

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.

Android Studio interface showing the SpinnerActivity.java file in the com.trivendra.gettickets package. The file contains the following code:

```
1 package com.trivendra.gettickets;
2
3 import android.content.Intent;
4 import android.os.Bundle;
5 import android.support.v7.app.AppCompatActivity;
6 import android.view.View;
7 import android.widget.AdapterView;
8 import android.widget.AdapterView.OnItemClickListener;
9 import android.widget.ArrayAdapter;
10 import android.widget.Button;
11 import android.widget.Spinner;
12
13 import java.util.List;
14
15 public class SpinnerActivity extends AppCompatActivity implements
16     OnItemSelectedListener {
17
18     // Spinner element
19     Spinner spinner, spinner2, spinner3, spinner4;
20
21     @Override
22     public void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_spinner);
25         Button proceed;
26
27         // Spinner element
28         spinner = findViewById(R.id.spinner);
29         spinner2 = findViewById(R.id.spinner2);
30         spinner3 = findViewById(R.id.spinner3);
31         spinner4 = findViewById(R.id.spinner4);
32
33         // Spinner click listener
34         spinner.setOnItemSelectedListener(this);
35
36         SpinnerActivity > loadSpinnerData()
```

The Build output shows the following tasks and their durations:

Task	Duration
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.

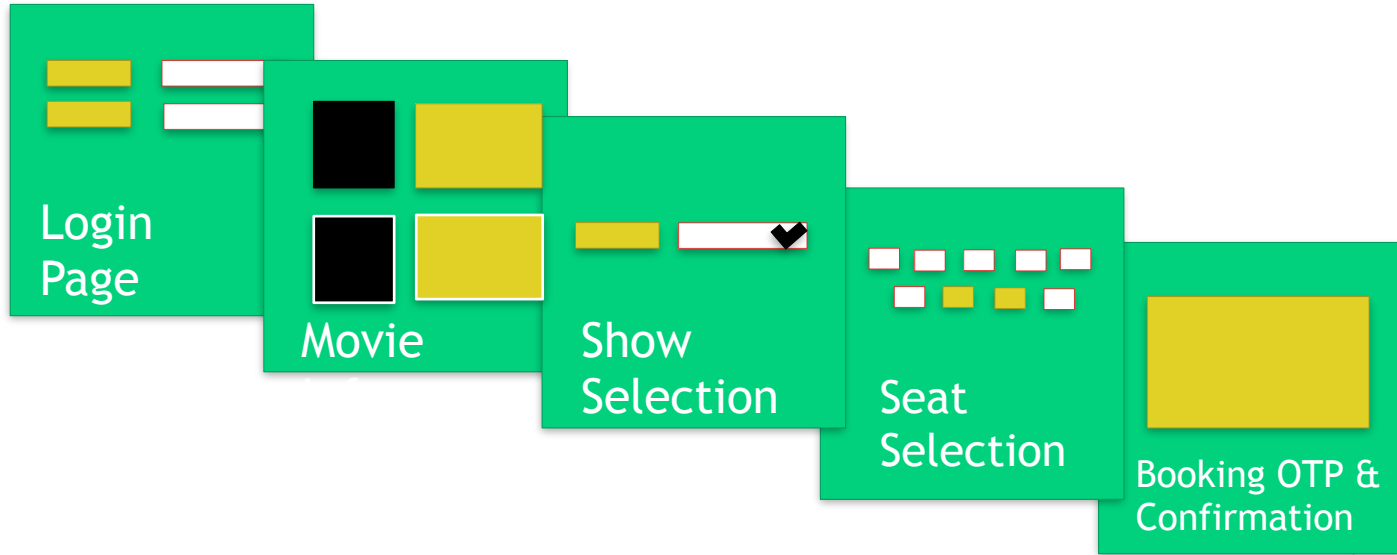
Project specs

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

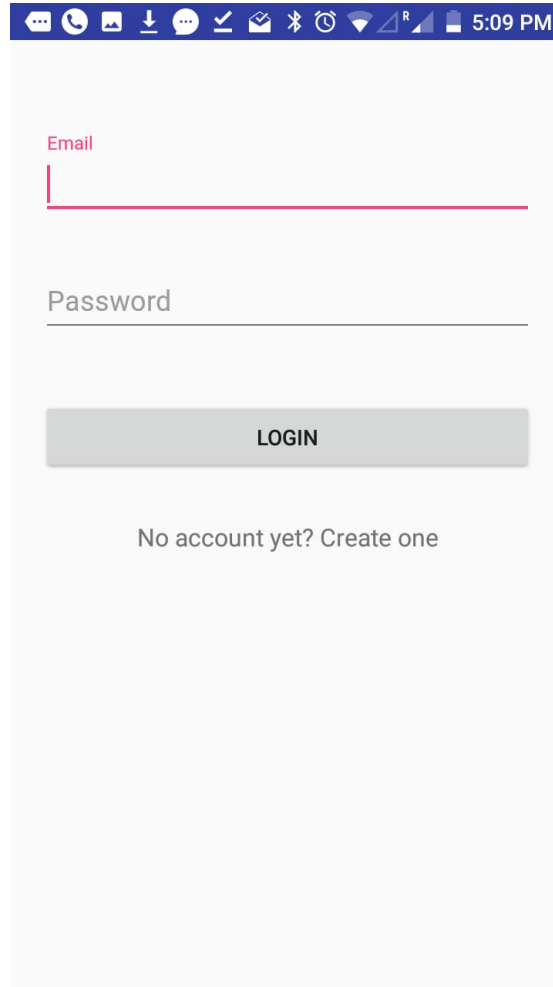
STEP 2

- ▶ App planning and Layout design

APP LAYOUT



Login page



A mobile application login screen mockup. At the top is a dark blue status bar with white icons for notifications, voice calls, messages, downloads, a checklist, email, Bluetooth, alarm, Wi-Fi, cellular signal, and battery level, followed by the time 5:09 PM. The main area is a light gray card. It features an 'Email' label in pink above a pink-outlined input field. Below this is a 'Password' label in gray above a gray-outlined input field. A gray 'LOGIN' button is centered below the fields. At the bottom, the text 'No account yet? Create one' is displayed in gray.

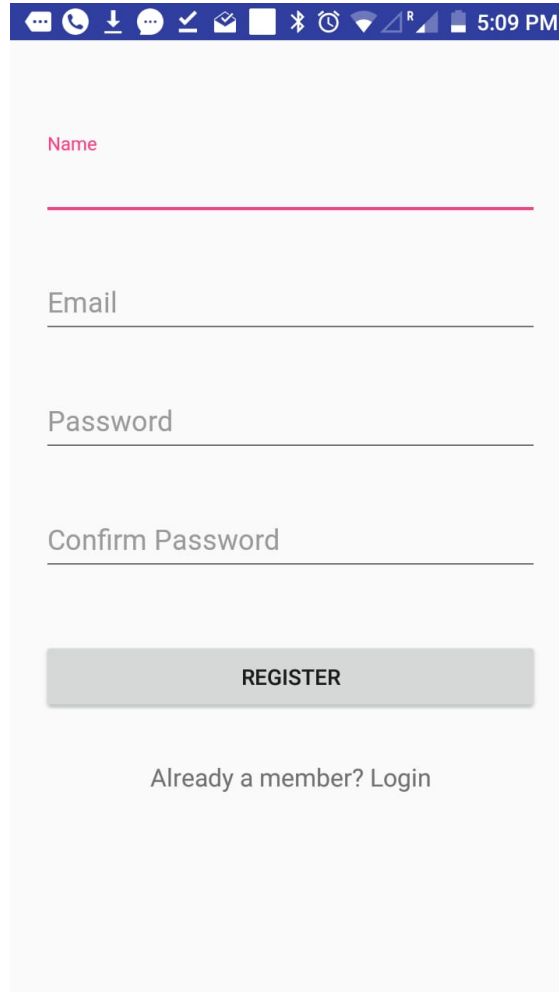
Email

Password

LOGIN

No account yet? Create one

Forgot Password

A mobile application registration form displayed on a smartphone screen. The screen has a dark blue status bar at the top with various icons and the time 5:09 PM. The form itself is on a light gray background and contains four text input fields: 'Name' (with a pink underline), 'Email', 'Password', and 'Confirm Password' (all with gray underlines). Below these fields is a gray 'REGISTER' button. At the bottom, there is a link that says 'Already a member? Login'.

5:09 PM

Name

Email

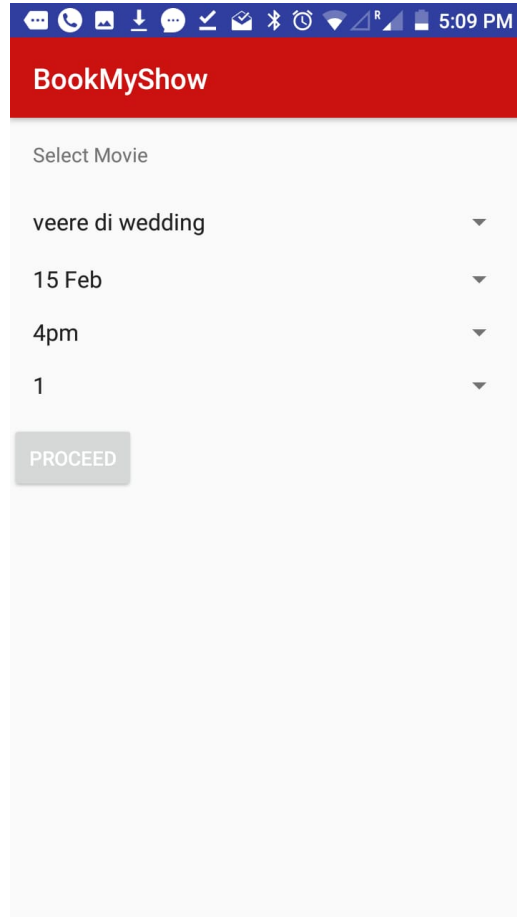
Password

Confirm Password

REGISTER

Already a member? [Login](#)

Booking Page



The screenshot shows a mobile application interface for 'BookMyShow'. At the top is a red header with the app name. Below it is a light gray area with a 'Select Movie' label. Four items are listed: 'veere di wedding', '15 Feb', '4pm', and '1', each with a downward arrow on the right. At the bottom of this area is a gray 'PROCEED' button. The top of the screen shows a standard Android status bar with various icons and the time '5:09 PM'.

BookMyShow

Select Movie

veere di wedding ▼

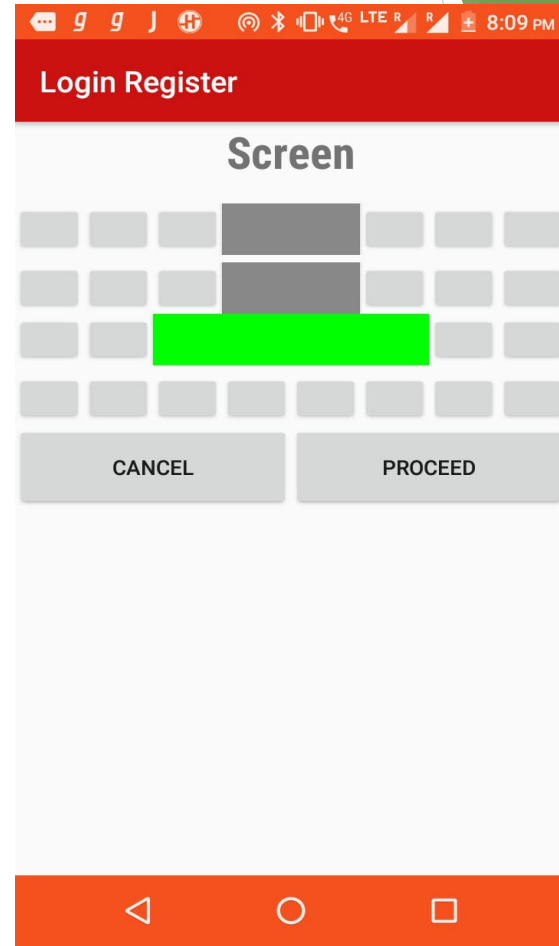
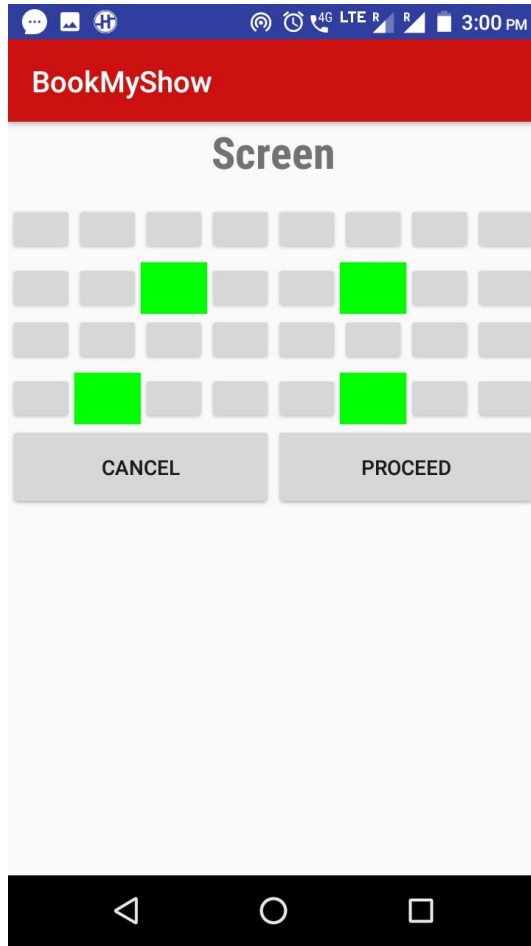
15 Feb ▼

4pm ▼

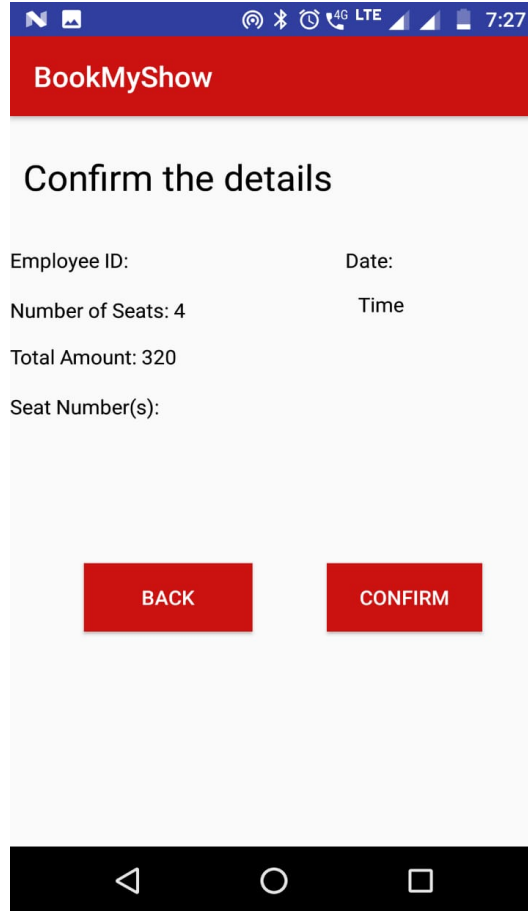
1 ▼

PROCEED

Seat Booking page



Confirmation and Payment



The image shows a mobile application interface for 'BookMyShow'. At the top is a red header bar with the app name 'BookMyShow' in white. Below the header, the title 'Confirm the details' is displayed in a large, bold, black font. The main content area is light gray and contains several labels for confirmation: 'Employee ID:', 'Date:', 'Number of Seats: 4', 'Time', 'Total Amount: 320', and 'Seat Number(s):'. At the bottom of the screen, there are two red rectangular buttons with white text: 'BACK' on the left and 'CONFIRM' on the right. The entire app interface is framed by a black border at the bottom, which contains three white icons: a triangle, a circle, and a square. Above the app interface, a status bar shows various icons including signal strength, LTE, and the time 7:27.

BookMyShow

Confirm the details

Employee ID: Date:

Number of Seats: 4 Time

Total Amount: 320

Seat Number(s):

BACK CONFIRM

Confirmation(OTP) and Payment

BookMyShow

Enter your phone number and confirm

9009099340

GET VERIFICATION CODE

Verification Code

BACK CONFIRM

7:33 PM

Messages • now ^

56161174
414432 is your verification code.

COPY '414432' MARK AS READ

CM AppLock

GET VERIFICATION CODE CLEAR ALL

Verification Code

BACK CONFIRM

Jio 4G - Idea

BookMyShow

Enter your phone number and confirm

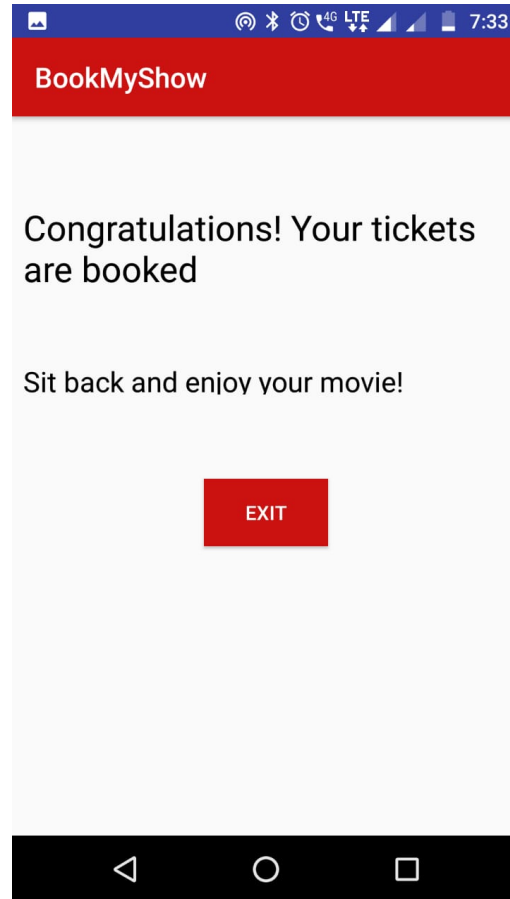
9009099340

GET VERIFICATION CODE

414432

BACK CONFIRM

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)

Firestore Realtime Database

Firestore 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 Firestore'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

The background of the slide is white with abstract green geometric shapes on the right side. These shapes include overlapping triangles and polygons in various shades of green, ranging from a light lime green to a dark forest green. The shapes are positioned on the right edge, creating a modern, layered effect.