Ticketbookr for android

- A Movie tickets booking app for employees of a firm.

Project by

- Akshat Jain
- Amrutha Dude
- Pranjal Chaubey
- Trivendra Singh

Timeline

- May 28: Project allotment.
- May 29 - June 7: Brush up on Android Studio and Java
- June 7: Planning and Roadmap development and design.
- June 8 - June 21: App design and Layout.
- June 16 Backend brushing.
- June 17- July 1: Backend development for local database (SQlite)
- June 25 - July 3: Realtime Database using firebase console.
- July 4 - July 8: App testing.

Background

- The inspiration for this app is BookMyShow, a renowned mobile app and website which lets us book movie tickets.
- The scope of this project is limited to the company theatre.
- The targeted users are the employees of the company.
- On a successful login, the app lists all the movies currently being showcased in the company theatre

ANDROID

- **Android** is a **software platform and operating system** for mobile devices based on the **Linux operating system** and developed by **Google** and the **Open Handset Alliance**.
- It allows developers to write managed code in a Java-like language that utilizes Google-developed Java libraries ,but does not support programs developed in native code.
- The unveiling of the **Android platform** on 5 November 2007 was announced with the founding of the **Open Handset Alliance**, a consortium of 34 hardware, software and telecom companies devoted to advancing open standards for mobile devices.
- When released in 2008, most of the Android platform will be made available under the **Apache free-software** and open-source license
- . This presentation first introduces the **Android Development Concepts** and then explains the Application Framework and the complexities that follow it.

FEATURES OF ANDROID OPERATING SYSTEM

1. Application Framework

It is used to write applications for Android. Unlike other embedded mobile environments, Android applications are all equal. From the point of security, the framework is based on UNIX file system permissions that assure applications have only those abilities that mobile phone owner gave them at install time.

2. Dalvik Virtual Machine

It is extremely low-memory based virtual machine, which was designed especially for Android to run on embedded systems and work well in low power situations. The Dalvik VM creates a special file format (.DEX) that is created through build time post processing.

3. SQLite

Extremely small (< 500kb) relational database management system, is integrated in Android. It is based on function calls and single file, where all definitions, tables and data are stored. This simple design is more than suitable for a platform such as Android.

4. Data Storage

SQLite is used for structured data storage .SQLite is a powerful and lightweight relational database engine available to all applications.

5. Connectivity SQLite

Extremely small (< 500kb) relational database management system, is integrated in Android. It is based on function calls and single file, where all definitions, tables and data are stored. This simple design is more than suitable for a platform such as Android. Android supports a wide variety of connectivity technologies including GSM,CDMA, Bluetooth, EDGE, EVDO, 3G and Wi-Fi.

6. Web Browser

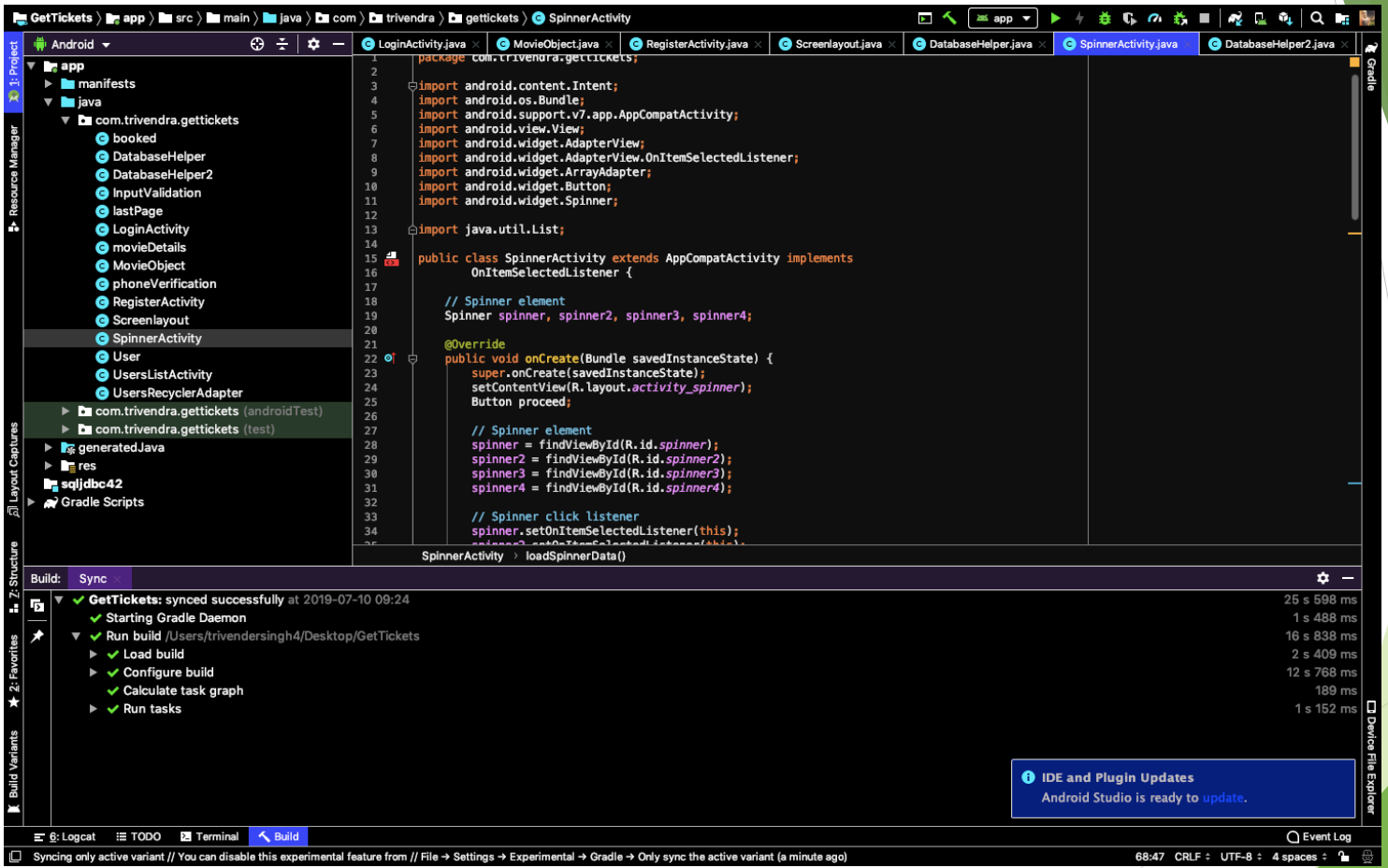
The web browser available in Android is based on the open-source Web Kit application framework. It includes LibWebCore which is a modern web browser engine which powers both the Android browser and an embeddable web view.

7. Java Virtual Machine

Software written in Java can be compiled into Dalvik byte codes and executed in the Dalvik virtual machine, which is a specialized VM implementation designed for mobile device use, although not technically a standard Java Virtual Machine.

STEP 1(Learn ANDROID STUDIO)

- Android is a linux-based Open Source operating system for mobile phones, TV, etc.
- Android Studio is a variant of IntelliJ Idea IDE used for developing android apps.
- It basically uses JAVA or Kotlin to create activities, which control the working of various objects of app, and XML, which shows how the app looks.


GetTickets › app › src › main › java › com › trivendra › gettickets › SpinnerActivity
app
Android
app
manifests
java
com.trivendra.gettickets
booked
DatabaseHelper
DatabaseHelper2
InputValidation
lastPage
LoginActivity
movieDetails
MovieObject
phoneVerification
RegisterActivity
Screenlayout
SpinnerActivity
User
UsersListActivity
UsersRecyclerAdapter
com.trivendra.gettickets (androidTest)
com.trivendra.gettickets (test)
generatedJava
res
sqljdbc42
Gradle Scripts
LoginActivity.java
MovieObject.java
RegisterActivity.java
Screenlayout.java
DatabaseHelper.java
SpinnerActivity.java
DatabaseHelper2.java
package com.trivendra.gettickets;
import android.content.Intent;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import android.view.View;
import android.widget.AdapterView;
import android.widget.AdapterView.OnItemSelectedListener;
import android.widget.ArrayAdapter;
import android.widget.Button;
import android.widget.Spinner;
import java.util.List;
public class SpinnerActivity extends AppCompatActivity implements
OnItemSelectedListener {
// Spinner element
Spinner spinner, spinner2, spinner3, spinner4;
@Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_spinner);
Button proceed;
// Spinner element
spinner = findViewById(R.id.spinner);
spinner2 = findViewById(R.id.spinner2);
spinner3 = findViewById(R.id.spinner3);
spinner4 = findViewById(R.id.spinner4);
// Spinner click listener
spinner.setOnItemSelectedListener(this);
SpinnerActivity › loadSpinnerData()
Build:
Sync
GetTickets: synced successfully at 2019-07-10 09:24
25 s 598 ms
Starting Gradle Daemon
1 s 488 ms
Run build /Users/trivendersingh4/Desktop/GetTickets
16 s 838 ms
Load build
2 s 409 ms
Configure build
12 s 768 ms
Calculate task graph
189 ms
Run tasks
1 s 152 ms
IDE and Plugin Updates
Android Studio is ready to update.
6: Logcat
TODO
Terminal
Build
Event Log
Syncing only active variant // You can disable this experimental feature from // File → Settings → Experimental → Gradle → Only sync the active variant (a minute ago)
68:47
CRLF
UTF-8
4 spaces

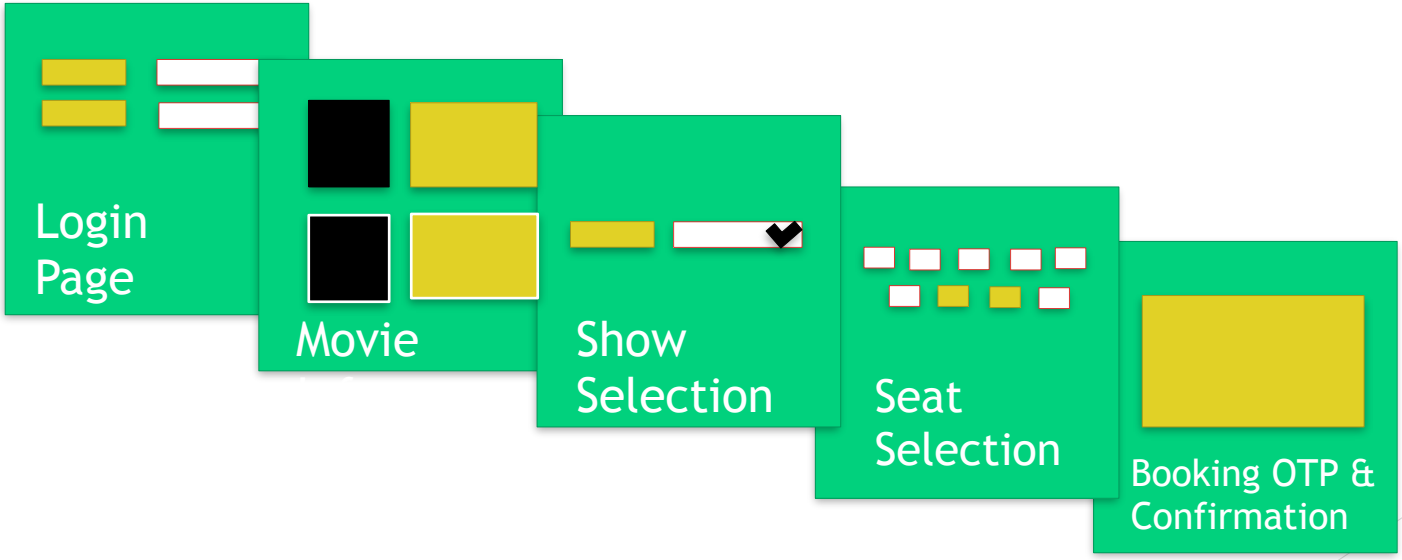
Project specs

- Project name: com.Grasim.ticketbookr
- Backward compatibility: Android 4.2
- Background knowledge required in: Java, Android Studio, SQL, SQLite framework.

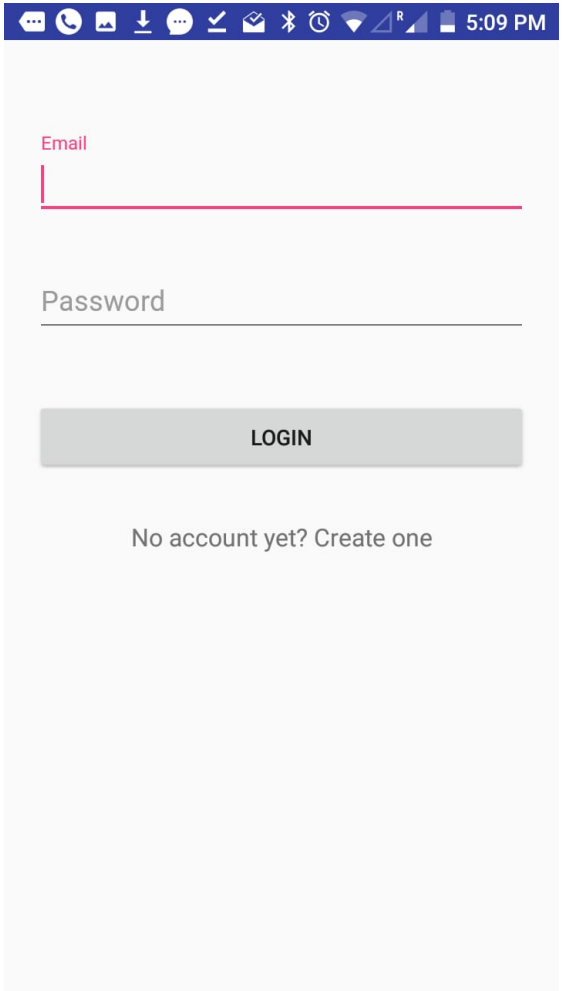
STEP 2

- App planning and Layout design

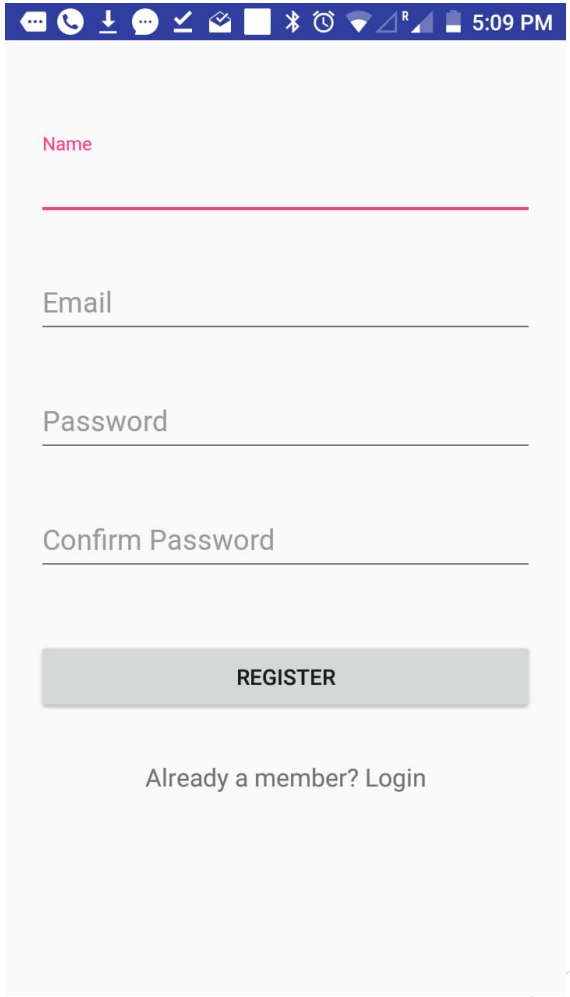
APP LAYOUT


Login Page
Movie
Show Selection
Seat Selection
Booking OTP & Confirmation

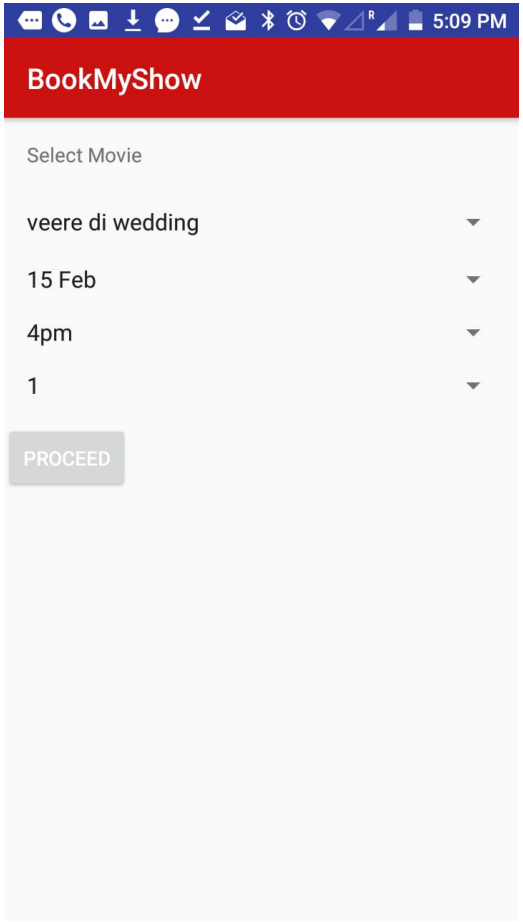
Login page



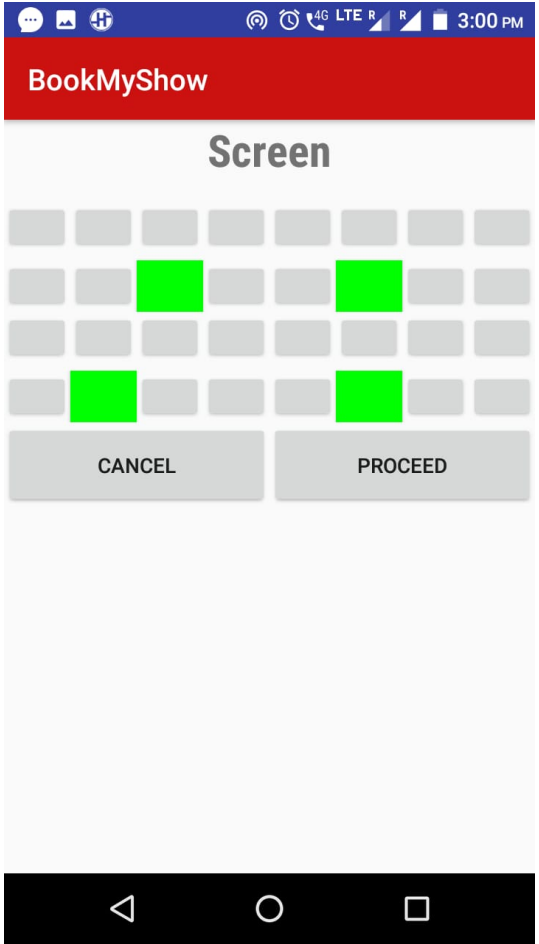
Forgot Password

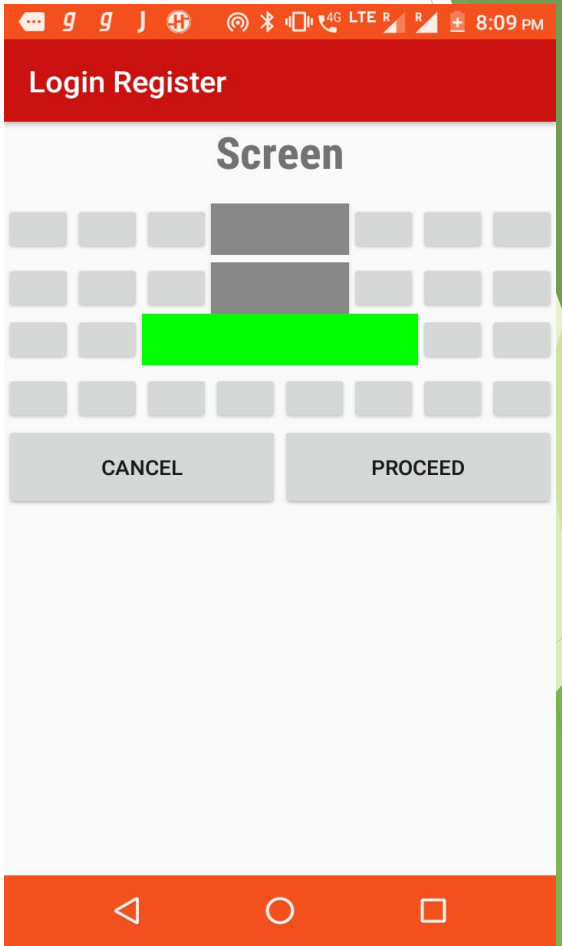


Booking Page

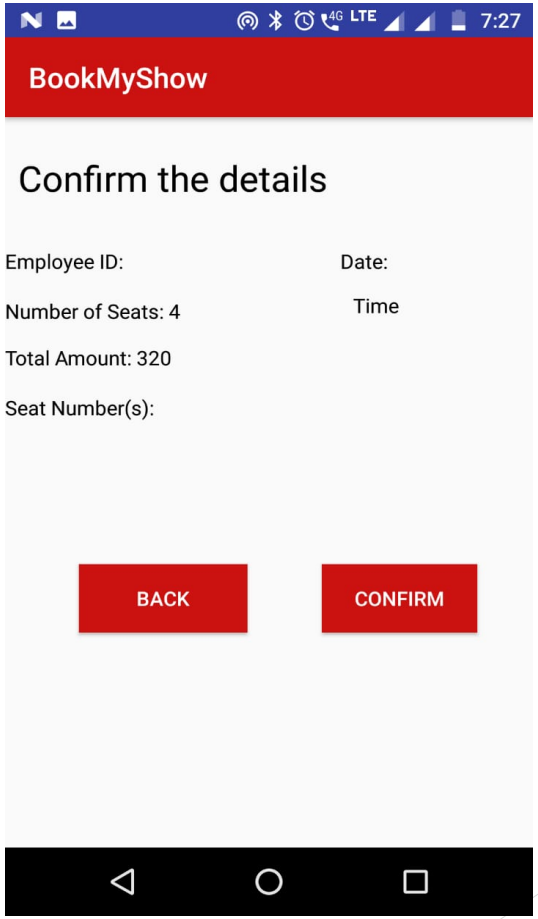


Seat Booking page

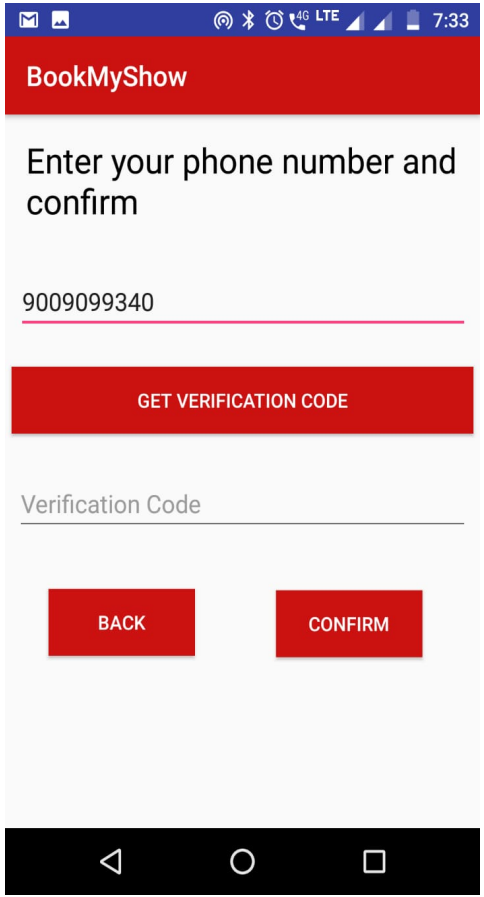


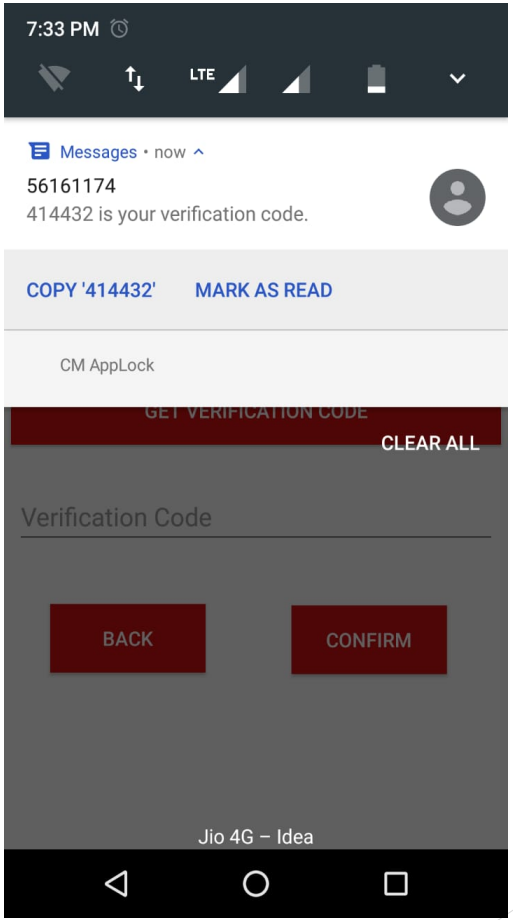


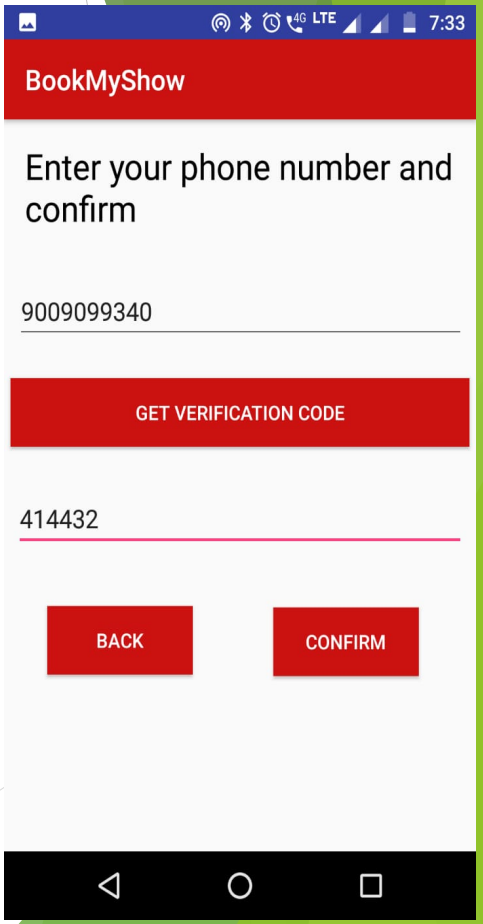
Confirmation and Payment



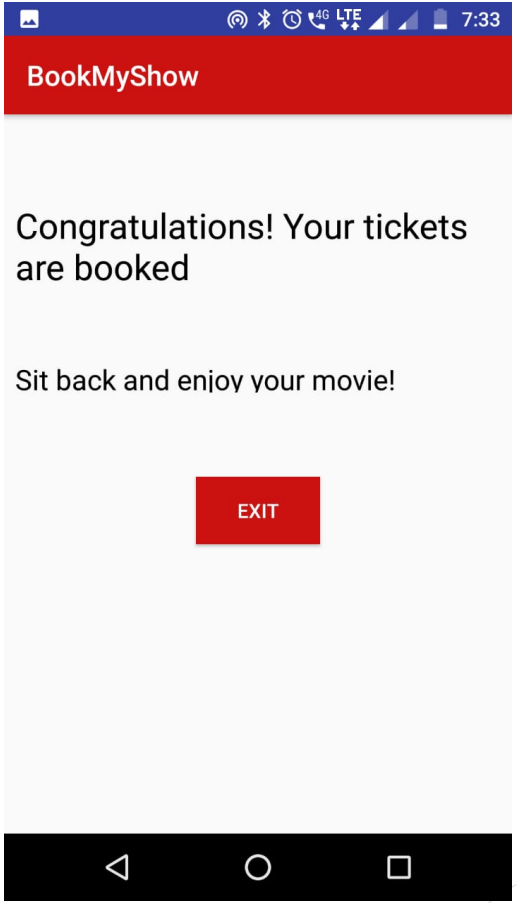
Confirmation(OTP) and Payment







Confirmation and Payment



STEP 3 (Firebase Integration)

Firebase Auth

Firebase Auth is a service that can authenticate users using only client-side code. It supports social login providers Facebook, GitHub, Twitter and Google (and Google Play Games). Additionally, it includes a user management system whereby developers can enable user authentication with email and password login stored with Firebase.

STEP 3 (Firebase Integration)

Firebase Realtime Database

Firebase provides a realtime database and backend as a service. The service provides application developers an API that allows application data to be synchronised across clients and stored on Firebase's cloud. The company provides client libraries that enable integration with Android, iOS, JavaScript, Java, Objective-C, Swift and Node.js applications. The database is also accessible through a REST API and bindings for several JavaScript frameworks such as AngularJS, React, Ember.js and Backbone.js. The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Developers using the realtime database can secure their data by using the company's server-side-enforced security rules.

Thank you