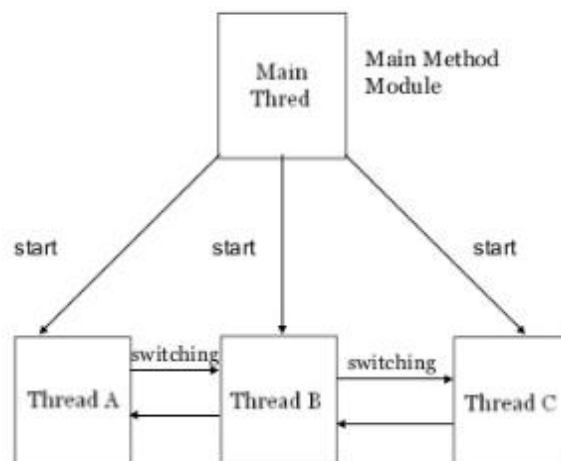


Figure 7: MULTITHREADED PROGRAM GRAPH

## 2.7 I/O FUNCTIONS:

Java I/O is used to process the input and produce the output. Java uses the stream concept to make I/O operations fast. The `java.io` package contains all the classes required for input/output operations. A stream is an abstraction that either produces or consumes information. A stream is linked to a physical device by the Java I/O system. Same I/O classes and methods can be applied to different types of devices.

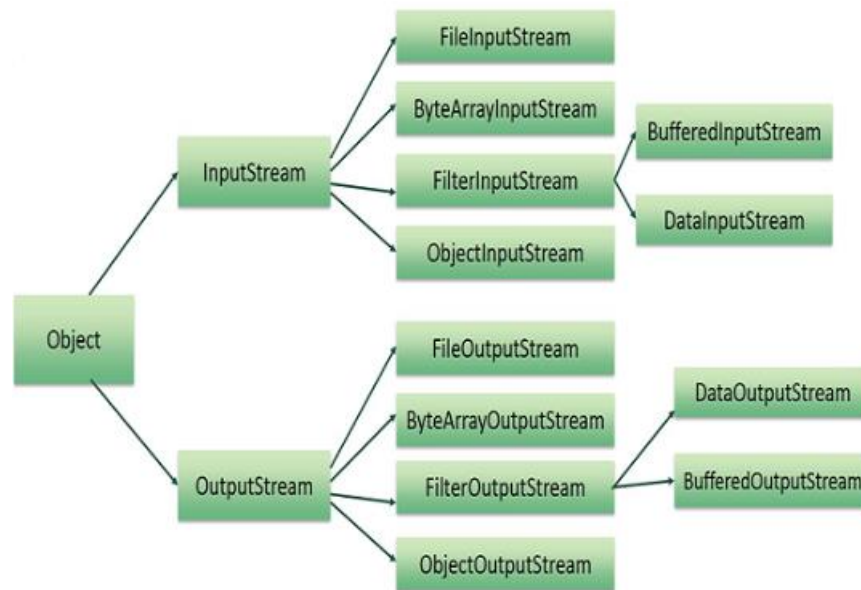


Figure 8: OVERVIEW OF I/O STREAMS

## 2.8 JAVA PACKAGES:

A mechanism for partitioning the class name space into more manageable chunks is the package in Java. The package is both a naming and visibility control mechanism. To create a package, simply include a package command as the first statement in the source file. The package statement defines a name space in which classes are stored. If no package statement is written, the class are put into default package. Import statement is used to bring certain classes or entire packages into visibility.

```
package com.javaguides.teacher;

import com.javaguides.course.Course;

public class Teacher {

    List<Course> courses = new ArrayList<>();
    public void addCourse(Course course) {
        courses.add(course);
    }
}
```

Figure 9: IMPORTING ONE PACKAGE TO ANOTHER

## 2.9 EXCEPTION HANDLING:

Java exception is an object that describes an exceptional (error) condition that has occurred in the piece of code. When error arises an object representing that exception is created and thrown in the method that caused the error which may choose whether to handle or pass the exception. There are five keywords in Java exception handling – try, catch, throw, throws, and finally. The risky statement to be monitored is contained within try block, if exception occur it is thrown and exception is catch and handled. Throw keyword manually throw an exception. Exception thrown out of method is specified by throws clause. The code that must be executed after a try block completes is put in finally block.

```
try {  
    // block of code to monitor for errors  
}  
  
catch (ExceptionType1 exOb) {  
    // exception handler for ExceptionType1  
}  
  
catch (ExceptionType2 exOb) {  
    // exception handler for ExceptionType2  
}  
// ...  
finally {  
    // block of code to be executed after try block ends  
}
```

Figure 10: GENERAL FORM OF EXCEPTION HANDLING

## 2.10 ENCAPSULATION:

Encapsulation is one of the most important principles of Object-Oriented Programming. It is a mechanism of wrapping the data(variables) and code acting on the methods together as a single unit. Here the variables or data of a class is hidden from any other class and can be accessed only through any member function of own class in which they are declared. Data are hidden from other classes with the help of data hiding concept which is achieved by making the members or methods of the class as private and the class is exposed to the end user or the world without

providing any details behind implementation using the abstraction concept, so it is also known as combination of data-hiding and abstraction. Encapsulation can be achieved by declaring all variables in the class as private and writing public methods in the class to set and get the values of variables.

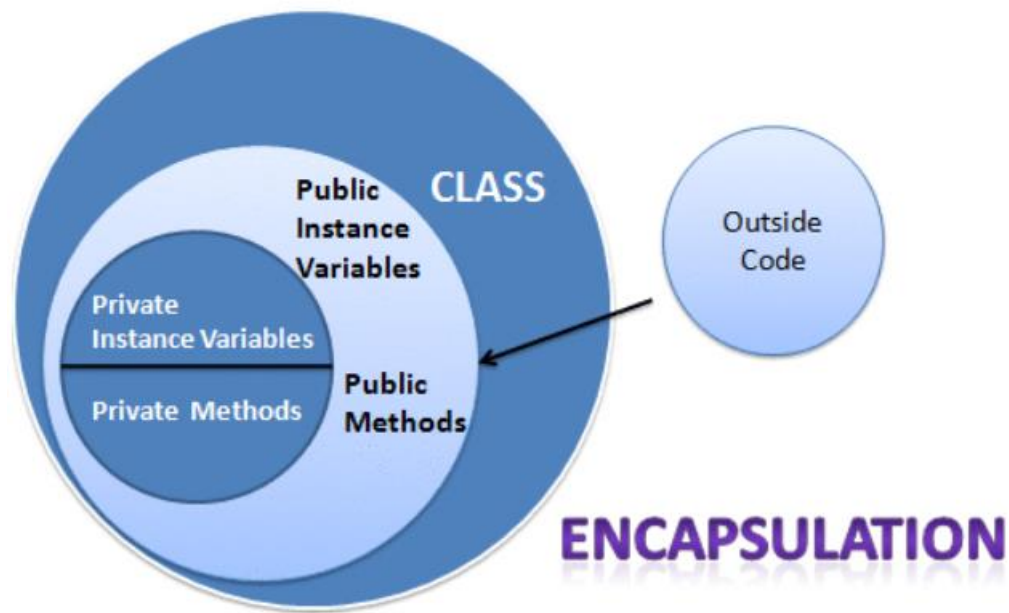


Figure 11: ENCAPSULATION

## Chapter 3

### 3. DESIGN

#### 3.1 DESIGN GOALS:

Data flow diagram is used to show the flow of data from external entities into the system. The data flow diagrams are pictorial or graphical representation of the Silver Screen ticket reservation system . It covers all the processes and data storage area.

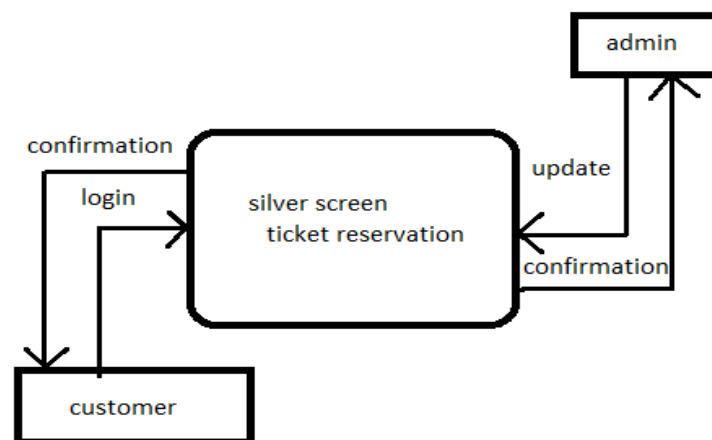


Figure 12: DIAGRAM FOR DATA FLOW

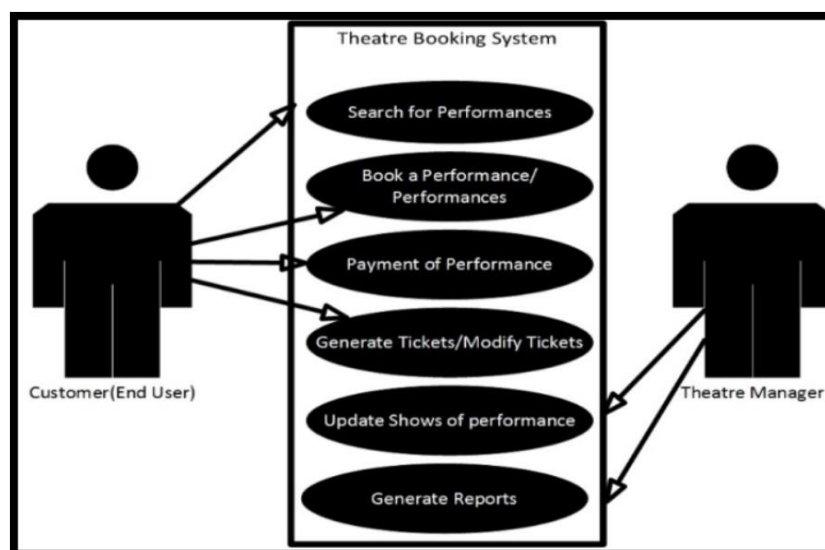


Figure 13: DIAGRAM SHOWING CONCEPTUAL SCENARIO

### 3.2 ALGORITHM:

1. This project is totally based on connecting people with the easiest method of booking and easiest way of formatting the data. It is the easiest way of communicating,
2. First the customer or the admin needs to sign in,
3. User, if don't have any account, can click the sign in link and register. Then can login into the system,
4. By clicking the sign in link the user will be asked to enter details like name, email, password, phone number, gender, and so on. All these data entered will be stored in the database,
5. If the user wants to login, is already registered, he can just type the correct username and password and login in the system and the class booking will be called,
6. When admin enter the username and password, admin class is called where he can add edit the details or change the password,
7. Login is only possible if he/she have registered ,
8. At booking class, the user can click the movie detail button to get all the movies details or can click the picture now showing to get the now showing movies posters,
9. The user can click the book now button to book his ticket and hence ticket class is called,

10. User then need to type the details like name, number, account no, seat type, quantity, time, date and the amount he/she wants to pay at the moment,
11. Noted that the amount must be paid from other application like PayPal, Google Pay, and so on ,
12. The booking is only confirmed if from the account no given some amount of money is transferred to the given number,
13. By clicking the book button , the user just register for the ticket. He just reserve his ticket,
14. All the data entered are saved in database which can be viewed by admin in admin class page,
15. The seat reservation process is completed by clicking book button and hence the user needs to show the payment detail while entering the hall.



## Chapter 4

### **4. IMPLEMENTATION**

#### **4.1 ADMIN MODULE:**

The admin has full access to the system. The admin can graphically view all the details, and has the authority to change the cost of the tickets, ticket availability and much more. The admin is provided with an id and password. He can put various kinds of notifications on the website. He can also monitor the transactions made throughout the day and the month and an algorithm will check the progress in the booking of tickets. He also can give various kinds of offers to the viewers. He can add new movies with their details. He also can delete movies or update the existing movie details. Admin can view the details of all the booking. Admin also has the access to the database where all the information are stored. This software helps in maintaining the record and updating the data in the software.

#### **4.2 SIGNUP MODULE:**

In signup module (i.e. user profile), the customer have to login first. After that they can check movies that are available and also the availability of tickets and their category. The category can be ODC, balcony, super balcony, or box. And they can also book their ticket according to their budget and need. Customer profile contains their name, contact details, address, and other necessary personal data which they need to enter for booking the movie ticket, that will also make the user as well as the admin of the hall to interact with each other easily. They can pay the amount and if they wish can pay at the counter without standing in a queue but, they must deposit some advance amount to confirm their movie tickets using different apps using the same account no. entered in the form then only the booking is confirmed.

### **4.3 TICKET MODULE:**

The customer can quickly search for the desired movie and the number of seats from the various options available. This all will be so user-friendly that the audience will not find any trouble in booking the movie ticket. Once the available movies are searched, and viewer finds the movie of his choice then this module helps the audience to book the movie ticket. Payment can be done by using PayPal, Paytm, UPI or Google Pay. The customer is asked for the details like name of the movie, time, date, seat number, seat type, his name and phone number and amount he want to pay whether full amount or only for reservation as well as his account number through which transaction is made.

### **4.4 BOOKING MODULE:**

This module is connected to movies module as well as details and ticket module. Movies module will help the viewer to see the movies available for the show. This is connected to the details module where details of all the movie is available that may be running or were run at the past. It is also connected to the ticket module where ticket is booked. This module help the user to see three functions. One is the movies now showing, other is the details of movie shown till now, and third is the connection to the booking or transaction page.

### **4.5 DATABASE:**

At database all the login details and signup details are saved. Using the saved data a person can login if the data is matched. All the movies details are also saved in the database. When a person book the movie, the booking details are also saved in the database which can be viewed by the admin. All the edition and booking are done at database using this software. This software is connected to database for convenient retrieval of data.

## Chapter 5

### 5. RESULTS

#### 5.1 RESULT VERIFICATION FOR INHERITANCE

```
import java.awt.BorderLayout;
public class admin extends login{

    public JFrame frame;
    private JTextField NAME;
    private JTextField DATE;
    private JTextField DESC;
    Connection con=null;

    PreparedStatement pat=null;
    ResultSet rs=null;
    private JTextField RATE;
    private JTextField id;
    private JTextField pass;
    private JTextField dir;
    /**
     * Launch the application.
     */
    public static void main(String[] args) {

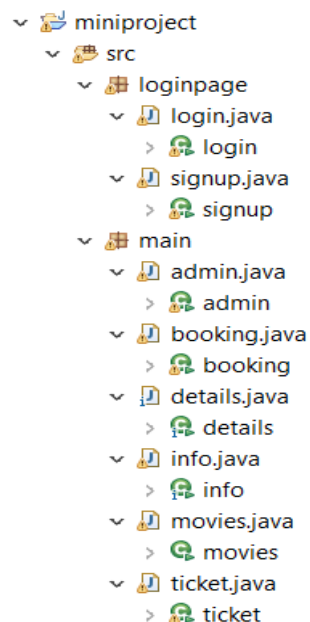
        EventQueue.invokeLater(new Runnable() {
            public void run() {
                try {
                    admin window = new admin();
                    window.frame.setVisible(true);
                } catch (Exception e) {
                    e.printStackTrace();
                }
            }
        });
    }
}
```

## 5.2 RESULT VERIFICATION FOR POLYMORPHISM

```
JLabel linkdes = new JLabel("Don't have any account?? CLICK HERE->");
linkdes.setBounds(20, 194, 250, 17);
frame.getContentPane().add(linkdes);

JLabel signinlink = new JLabel("SIGN IN");
signinlink.setFont(new Font("Tahoma", Font.BOLD, 14));
signinlink.addMouseListener(new MouseAdapter() {
    @Override
    public void mouseClicked(MouseEvent e) {
        signup.main(null);
    }
});
signinlink.setForeground(Color.RED);
signinlink.setBounds(276, 191, 84, 18);
frame.getContentPane().add(signinlink);
```

## 5.3 RESULT VERIFICATION FOR PACKAGES AND CLASS



## 5.4 RESULT VERIFICATION FOR EXCEPTION HANDLING

```
JButton userlog = new JButton("LOGIN");
userlog.setFont(new Font("Tahoma", Font.BOLD, 14));
userlog.addActionListener(new ActionListener() {
    public void actionPerformed(ActionEvent e) {
        int c=0;
        String pass=passworduser.getText();
        String driver="com.mysql.jdbc.Driver";
        String url="jdbc:mysql://localhost:3306/userdetails";

        try {
            Class.forName(driver);
            Connection con=DriverManager.getConnection(url, "root", "1234");
            String query="Select username,password from userinfo where username='"+nameuser.getText()+"'";
            PreparedStatement pa=con.prepareStatement(query);
            ResultSet rs=pa.executeQuery();
            while(rs.next()) {
                if(pass.contentEquals(rs.getString("password"))) {
                    JOptionPane.showMessageDialog(null,"WELCOME TO SILVERSCREEN");
                    c=1;
                    frame.setVisible(false);

                    booking.main(null);

                }
            }
            if(c==0)
            {
                JOptionPane.showMessageDialog(null,"login not successfull.. try again");
            }
        } catch (Exception e2) {
            JOptionPane.showMessageDialog(null,"login not successfull");
        }
    }
});
```

## 5.5 RESULT VERIFICATION FOR I/O FUNCTIONS

```
public void actionPerformed(ActionEvent e) {
    String filepath="C:\\Users\\Stuti\\Desktop\\table.txt.txt";
    File file=new File(filepath);
    try {
        BufferedReader br=new BufferedReader(new FileReader(file));
        String firstLine=br.readLine().trim();
        String[] columnsName=firstLine.split(" , ");
        DefaultTableModel model=(DefaultTableModel)table.getModel();
        model.setColumnIdentifiers(columnsName);

        Object[] tableLines =br.lines().toArray();
        for(int i=0;i < tableLines.length;i++) {
            String line=tableLines[i].toString().trim();
            String[] dataRow = line.split(" / ");
            model.addRow(dataRow);
        }
        br.close();

    } catch (FileNotFoundException e3) {
        // TODO Auto-generated catch block
        e3.printStackTrace();
    } catch (IOException e1) {
        // TODO Auto-generated catch block
        e1.printStackTrace();
    }
}
```

## 5.6 RESULT VERIFICATION FOR MULTITHREADING

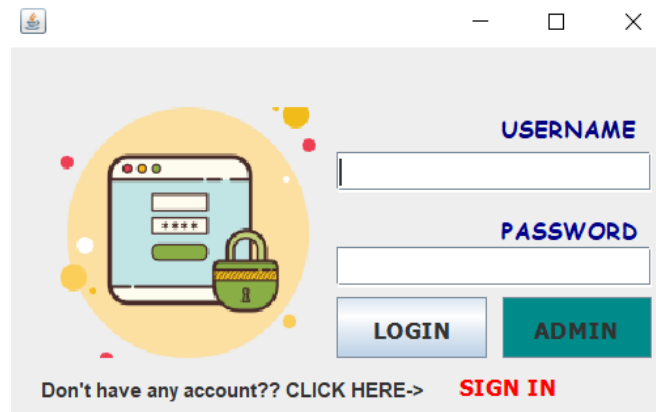
```
String filepath="C:\\Users\\Stuti\\Desktop\\output.txt";
File file=new File(filepath);
try {
    BufferedReader br=new BufferedReader(new FileReader(file));
    String firstLine=br.readLine().trim();
    String[] columnsName=firstLine.split(" , ");

    DefaultTableModel model=(DefaultTableModel)table.getModel();
    model.setColumnIdentifiers(columnsName);
    for(int i=1;i<5;i++)
    {
        try
        {
            Thread.sleep(500);
        }catch(InterruptedException e2){}
    }

    Object[] tableLines =br.lines().toArray();
    for(int i=0;i < tableLines.length;i++) {
        String line=tableLines[i].toString().trim();
        String[] dataRow = line.split(" / ");
        model.addRow(dataRow);
        try {
            Thread.sleep(500);
        } catch (InterruptedException e1) {
            // TODO Auto-generated catch block
            e1.printStackTrace();
        }
    }
    br.close();

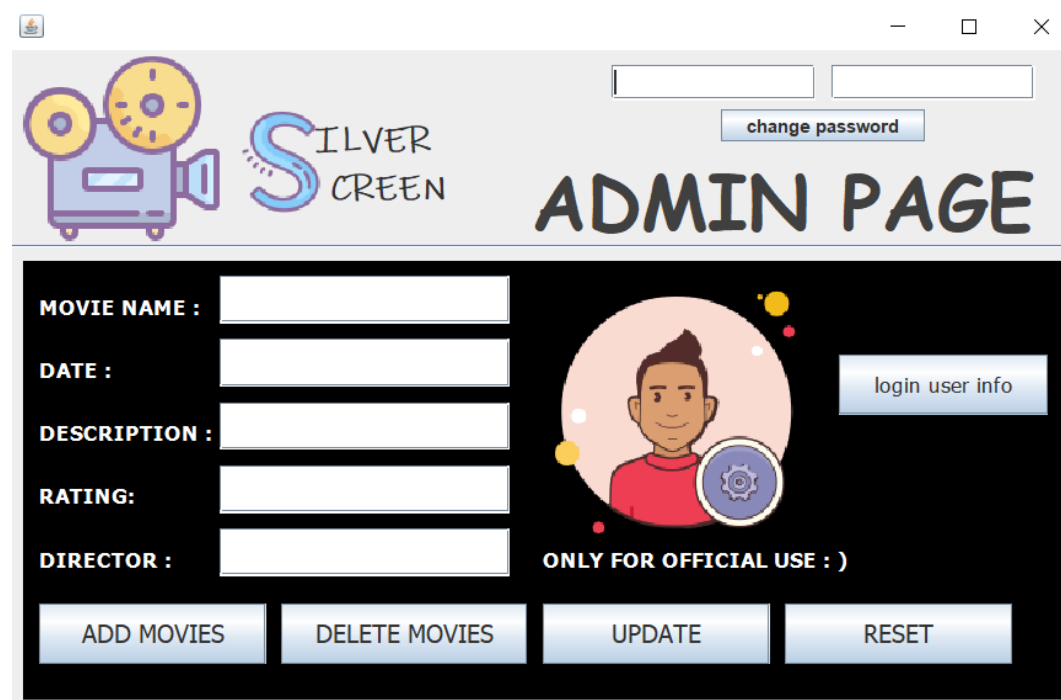
} catch (FileNotFoundException e3) {
    // TODO Auto-generated catch block
    e3.printStackTrace();
} catch (IOException e1) {
    // TODO Auto-generated catch block
    e1.printStackTrace();
}
```

## OUTPUTS:



A login window titled 'LOGIN CLASS'. It features a central illustration of a computer monitor with a lock icon and a green padlock, surrounded by yellow and red dots. To the right of the illustration are two input fields labeled 'USERNAME' and 'PASSWORD'. Below these fields are two buttons: 'LOGIN' (light blue) and 'ADMIN' (teal). At the bottom, there is a link that says 'Don't have any account?? CLICK HERE-> SIGN IN' in red text.

Figure 14: FRAME FOR LOGIN CLASS



An admin page window titled 'ADMIN PAGE'. The header includes a logo of a movie camera and the text 'SILVER SCREEN'. On the right, there are two input fields and a 'change password' button. The main content area has a dark background and contains a list of labels for movie information: 'MOVIE NAME :', 'DATE :', 'DESCRIPTION :', 'RATING:', and 'DIRECTOR :', each followed by an input field. To the right of these fields is a circular profile picture of a man with a gear icon, and a 'login user info' button. Below the profile picture, it says 'ONLY FOR OFFICIAL USE : )'. At the bottom, there are four buttons: 'ADD MOVIES', 'DELETE MOVIES', 'UPDATE', and 'RESET'.

Figure 15: FRAME FOR ADMIN CLASS



BOOKING DETAILS OF USER

click here and wait for the result to be processed -->

**PRESS HERE TO GET INFO**

accountno	name	number	movie	time	date	seat type	price	quantity	total
34567	sneha	3456789	SOORYAVAN...	1:30	27 April 2020	SUPERBALC...	300	2	600

Figure 16: FRAME FOR INFO CLASS

 **SIGNIN**  
enter details carefully

first name: \*

last name: \*

email: \*

username: \*

password: \*

re-password: \*

phoneno: \*

gender: \*

**REGISTER** **RESET**

Figure 17: FRAME FOR SIGNIN CLASS

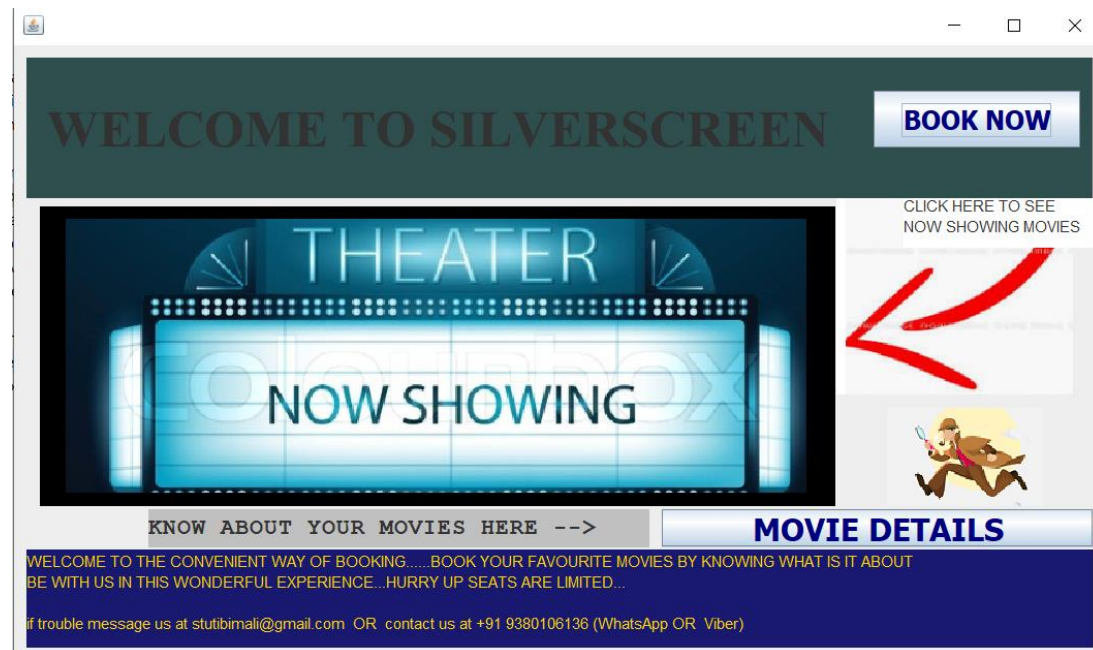


Figure 18: FRAME FOR BOOKING CLASS

DETAILS OF MOVIES

click here and wait for sometime to process the data ->

PRESS HERE FOR DETAILS

movie	release	director	rating	description
ONWARD	6th Mar 2020	Dan Scanlon	4	To celebrate turning 16, Ian Lightf.
SOUL	19th Jun 2020	Pete Docter	3	Just when Joe thinks his dream ...
Minions: The Rise of Gru	10th Jul 2020	Kyle Balda	5	This summer, from the biggest a.
TENET	17th Jul 2020	Christopher Nolan	3	He died instead of giving up his co
ANGREGI MEDIUM	March 13th 2020	Homi Adajania	4	Champak, an unassuming mithai
BLOODSHOT	March 13th 2020	Dave Wilson	4	After he and his wife are murder.
MULAN	March 27th 2020	Nikola Jean Caro	3	To save her ailing father from ser.
SOORYAVANSHI	March 24, 2020	Rohit Shetty	3	Sooryavanshi is an upcoming Indi

Figure 19: FRAME FOR DETAIL CLASS

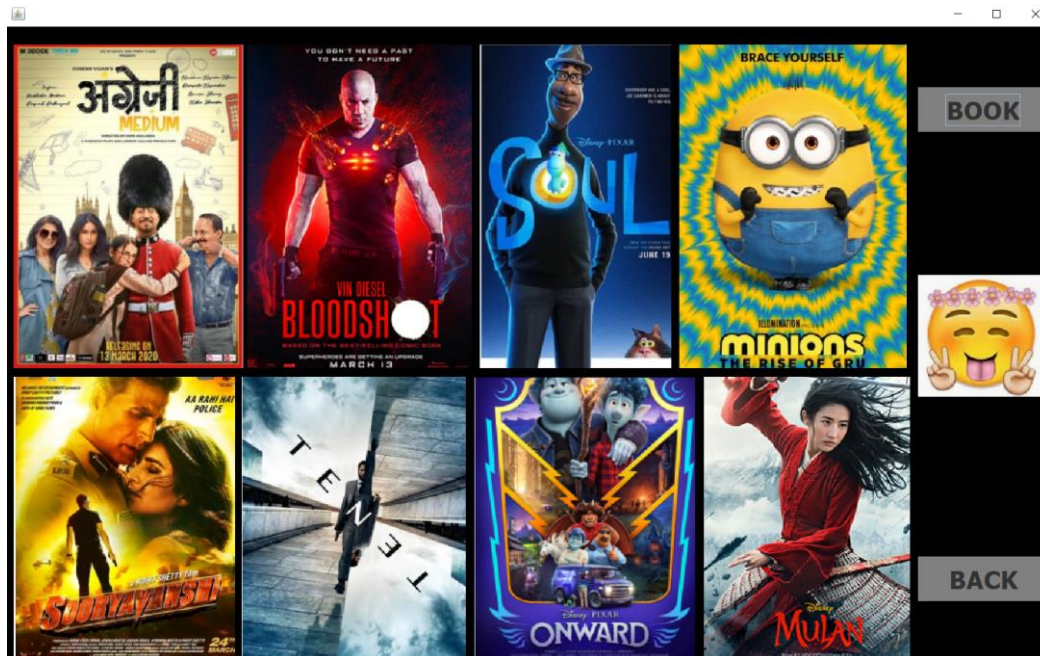


Figure 20: FRAME FOR MOVIE CLASS

TICKET BOOKING

RESET

NAME \*
ACCOUNT NO \*
ADD
MONEY DETAILS

PHONE NO \*

☐ ODC
☐ BALCONY
☐ BOX
☐ SUPERBALCONY

MOVIE \*
TIME \*
DATE \*

SUB TOTAL
PAY \*
BALANCE

NOTE : Enter the account no carefully, it will be matched with the transaction account no.  
The amount can be payed using PayPal, UPI , Google Pay or Paytm at the number +919380106136

BOOK
BACK

Figure 21: FRAME FOR TICKET CLASS

## DATABASE OUTPUTS

	firstname	lastname	email	username	password	phoneno	gender
▶	aaaradhya	bimali	aabimali@gmail.com	aaru	123456	9876533325	male
	nagendra	sharma	nbimali12@gmail.com	nbi	1475369	9851085495	male
	priyam	ss	abpri@hotmail.com	piyaa	123345	9876729222	male
	queen	kk	njhas@yahoo.com	quuo	pro123	9876576545	female
	rahul	owosti	rahul123@gmail.com	rahul	popmusic	9876543210	male
	sneha	bimali	snehabimali@gmail.com	sneha	staysafe123	9380106136	female
	sparsha	pant	sparsha12i@gmail.com	spar	1234	9874930200	female
	usha	rijal	usha@gmail.com	usha	12re5	9876617899	female
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 2: SIGNUP TABLE

	movie_name	release_date	director	rate	description
▶	ANGREZI MEDIUM	March 13th 2020	Homi Adajania	3	Champak, an unassuming mithai shop owner and single father is set to fulfil his daughter's dream to study in London...
	BLOODSHOT	March 13th 2020	Dave Wilson	3	After he and his wife are murdered, marine Ray Garrison is resurrected by a team of scientists. Enhanced with nan...
	MINIONS:THE RISE OF GRU	10th Jul 2020	Kyle Balda	4	This summer, from the biggest animated franchise in history and global cultural phenomenon, comes the untold stor...
	MULAN	March 27th 2020	Nikola Jean Caro	3	To save her ailing father from serving in the Imperial Army, a fearless young woman disguises herself as a man to b...
	ONWARD	6th Mar 2020	Dan Scanlon	4	To celebrate turning 16, Ian Lightfoot and his brother, Barley, get a magical gift: a spell that'll bring back their dece...
	SOORYAVANSHI	March 24, 2020	Rohit Shetty	4	Sooryavanshi is an upcoming Indian Hindi-language action film directed by Rohit Shetty and written by Yunus Sajaw...
	SOUL	19th Jun 2020	Pete Docter	3	Just when Joe thinks his dream might be in reach, a single unexpected step sends him to a fantastical place where ...
	TENET	17th Jul 2020	Christopher Nolan	3	He died instead of giving up his colleagues. He passed the test. Now our hero's needed to help prevent a dangerou...
*	NULL	NULL	NULL	NULL	NULL

Figure 3: MOVIE DETAIL TABLE

	id	accountno	name	phoneno	movie	time	date	ticketid	tickettype	quantity	price	total
▶	1	1234567890	stuti	9813781888	MULAN	1:30	18 April 2020	7	BALCONY	1	400	400
	3	87654	wkld	987654	ANGREZI MEDIUM	11:30	18 April 2020	41	BOX	2	450	900
	4	34567	sneha	3456789	SOORYAVANSHI	1:30	27 April 2020	42	SUPERBALCONY	2	300	600
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 4: BOOKING TABLE

	id	nameo	phoneno	subtotal	pay	balance	accountno
▶	42	sneha	3456789	600	500	-100	34567
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Figure 5: TRANSACTION TABLE

## Chapter 6

## **CONCLUSION**

This project helps to book movie ticket, view available movies, view available tickets for costumers and for admin helps to add/ remove movies, increase/ decrease price, customize available seats. To access this project first we need to login or if new customers signup. All the data entered are stored in database. And data are extracted from database as well. This project makes convenient for employs of hall as well as the customers.

## **Chapter 7**

## **REFERENCES**

For my project, following references were used for better understanding:

- 1] <https://stackify.com/oops-concepts-in-java/>
- 2] [www.youtube.com](http://www.youtube.com)
- 3] [www.slideshare.net](http://www.slideshare.net)
- 4] [www.javatpoint.com](http://www.javatpoint.com)
- 5] Herbert Schildt – Java the Complete Reference -McGraw-Hill Education
- 6] <https://www.geeksforgeeks.org/>