THEATRE TICKET BOOKING

A Project Report submitted to
Bishop Heber College (Autonomous), Tiruchirappalli
affiliated to Bharathidasan University, Tiruchirappalli – 620024
in partial fulfilment of the requirements for the award of the degree of

Bachelor of Vocation in Information Technology

By
M. SANKARANARAYANAN
(Register No.: 215915143)

Under the guidance of
Dr. M. UMA MAHESWARI, MCA, M.Phil., Ph.D.

Department of Information Technology
Bishop Heber College (Autonomous)
(Nationally Re-accredited with 'A' Grade by NAAC with a CGPA of 3.58 out of 4)
(Recognized by UGC as "College of Excellence")
Tiruchirappalli – 620 017

APRIL 2023

Department of Information Technology,
Bishop Heber College (Autonomous)
Tiruchirappalli – 620 017, Tamil Nadu, India
Phone No.: 0431 – 277 0136

CERTIFICATE

This Viva-Voce examination for the candidate **M. SANKARANARAYANAN**

(Reg. No. : 215915143) was held on _____

Signature of the HOD

Examiners:

1.

2.

Dr. M. UMA MAHESWARI, MCA, M.Phil., Ph.D.
Assistant Professor,
Department of Information Technology,
Bishop Heber College (Autonomous),
Tiruchirappalli – 620 017.

Date:

CERTIFICATE

This is to certify that the project work entitled **"THEATRE TICKET BOOKING"** is a Bonafide work done under my supervision by **M.SANKARANARAYANAN (Reg. No. 215915143)** and submitted to Bishop Heber College (Autonomous), Tiruchirappalli – 620 017 in partial fulfilment of the requirements for the award of the degree of Bachelor of Vocation in Information Technology during the even semester of the academic year (2022 – 2023).

Signature of the Guide

DECLARATION

I hereby declare that the work presented in this project work report is the original work done by me under the guidance of **Dr. M. UMA MAHESWARI, MCA, M.Phil., Ph.D.** Assistant Professor, Department of Information Technology, Bishop Heber College (Autonomous), Tiruchirapalli-620 017, and has not been included in any other project work submitted for any other degree.

Name of the Candidate	: M. SANKARANARAYANAN
Register Number	215915143
Semester	: FOUR
Academic Year	: 2022 – 2023
Course Code	: U21ITPJ4

Signature of the Candidate

ACKNOWLEDGEMENTS

First of all, I would thank **ALMIGHTY GOD** for granting abundant grace, good health and knowledge to do this Project.

I express my sincere gratitude to D**r. D. PAUL DHAYABARAN, M.Sc., M.Phil., PGDCA., Ph.D.,** Principal, Bishop Heber College (Autonomous), Tiruchirappalli for his blessings.

I am highly indebted to D**r. J. JOHN RAYBIN JOSE, M. Sc., MCA, M. Phil., PGDCA, Ph. D., SET,** Associate Professor and Head, Department of Information Technology, Bishop Heber College (Autonomous), Trichy for providing his support and guidance during this project work.

I wish to place on record my gratitude to **Dr. M. UMA MAHESWARI, MCA, M.Phil., Ph.D.** Assistant Professor, Department of Information Technology, Bishop Heber College (Autonomous), Trichy for her support during this project work.

I record my gratitude to all the staff members of the Department of Information Technology for their contribution in my academic endeavors and developments.

I record my deep sense of gratitude to my beloved parents and my friends for their encouragement and moral support extended during the period of my project.

M. SANKARANARAYANAN

ABSTRACT

The primary purpose of making the theatre ticket booking system is to create an automatic online based system which will provide an easy and an alternate way to book a ticket for a movie. The customer will need to enter the information related to film after which the system will provide options to book the ticket. In this Online cinema ticket booking system, the staff of the cinema hall won't have to do anything for the ticket booking as the process will be done by the customer through the web which will reduce the number of staff required in the booking counter.

CONTENTS

1.INTRODUCTION

"Theatre Ticket Booking" is a web-based application developed using Android Studio. This project is developed for ticket booking of movie online of any theater of Trichy. Now a days, use of internet is used everywhere so it's make easy for user to book the tickets of movies online it saves the time of standing in the queue to book the tickets.

This project has mainly three functionalities it not only books the tickets but it also provides some information to users which can be helpful to them like list of theaters in Trichy, list of phone numbers of these theaters, photo gallery many all these facilities are provided here. And the most important part is it is easy to use for users they can directly see the useful information.

2.SYSTEM STUDY

A Theatre Booking System is a project that allows users to check for available seats in the theatre and can book them at the same time. It happens when people go for a movie and they return in disappointment due to houseful. Hence the project is intended to overcome this situation by automating the manual booking system, online. Users can make prior bookings for desired seats for their desired show at desired time and cost. The system provides additional optional things like food packages, discounts etc. It also calculates all the associated charges incurred in booking the ticket and the items. This is an advanced booking system that makes user customize their show according to their needs.

2.1. PROJECT DESCRIPTION

An online theatre ticket booking system involves interaction between the representatives of a business and the customer. Both parties have access to the system but they can use it in different ways i.e. they each have different roles to play. The business representatives access the system as administrators (admins) while the customers access the system as registered users or visitors (for those who are not registered).

2.1.1. Existing System

Today, there are many web-based systems are available which provides the online booking of a movie ticket, but the refund is not an easy process while in the present system once someone has chosen the seats they can't change it. The user who has booked the ticket needs to go to the booking counter because there is no electronic ticket system, so the customers have to wait in the queue

2.1.2. Proposed System

The user can search for the movies which will be going to release in the future, so they will have an option to book the ticket in advance. The Theatre Booking System will make the user experience much better than the present system. The Online Theatre Booking System will provide a much better experience for booking of movie tickets. If seats are available, the customer can change the position of the seats.

MODULE DESCRIPTION

A module is a software component or part of a program that contains one or more routines. One or more independently developed modules make up a program. An enterprise-level software application may contain several different modules, and each module serves unique and separate business operations. Modules make a programmer's job easy by allowing the programmer to focus on only one area of the functionality of the software application. Modules are typically incorporated into the program (software) through interfaces.

MODULES

The THEATRE TICKET BOOKING system applications contains four modules.

The details of these modules are as following the this are all menu modulus.

- **New user page**

  The new user page or Registration page deals the details of the user such as information like full name, mobile number, password and confirm password.

- **Login Page**

  The created User account can log-in using their Phone number and password to access the application.

- **Home**

  This module manage the movie list

- **Time and Screen**

  This module manage the movie time, date and screen list

- **Seat Selecting**

  This module manage the seat selecting

- **Booking Ticket**

  This module manage the booking ticket and payment

2.2. REQUIREMENT ANALYSIS

Requirements analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups of users, avoidance of feature creep and documentation of all aspects of the project development process from start to finish. Energy should be directed towards ensuring that the final system or product conforms to client needs rather than attempting to user expectations to fit the requirements. Requirements analysis is a team effort that demands a combination of hardware, software and human factors engineering expertise as well as skills in dealing with people.

2.2.1. HARDWARE REQUIREMENT

The hardware specification of the laptop computer system available for developing the project is given below

Processor	**: Intel Core I5**
Hard disk	**: 500GB**
RAM	**:4GB**
Keyboard	**: Standard QWERTY Keyboard**
Mouse	**: Standard mouse with 2 buttons**

2.2.2. SOFTWARE REQUIREMENTS

A software requirement specification (SRS) is a complete description of the behavior of the system to be developed. It includes a set of use cases that describe all the interaction the users will have with the software.

WINDOWS

An operating system is software that communicate with the hardware and allows other to run. It comprises the system software, and other utilities. The Operating system used as platform to develop this project in windows 8.1 Pro. It is a multi-user operating system.

ANDROID STUDIO

Android Studio is the official Integrated Development Environment (IDE) for android application development. Android Studio provides more features that enhance our productivity while building Android apps. Android Studio was announced on 16th May 2013 at the Google I/O conference as an official IDE for Android app development. It started its early access preview from version 0.1 in May 2013. The first stable built version was released in December 2014, starts from version 1.0. Since 7th May 2019, Kotlin is Google's preferred language for Android application development. Besides this, other programming languages are supported by Android Studio.

FIREBASE

Firebase is a Backend-as-a-Service (Baas). It provides developers with a variety of tools and services to help them develop quality apps, grow their user base, and earn profit. It is built on Google’s infrastructure. Firebase is categorized as a NoSQL database program, which stores data in JSON-like documents.

3.SYSTEM DESIGN

Systems design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

3.1. LOGICAL DESIGN

Logical design is an abstract concept in computer programming by which programmers arrange data in a series of logical relationships known as attributes or entities. An entity refers to a chunk of information, whereas an attribute defines the unique properties of an entity.

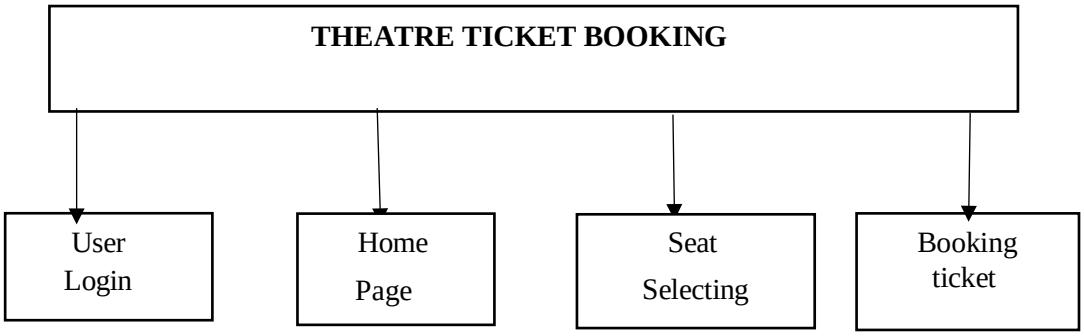


Fig 3.1. Logical Design for Theatre Ticket Booking Application

3.2 DATABASE DESIGN

Database design is the process of producing a detailed data model of database. This data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a data definition language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity. A good database design is important in ensuring consistent data, elimination of data redundancy, efficient execution of queries and high-performance application. Taking the time to design a database saves time and

frustration during development, and a well-designed database ensures ease of access and retrieval of information.

3.2.1 DATAFLOW DIAGRAM

Data Flow Diagram (DFD) provides a visual representation of the flow of information (i.e., data) within a system. By drawing a Data Flow Diagram, the information provided by and delivered to someone who takes part in system processes, the information needed in order to complete the processes and the information needed to be stored and accessed.

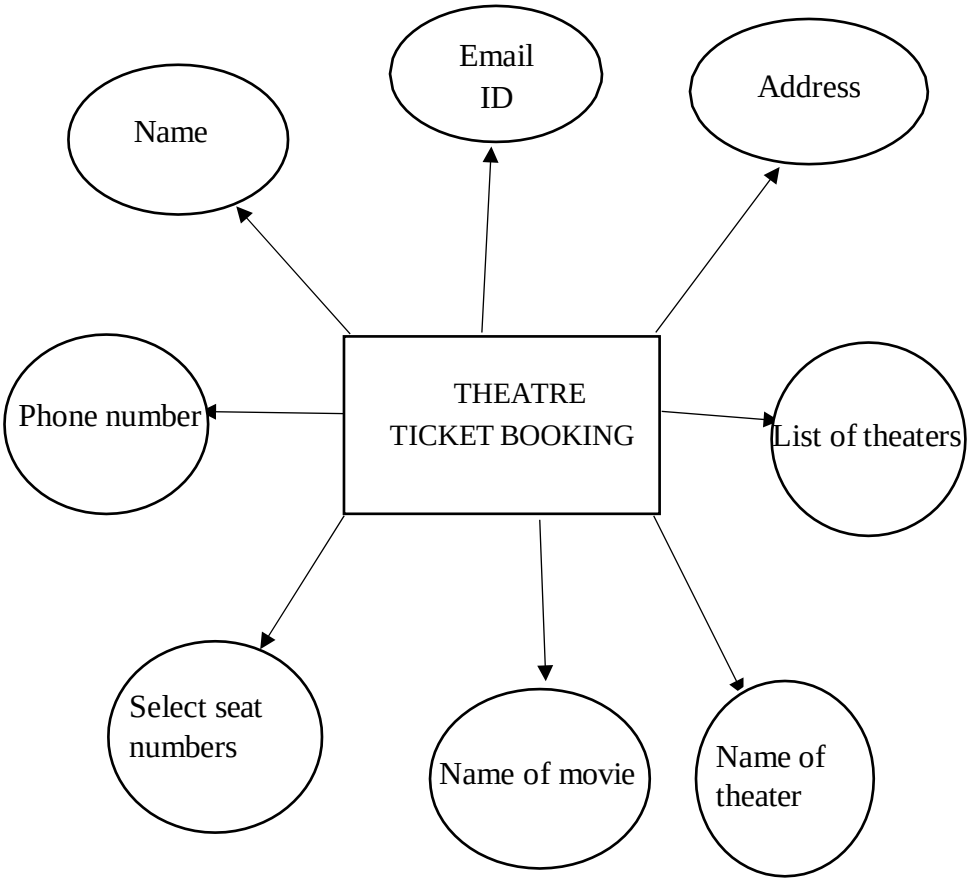
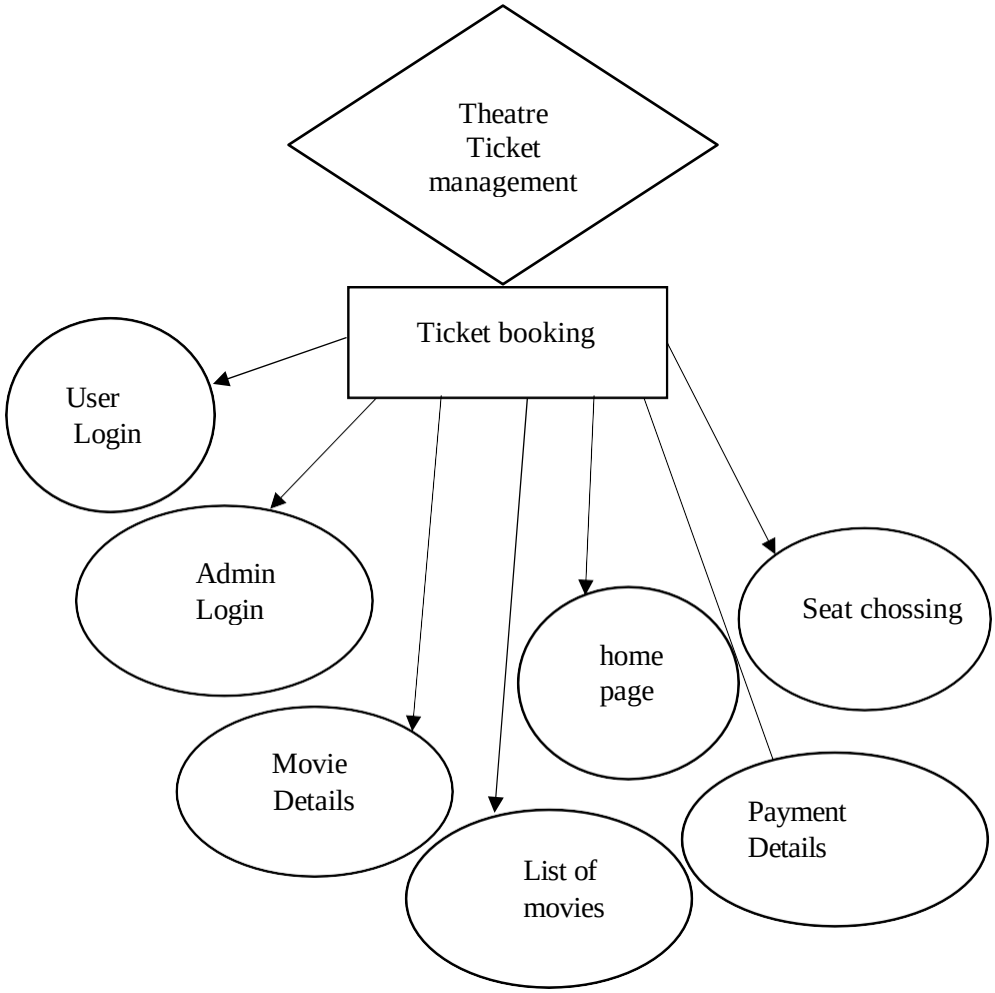


Fig 3.2 Data flow Diagram for "Theatre ticket booking System"

3.2.2 ENTITY RELATIONSHIP DIAGRAM

An entity-relationship diagram (ERD) is a data modelling technique that graphically illustrates an information system's entities and the relationships between those entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.



3.2 Entity Relationship Diagram

4.SYSTEM DEVELOPMENT

System development is the process of code construction. It includes in the internal development of customized systems, the creation of database systems, or the acquisition of third party developed software.

4.1. PROGRAM DEVELOPMENT

Android Manifest:

```
package com.flag.bookmyticket;

import android.content.Context;
import android.content.DialogInterface;
  import android.content.Intent;
import android.content.SharedPreferences;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

public class LoginActivity extends AppCompatActivity {

   private EditText email,password;
   String EmailHolder, PasswordHolder;
   Boolean EditTextEmptyHolder;
   SQLiteDatabase sqLiteDatabaseObj;
   SQLiteHelper sqLiteHelper;
   Cursor cursor;
   String TempPassword = "NOT_FOUND" ;
   public static final String UserEmail = "";
   SharedPreferences sharedpreferences;
   public static final String mypreference = "mypref";
   public static final String Email = "emailKey";
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        getSupportActionBar().setBackgroundDrawable(new
ColorDrawable(Color.parseColor("#d32f2f")));

        email =(EditText)findViewById(R.id.emailET);
        password =(EditText)findViewById(R.id.passwordET);

        sqLiteHelper = new SQLiteHelper(this);

        TextView register = (TextView)findViewById(R.id.register);
        Button loginbtn = (Button)findViewById(R.id.loginbtn);
        sharedpreferences = getSharedPreferences(mypreference,
Context.MODE_PRIVATE);

        register.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent in = new Intent(getApplicationContext(),RegisterActivity.class);
                startActivity(in);
                overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_left);
            }
        });

        loginbtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                CheckEditTextStatus();
                Login();
            }
        });

    }
    public void Login(){
        if(EditTextEmptyHolder) {
            sqLiteDatabaseObj = sqLiteHelper.getWritableDatabase();
            cursor = sqLiteDatabaseObj.query(SQLiteHelper.TABLE_NAME, null, " " +
SQLiteHelper.Table_Column_2_Email + "=?", new String[]{EmailHolder}, null,
null, null);
            while (cursor.moveToNext()) {
                if (cursor.isFirst()) {
                    cursor.moveToFirst();
                    TempPassword =
cursor.getString(cursor.getColumnIndex(SQLiteHelper.Table_Column_4_Password))
;
                    cursor.close();
                }
```

```
            }
            CheckFinalResult();
        }
        else {
            Toast.makeText(LoginActivity.this,"Please Enter UserName or
Password.",Toast.LENGTH_LONG).show();
        }
    }

    public void CheckEditTextStatus(){
        EmailHolder = email.getText().toString();
        PasswordHolder = password.getText().toString();

        if( TextUtils.isEmpty(EmailHolder) || TextUtils.isEmpty(PasswordHolder)){
            EditTextEmptyHolder = false ;
        }
        else {
            EditTextEmptyHolder = true ;
        }
    }

    public void CheckFinalResult(){
        if(TempPassword.equalsIgnoreCase(PasswordHolder))
        {
            Toast.makeText(LoginActivity.this,"Login
Successful",Toast.LENGTH_LONG).show();
            SharedPreferences.Editor editor = sharedpreferences.edit();
            editor.putString(Email, EmailHolder);
            editor.commit();
            Intent intent = new Intent(LoginActivity.this, HomeActivity.class);
            intent.putExtra(UserEmail, EmailHolder);
            startActivity(intent);
            overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_left);
        }
        else {
            Toast.makeText(LoginActivity.this,"UserName or Password is Wrong, Please
Try Again.",Toast.LENGTH_LONG).show();
        }
        TempPassword = "NOT_FOUND" ;
    }

    @Override
    public void onBackPressed() {
        new AlertDialog.Builder(this)
                .setIcon(android.R.drawable.ic_dialog_alert)
                .setTitle("Quit Application")
                .setMessage("Are you sure you want to Exit?")
                .setPositiveButton("Yes", new DialogInterface.OnClickListener()
                {
                    @Override
```

```
                public void onClick(DialogInterface dialog, int which) {
                    finishAffinity();
                }

            })
            .setNegativeButton("No", null)
            .show();
    }

    public void ADMINLOG(View view) {
        Intent intent = new Intent(this,AdminLoginActivity.class);
        startActivity(intent);
    }
}
package com.flag.bookmyticket;

import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.os.Bundle;
import android.text.Html;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;

public class RegisterActivity extends AppCompatActivity {

    private EditText name,email,mobile,password;
    SQLiteDatabase sqLiteDatabaseObj;
    String SQLiteDataBaseQueryHolder ;
    SQLiteHelper sqLiteHelper;
    Boolean EditTextEmptyHolder;
    String NameHolder, EmailHolder, MobileHolder, PasswordHolder;
    String F_Result = "Not_Found";
    Cursor cursor;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_register);
```

```
        getSupportActionBar().setBackgroundDrawable(new
ColorDrawable(Color.parseColor("#d32f2f")));


        TextView login = (TextView)findViewById(R.id.login);

         name = (EditText)findViewById(R.id.nameET);
         email = (EditText)findViewById(R.id.emailET);
         mobile = (EditText)findViewById(R.id.mobileET);
         password = (EditText)findViewById(R.id.passwordET);

        Button registerBtn = (Button)findViewById(R.id.registerbtn);
        sqLiteHelper = new SQLiteHelper(this);

        registerBtn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                SQLiteDataBaseBuild();
                SQLiteTableBuild();
                CheckEditTextStatus();
                CheckingEmailAlreadyExistsOrNot();
                EmptyEditTextAfterDataInsert();
            }
        });

        login.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View v) {
                Intent in = new Intent(getApplicationContext(),LoginActivity.class);
                startActivity(in);
                overridePendingTransition(R.anim.slide_in_left, R.anim.slide_out_right);
            }
        });
    }
    public void SQLiteDataBaseBuild(){
        sqLiteDatabaseObj =
openOrCreateDatabase(SQLiteHelper.DATABASE_NAME,
Context.MODE_PRIVATE, null);
    }
    public void SQLiteTableBuild() {
        sqLiteDatabaseObj.execSQL("CREATE TABLE IF NOT EXISTS " +
SQLiteHelper.TABLE_NAME + "(" + SQLiteHelper.Table_Column_ID + "
INTEGER PRIMARY KEY AUTOINCREMENT NOT NULL, " +
SQLiteHelper.Table_Column_1_Name + " VARCHAR, " +
SQLiteHelper.Table_Column_2_Email + " VARCHAR, " +
SQLiteHelper.Table_Column_3_Mobile + " VARCHAR, " +
SQLiteHelper.Table_Column_4_Password + " VARCHAR);");
    }

    public void InsertDataIntoSQLiteDatabase(){
```

```
        if(EditTextEmptyHolder == true)
        {
            SQLiteDataBaseQueryHolder = "INSERT INTO
"+SQLiteHelper.TABLE_NAME+" (name,email,mobile,password)
VALUES("'+NameHolder+"', "'+EmailHolder+"',
"'+MobileHolder+"',"'+PasswordHolder+"');";
            sqLiteDatabaseObj.execSQL(SQLiteDataBaseQueryHolder);
            sqLiteDatabaseObj.close();
            Toast.makeText(RegisterActivity.this,"User Registered Successfully",
Toast.LENGTH_LONG).show();
            Intent in = new Intent(getApplicationContext(),LoginActivity.class);
            startActivity(in);
            overridePendingTransition(R.anim.slide_in_left, R.anim.slide_out_right);
        }
        else {
            Toast.makeText(RegisterActivity.this,"Please Fill All The Required Fields.",
Toast.LENGTH_LONG).show();
        }

    }


    public void EmptyEditTextAfterDataInsert(){
        name.getText().clear();
        email.getText().clear();
        mobile.getText().clear();
        password.getText().clear();
    }

    public void CheckEditTextStatus(){
        NameHolder = name.getText().toString() ;
        EmailHolder = email.getText().toString();
        MobileHolder = mobile.getText().toString();
        PasswordHolder = password.getText().toString();

        if(TextUtils.isEmpty(NameHolder) || TextUtils.isEmpty(EmailHolder) ||
TextUtils.isEmpty(PasswordHolder)){
            EditTextEmptyHolder = false ;
        }
        else {
            EditTextEmptyHolder = true ;
        }
    }


    public void CheckingEmailAlreadyExistsOrNot(){
        sqLiteDatabaseObj = sqLiteHelper.getWritableDatabase();
        cursor = sqLiteDatabaseObj.query(SQLiteHelper.TABLE_NAME, null, " " +
SQLiteHelper.Table_Column_2_Email + "=?", new String[]{EmailHolder}, null,
```

```
null, null);
        while (cursor.moveToNext()) {
            if (cursor.isFirst()) {
                cursor.moveToFirst();
                F_Result = "Email Found";
                cursor.close();
            }
        }
        CheckFinalResult();
    }

    public void CheckFinalResult(){

        if(F_Result.equalsIgnoreCase("Email Found"))
        {
            Toast.makeText(RegisterActivity.this,"Email Already
Exists",Toast.LENGTH_LONG).show();
        }
        else {
            InsertDataIntoSQLiteDatabase();
        }
        F_Result = "Not_Found" ;
    }

    @Override
    public void onBackPressed() {
        new AlertDialog.Builder(this)
                .setIcon(android.R.drawable.ic_dialog_alert)
                .setTitle("Quit Application")
                .setMessage("Are you sure you want to Exit?")
                .setPositiveButton("Yes", new DialogInterface.OnClickListener()
                {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        finishAffinity();
                    }

                })
                .setNegativeButton("No", null)
                .show();
    }
}
package com.flag.bookmyticket;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
```

```
import android.os.Bundle;
import android.view.MenuItem;
import android.widget.Toast;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import com.google.android.material.bottomnavigation.BottomNavigationView;

import java.util.ArrayList;
import java.util.List;

public class HomeActivity extends AppCompatActivity {

    private RecyclerView recyclerView;
    private RecyclerViewAdapter recyclerViewAdapter;
    private List<Movie> movieList;
    public static String imageIntent = "imageIntent";
    public static  String titleIntent ="titleIntent";
    public static  String ratingIntent ="ratingIntent";
    public static  String runtimeIntent ="runtimeIntent";
    public static  String genreIntent ="genreIntent";

    SharedPreferences sharedpreferences;
    public static final String mypreference = "mypref";
    public static final String Email = "emailKey";


    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_home);
        getSupportActionBar().setBackgroundDrawable(new
ColorDrawable(Color.parseColor("#d32f2f")));



        BottomNavigationView navView = findViewById(R.id.nav_view);
        navView.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {
            @Override
            public boolean onNavigationItemSelected(@NonNull MenuItem item) {
                switch (item.getItemId()) {
                    case R.id.navigation_home:
                        break;
                    case R.id.navigation_dashboard:
                        Intent a = new Intent(HomeActivity.this,BookingsActivity.class);
```

```
                startActivity(a);
                overridePendingTransition(R.anim.slide_in_right,
R.anim.slide_out_left);
                break;
            case R.id.navigation_notifications:
                Intent b = new Intent(HomeActivity.this,ProfileActivity.class);
                startActivity(b);
                overridePendingTransition(R.anim.slide_in_right,
R.anim.slide_out_left);
                break;
        }
        return false;
      }
    });

    sharedpreferences = getSharedPreferences(mypreference,
Context.MODE_PRIVATE);
    Toast.makeText(getApplicationContext(),sharedpreferences.getString(Email,
""),Toast.LENGTH_LONG).show();

    movieList = new ArrayList<>();
    prepareMovie();
    recyclerView = (RecyclerView)findViewById(R.id.recyclerView);
    recyclerViewAdapter = new RecyclerViewAdapter(movieList);
    RecyclerView.LayoutManager layoutManager = new
LinearLayoutManager(this);
    recyclerView.setLayoutManager(layoutManager);



    recyclerViewAdapter.setOnItemClickListener(new ClickListener<Movie>(){
      @Override
      public void onItemClick(Movie data) {
        Intent intent = new Intent(HomeActivity.this, ScreenSelection.class);
        Bundle extras = new Bundle();
        extras.putInt(imageIntent, data.getImage());
        extras.putString(titleIntent,data.getTitle());
        extras.putString(ratingIntent,data.getRating());
        extras.putString(runtimeIntent,data.getRuntime());
        extras.putString(genreIntent,data.getGenre());
        intent.putExtras(extras);
        startActivity(intent);
        overridePendingTransition(R.anim.slide_in_right, R.anim.slide_out_left);
      }
    });

    recyclerView.setAdapter(recyclerViewAdapter);
  }
  private void prepareMovie(){
    Movie movie = new Movie("Laththi ","U/A","Action,  Adventure","2h
```

```
3m","",R.drawable.one);
        movieList.add(movie);
        movie = new Movie("LOVE TODAY ","A","Romantic, Comedy","2h
34m","",R.drawable.two);
        movieList.add(movie);
        movie =new Movie("Connect (2021)","U/A","Horror, Supernatural","1h
39m","",R.drawable.three);
        movieList.add(movie);
        movie =new Movie("Gatta
Kusththi ","U","Romantic, Comedy, Sports, Family","3h 47m","",R.drawable.four);
        movieList.add(movie);
        movie =new Movie("Raangi ","U","Thriler, Action, Adventure","2h
1m","",R.drawable.five);
        movieList.add(movie);
        movie =new Movie("SPIDER-MAN (INTO THE SPIDER VERSE) ","PG-
13","Family, Action","1h 40m","",R.drawable.six);
        movieList.add(movie);
        movie =new Movie("Operation Fortune: Ruse de
Guerre ","A","Comedy, Family, Animation","1h 54m","",R.drawable.seven);
        movieList.add(movie);
        movie =new Movie("Avatar: The Way of Water ","U/A","Sci-Fi, Action","3h
2m","",R.drawable.eight);
        movieList.add(movie);
        movie =new Movie("Creed[III] ","A"," Drama, Sport","1h
54m","",R.drawable.nine);
        movieList.add(movie);
        movie =new Movie("Black Adam ","PG-13","Fantasy, Action, Adventure","2h
5m","",R.drawable.ten);
        movieList.add(movie);
    }
    @Override
    public void onBackPressed() {
        new AlertDialog.Builder(this)
                .setIcon(android.R.drawable.ic_dialog_alert)
                .setTitle("Quit Application")
                .setMessage("Are you sure you want to Exit?")
                .setPositiveButton("Yes", new DialogInterface.OnClickListener()
                {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        finishAffinity();
                    }

                })
                .setNegativeButton("No", null)
                .show();
    }
} package com.flag.bookmyticket;

import androidx.annotation.NonNull;
```

```
import androidx.appcompat.app.AlertDialog;
import androidx.appcompat.app.AppCompatActivity;
import androidx.recyclerview.widget.DefaultItemAnimator;
import androidx.recyclerview.widget.LinearLayoutManager;
import androidx.recyclerview.widget.RecyclerView;

import android.content.Context;
import android.content.DialogInterface;
import android.content.Intent;
import android.content.SharedPreferences;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.graphics.Color;
import android.graphics.drawable.ColorDrawable;
import android.os.Bundle;
import android.view.MenuItem;

import com.google.android.material.bottomnavigation.BottomNavigationView;

import java.util.ArrayList;

public class BookingsActivity extends AppCompatActivity {

    ArrayList<Bookings> arrayList;
    RecyclerView recyclerView;
    SQLiteHelper database_helper;
    SQLiteDatabase sqLiteDatabaseObj;

    SharedPreferences sharedpreferences;
    public static final String mypreference = "mypref";
    public static final String Email = "emailKey";

    String currentUser="";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_bookings);
        getSupportActionBar().setBackgroundDrawable(new
ColorDrawable(Color.parseColor("#d32f2f")));


        recyclerView = (RecyclerView) findViewById(R.id.recycler_view);
        database_helper = new SQLiteHelper(this);
        displayBookings();

        BottomNavigationView navView = findViewById(R.id.nav_view);
        navView.setOnNavigationItemSelectedListener(new
BottomNavigationView.OnNavigationItemSelectedListener() {
            @Override
```

```
        public boolean onNavigationItemSelected(@NonNull MenuItem item) {
            switch (item.getItemId()) {
                case R.id.navigation_home:
                    Intent a = new Intent(BookingsActivity.this,HomeActivity.class);
                    startActivity(a);
                    overridePendingTransition(R.anim.slide_in_left,
R.anim.slide_out_right);
                    break;
                case R.id.navigation_dashboard:
                    break;
                case R.id.navigation_notifications:
                    Intent b = new Intent(BookingsActivity.this,ProfileActivity.class);
                    startActivity(b);
                    overridePendingTransition(R.anim.slide_in_right,
R.anim.slide_out_left);
                    break;
            }
            return false;
        }
    });
  }

  public void displayBookings() {
    arrayList = new ArrayList<>(getBookings());
    recyclerView.setLayoutManager(new LinearLayoutManager(this,
LinearLayoutManager.VERTICAL, false));
    recyclerView.setItemAnimator(new DefaultItemAnimator());
    BookingsAdapter adapter = new BookingsAdapter(getApplicationContext(), this,
arrayList);
    recyclerView.setAdapter(adapter);
  }

  public ArrayList<Bookings> getBookings() {
    sharedpreferences = getSharedPreferences(mypreference,
Context.MODE_PRIVATE);
    currentUser = sharedpreferences.getString(Email, "");
    String TABLE_NAME = "SeatsTable";
    ArrayList<Bookings> arrayList = new ArrayList<>();

    // select all query
    String group_by_clause="GROUP by
title_name,screen_name,show_name,show_date,seats_booked_user";
    String select_query= "SELECT
title_name,screen_name,show_name,show_date,seats_booked_user,group_concat(sea
t_name) FROM  SeatsTable where seats_booked_user = ''' + "" +currentUser+ "" + '''
"+group_by_clause;

    sqLiteDatabaseObj = database_helper.getWritableDatabase();
    Cursor cursor = sqLiteDatabaseObj.rawQuery(select_query, null);
```

```
        // looping through all rows and adding to list
        if (cursor.moveToFirst()) {
            do {
                Bookings bookings = new Bookings();
                bookings.setTitle(cursor.getString(0));
                bookings.setScreen(cursor.getString(1));
                bookings.setShow(cursor.getString(2));
                bookings.setDate(cursor.getString(3));
                bookings.setSeats(cursor.getString(5));
                arrayList.add(bookings);
            }while (cursor.moveToNext());
        }
        return arrayList;
    }

    @Override
    public void onBackPressed() {
        new AlertDialog.Builder(this)
                .setIcon(android.R.drawable.ic_dialog_alert)
                .setTitle("Quit Application")
                .setMessage("Are you sure you want to Exit?")
                .setPositiveButton("Yes", new DialogInterface.OnClickListener()
                {
                    @Override
                    public void onClick(DialogInterface dialog, int which) {
                        finishAffinity();
                    }

                })
                .setNegativeButton("No", null)
                .show();
    }
}
```

XML CODE

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    tools:context=".LoginActivity"
```

```
android:background="@drawable/background">

<RelativeLayout
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:layout_centerVertical="true"
   android:layout_centerHorizontal="true"
   android:orientation="vertical">

   <TextView
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
      android:id="@+id/title"
      android:text="LOGIN"
      android:textSize="32sp"
      android:gravity="center"
      android:textStyle="bold"
      android:textColor="@color/colorPrimary"/>

   <EditText
      android:id="@+id/emailET"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:layout_below="@+id/title"
      android:layout_marginStart="20dp"
      android:layout_marginTop="20dp"
      android:layout_marginEnd="20dp"
      android:layout_marginBottom="20dp"
      android:background="@drawable/rounded_edittext"
      android:hint="Email"
      android:inputType="textEmailAddress"
      android:padding="15dp"
      android:textColor="@color/colorPrimary"
      android:textColorHint="#9e9e9e" />

   <EditText
      android:id="@+id/passwordET"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:layout_below="@+id/emailET"
      android:layout_marginStart="20dp"
      android:layout_marginTop="20dp"
      android:layout_marginEnd="20dp"
      android:layout_marginBottom="20dp"
      android:background="@drawable/rounded_edittext"
      android:hint="Password"
      android:inputType="textPassword"
      android:padding="15dp"
      android:textColor="@color/colorPrimary"
      android:textColorHint="#9e9e9e" />

   <Button
      android:id="@+id/loginbtn"
      android:layout_width="match_parent"
      android:layout_height="wrap_content"
```

```
        android:layout_below="@+id/passwordET"
        android:layout_marginStart="20dp"
        android:layout_marginTop="20dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="20dp"
        android:background="@drawable/rounded_edittext"
        android:text="Login"
        android:textColor="@color/colorPrimary"
        android:textSize="18sp"
        android:textStyle="bold" />

    <TextView
        android:id="@+id/register"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@id/loginbtn"
        android:layout_marginTop="10dp"
        android:clickable="true"
        android:gravity="center"
        android:text="New user? Register here"
        android:textColor="@color/colorPrimary"
        android:textSize="20sp"
        android:textStyle="bold"
        tools:ignore="TouchTargetSizeCheck" />

    <Button
        android:id="@+id/admin"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/register"
        android:layout_marginStart="20dp"
        android:layout_marginTop="100dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="20dp"
        android:background="@drawable/rounded_edittext"
        android:text="ADMIN LOGIN"
        android:textColor="@color/colorPrimary"
        android:textSize="18sp"
        android:textStyle="bold"
        android:onClick="ADMINLOG"/>

  </RelativeLayout>
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>
  <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
  xmlns:tools="http://schemas.android.com/tools"
  android:layout_width="match_parent"
  android:layout_height="match_parent"
  tools:context=".RegisterActivity"
  android:background="@drawable/wellcome">

  <RelativeLayout
    android:layout_width="match_parent"
```

```
android:layout_height="wrap_content"
android:layout_centerVertical="true"
android:layout_centerHorizontal="true"
android:orientation="vertical">

<TextView
   android:layout_width="match_parent"
   android:layout_height="wrap_content"
   android:id="@+id/title"
   android:text="SIGN UP"
   android:textSize="32sp"
   android:gravity="center"
   android:textColor="#d32f2f"
   android:textStyle="bold"/>

<EditText
   android:id="@+id/nameET"
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:layout_below="@+id/title"
   android:layout_marginStart="20dp"
   android:layout_marginTop="20dp"
   android:layout_marginEnd="20dp"
   android:layout_marginBottom="20dp"
   android:background="@drawable/rounded_edittext"
   android:hint="Name"
   android:inputType="text"
   android:padding="15dp"
   android:textColor="#000"
   android:textColorHint="#9e9e9e" />

<EditText
   android:layout_margin="20dp"
   android:layout_below="@+id/nameET"
   android:inputType="textEmailAddress"
   android:id="@+id/emailET"
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:padding="15dp"
   android:textColor="#000"
   android:hint="Email"
   android:textColorHint="#9e9e9e"
   android:background="@drawable/rounded_edittext" />


<EditText
   android:id="@+id/mobileET"
   android:layout_width="fill_parent"
   android:layout_height="wrap_content"
   android:layout_below="@+id/emailET"
   android:layout_marginStart="20dp"
   android:layout_marginTop="20dp"
   android:layout_marginEnd="20dp"
   android:layout_marginBottom="20dp"
   android:background="@drawable/rounded_edittext"
```

```
        android:hint="Mobile"
        android:inputType="phone"
        android:padding="15dp"
        android:textColor="#000"
        android:textColorHint="#9e9e9e" />

    <EditText
        android:id="@+id/passwordET"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/mobileET"
        android:layout_marginStart="20dp"
        android:layout_marginTop="20dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="20dp"
        android:background="@drawable/rounded_edittext"
        android:hint="Password"
        android:inputType="textPassword"
        android:padding="15dp"
        android:textColor="#000"
        android:textColorHint="#9e9e9e" />

    <Button
        android:id="@+id/registerbtn"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/passwordET"
        android:layout_marginStart="20dp"
        android:layout_marginTop="20dp"
        android:layout_marginEnd="20dp"
        android:layout_marginBottom="20dp"
        android:background="#d32f2f"
        android:text="Register"
        android:textColor="#fff"
        android:textStyle="bold" />

    <TextView
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="10dp"
        android:id="@+id/login"
        android:layout_below="@id/registerbtn"
        android:text="Existing user? Login here"
        android:gravity="center"
        android:textColor="@color/purple_200"
        android:textSize="20sp"
        android:textStyle="bold"/>

  </RelativeLayout>
</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
  xmlns:app="http://schemas.android.com/apk/res-auto"
```

```
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/container"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="@drawable/moviemain"
    android:layout_marginTop="5dp">


    <ScrollView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:orientation="vertical">




    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MainActivity"
        android:layout_marginBottom="75dp"
        android:background="#fff">

        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/recyclerView"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            android:background="@android:color/white"
            tools:listitem="@layout/recyclerview_adapter_layout" />
    </LinearLayout>
</ScrollView>
    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/nav_view"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="0dp"
        android:layout_marginEnd="0dp"
        android:background="@color/colorPrimary"
        app:itemIconTint="@drawable/tab_color"
        app:itemTextColor="@drawable/tab_color"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:menu="@menu/bottom_nav_menu" />



</androidx.constraintlayout.widget.ConstraintLayout>

<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    android:id="@+id/container"
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
    android:background="#ffffff"
    android:layout_marginTop="5dp">

    <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        tools:context=".MainActivity"
        android:layout_marginBottom="75dp"
        android:background="#fff">
        <androidx.recyclerview.widget.RecyclerView
            android:id="@+id/recycler_view"
            android:background="@android:color/white"
            android:layout_width="match_parent"
            android:layout_height="match_parent"
            tools:listitem="@layout/bookings_adapter_layout"/>
    </LinearLayout>

    <com.google.android.material.bottomnavigation.BottomNavigationView
        android:id="@+id/nav_view"
        android:layout_width="0dp"
        android:layout_height="wrap_content"
        android:layout_marginStart="0dp"
        android:layout_marginEnd="0dp"
        android:background="@color/colorPrimary"
        app:itemIconTint="@drawable/tab_color"
        app:itemTextColor="@drawable/tab_color"
        app:layout_constraintBottom_toBottomOf="parent"
        app:layout_constraintLeft_toLeftOf="parent"
        app:layout_constraintRight_toRightOf="parent"
        app:menu="@menu/bottom_nav_menu" />



</androidx.constraintlayout.widget.ConstraintLayout>
```

5.SYSTEM TESTING

System testing of software or hardware is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements. System testing is carried out by specialist's testers or independent testers. The system testing should investigate both functional and non-functional requirements of the testing.

5.1. PROJECT TESTING

The project testing is used to test the whole project by using the different of testing techniques. Testing is a series of different tests that whose primary purposefully exercise the computer-based system.

5.1.1. UNIT TESTING:

Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code. may require developing test driver modules or test harnesses.

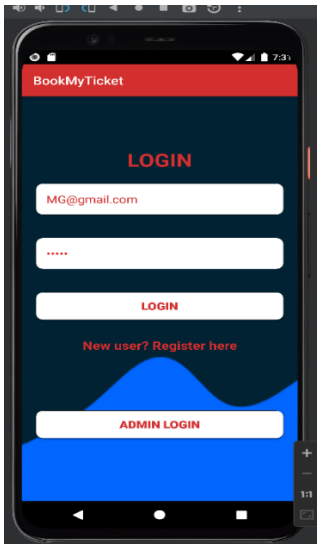


Fig 5.1 Unit Testing on Login Page

5.2. INTEGRATION TESTING:

Testing of integrated modules to verify combined functionality after integration. Modules are typically code modules, individual applications, client and server applications on a network, etc. This type of testing is especially relevant to client/server and distributed system.

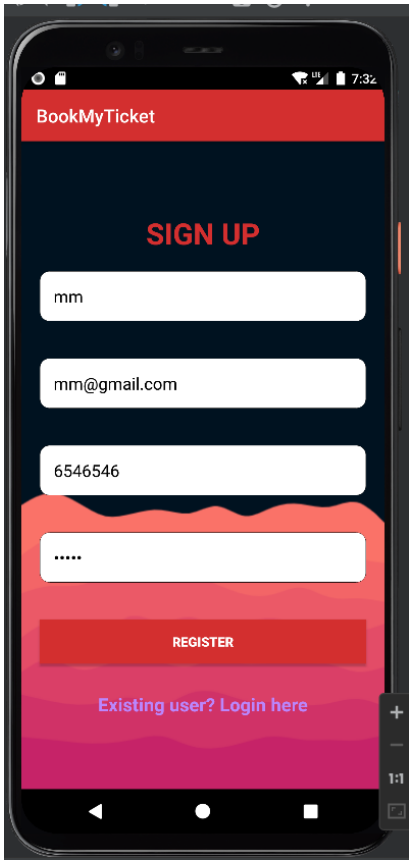


Fig 5.2 Integration Testing on register Page

5.3 VALIDATION TESTING

This model of testing is extremely important especially if someone want to be one of the best software testers. The software verification and validation testing is the process after the validation testing stage is secondary to verification testing.

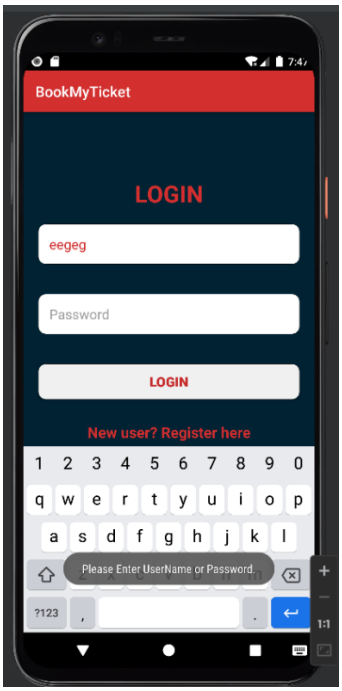


Fig 5.3 Validation Testing on Login (After Login)

Table 5.1 Validation Testing on Login Page

FIELD NAME	FIELD TYPE	VALIDATION TEST
User Name	Text	Required Field Validator
Password	Number	Range Validator

6. SYSTEM IMPLEMENTATION

System Implementation is the stage of the project when the theoretical design is turned out into a working system. Thus, it can be consider being the most critical stage in achieving a successful new system and in giving the user, confidence that the new system will work and be effective.

6.1 SOFTWARE DEPLOYMENT

Software deployment is the process of making the application work on a target device, whether it be a test server, production environment or a user's computer. Software deployment should only take place after through testing to ensure that all the flaws a bug have been identified a fixed. It is usually a planned initiative that consists of different steps or stages that occur in the production of operational software.

6.2 SOFTWARE DEMONSTRATION

Demo software is a trial version of a software program which allows people to use it for free while they decide whether or not to buy it. For example, a program may not allow people to save files, meaning that people can use the program to see how it feels, but they cannot save the work they produce.

6.2.1 . Login Page

The created user account can log-in using their E-mail id and password to access the application.

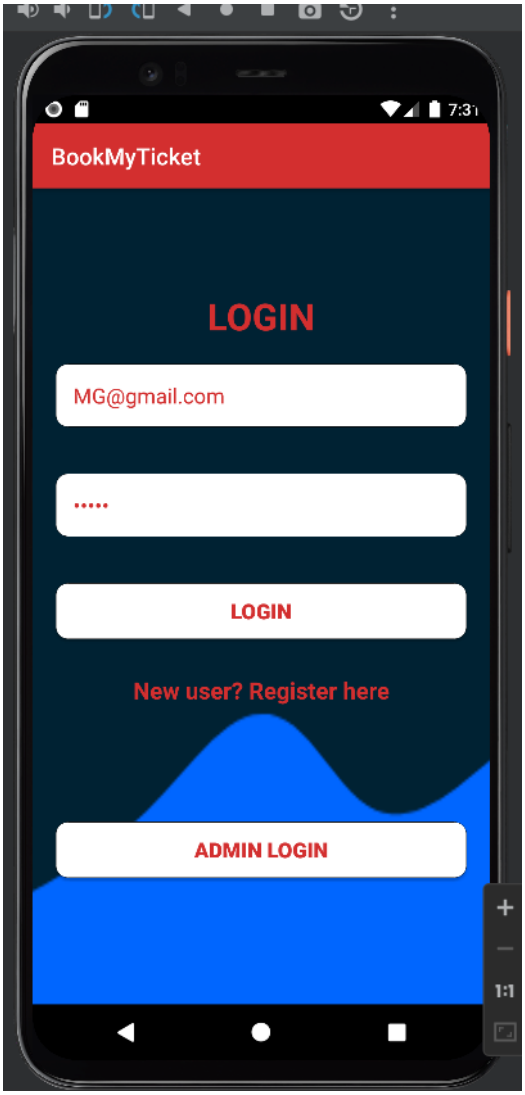


Fig 6.1 Login Page

6.2.2 . Sign in Page

The new user page or Registration page deals the details of the user such as information like full name, mobile number, password and confirm password.

Fig 6.2 Sign Page

6.2.3 . Home

The user can know what movies in screen now running on theaters easily using this application

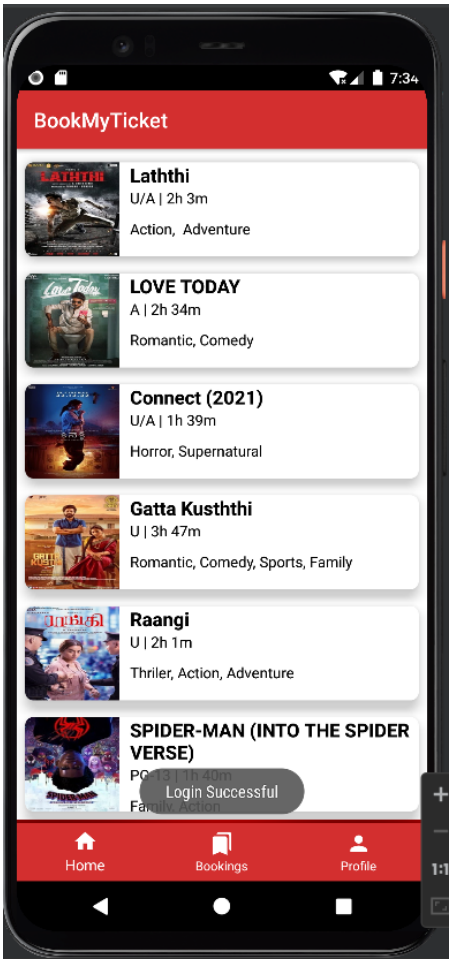


Fig 6.3 Home Page

6.2.4 . Admin login

The created Admin account can log-in using their E-mail id and password to access the application.

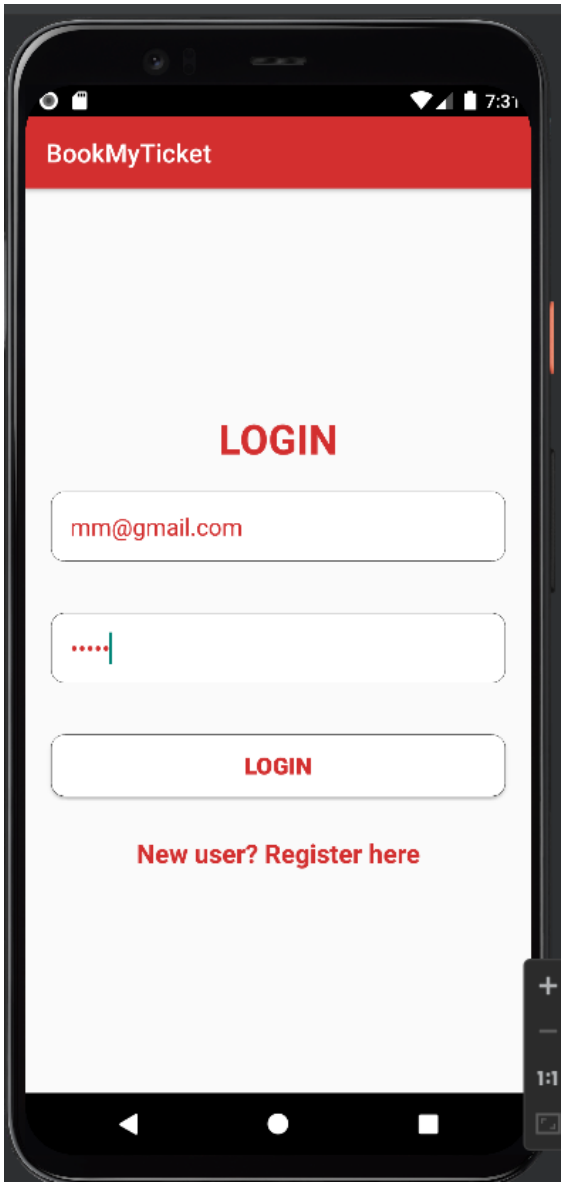


Fig 6.4 Admin Login

6.2.5 . Movie Details

The user can see the movie details in date and time screen etc.

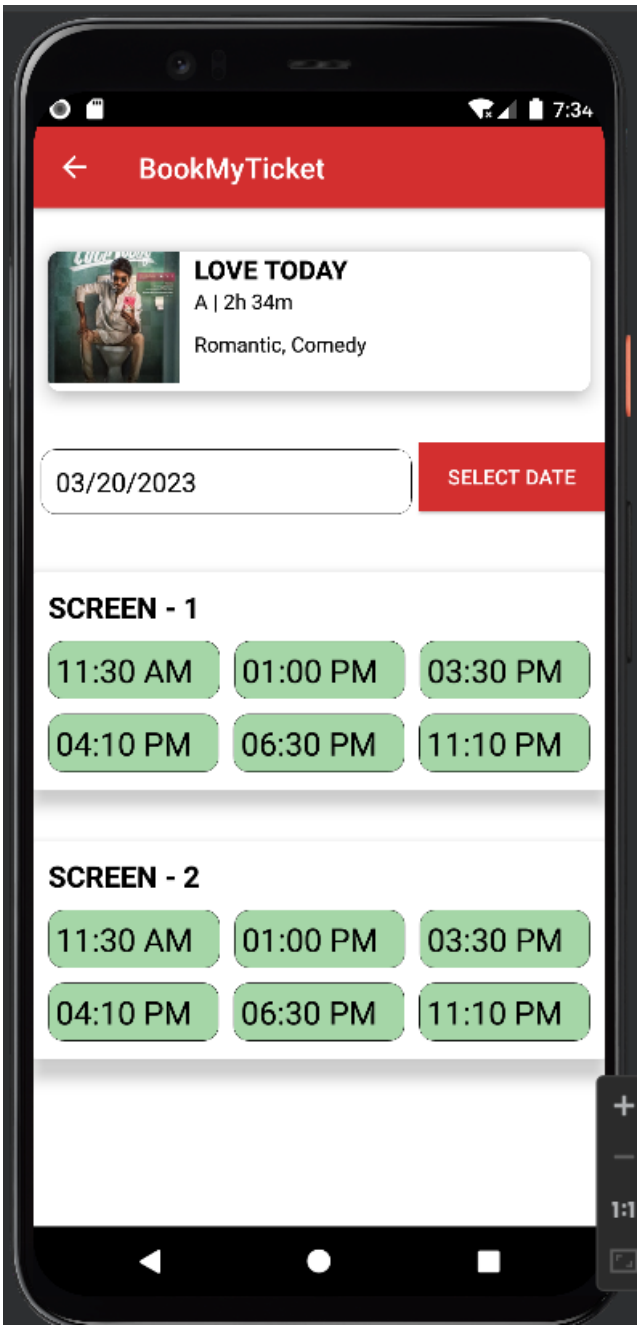


Fig 6.5 Movie Details

6.2.6 . Seat selection

The user can select the seats in the movie seat page

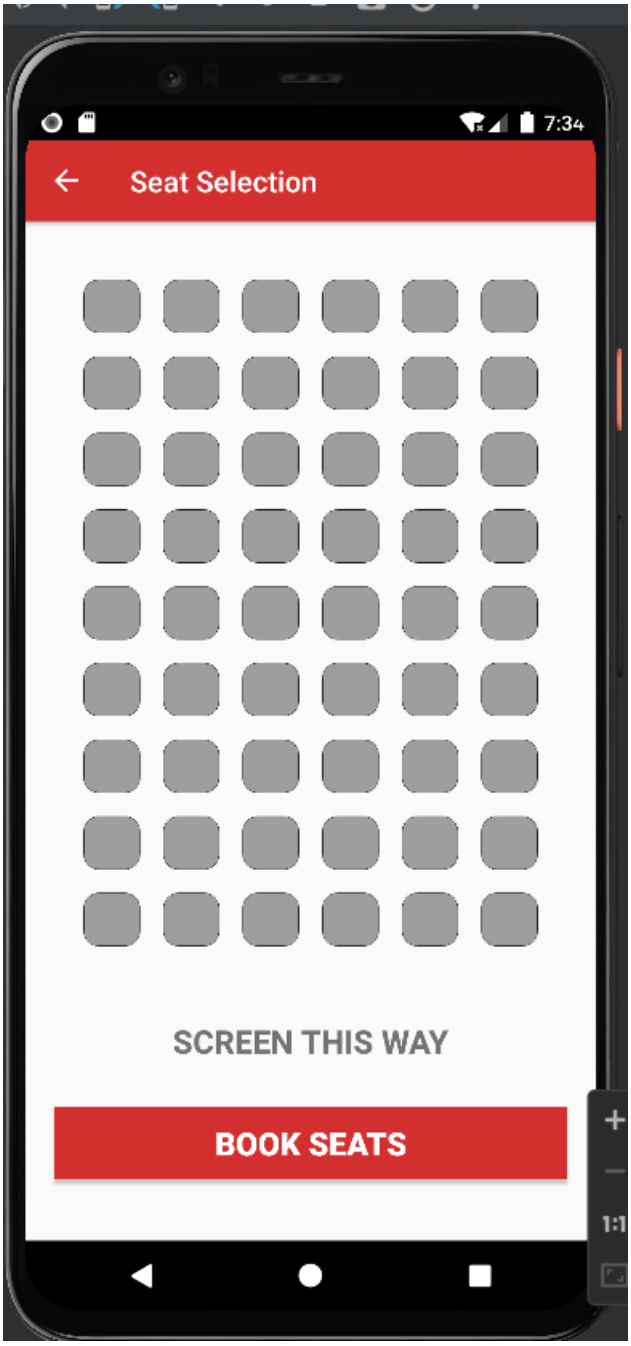


Fig 6.6 Seat selection

7 CONCLUSION

The Mobile Application “theatre Ticket Booking” is designed, tested and implemented successfully. It has features to book all the kinds of movie ticket, provide coordination and user communication, improve day-to-day operations. All the basic requirements of a Ticket booking application are provided in the proposed mobile application in order to manage it perfectly.

We have designed the project to provide the user with easy retrieval of data, details of theatre and necessary feedback as much as possible. In this project, the user is provided with a application that can be used to book movie tickets online. To implement this as a web application. has advantages such as enhanced performance, scalability, built- in security and simplicity.

8 BIBLIOGRAPHY

Books References:

1. Barry A. Burd, “Android Application Development”,2020.
2. Lauren Darcey, “Introduction to Android Application”,2013

Web References:

1. https://google-developer-training.github.io/android-developer-fundamentals- course-concepts-v2/unit-1-get-started/lesson-1-build-your-first-app/1-0-c- introduction-to-android/1-0-c-introduction-to-android.html
2. https://developer.android.com/training/basics/firstapp
3. https://github.com/AppIntro/AppIntro