#### Visvesvaraya Technological University

#### BELGAVI, KARNATAKA - 590 014.

****

#### A MINI PROJECT REPORT

#### ON

“HOTEL BOOKING SYSTEM”

#### Submitted By

# NIKHITHA R KARANTH [4PM20CS064]

# NOOR AYESHA NAAZ [4PM20CS065]

# TRUPTHI K [4PM20CS119]

# VAISHNAVI H S [4PM20CS120]

## **Submitted in partial fulfillment of the requirements for the 6th Semester of**

**Bachelor of Engineering in**

**Computer Science and Engineering**

# UNDER THE GUIDANCE OF

**Mr. Ranjan V**

**Assistant Professor, Dept. of CS&E.**

##### PESITM, Shivamoga

##### pesitlogo

#### Department of Computer Science & Engineering

#### July-2023

#### PES Institute of Technology & Management

#### NH-206, Sagar Road, Shimoga-577 204

#### (Affiliated to Visvesvaraya Technological University, Belgavi)

**PES INSTITUTE OF TECHNOLOGY AND MANAGEMENT**

**Shivamogga-577204**

**Department of Computer Science & Engineering**

**CERTIFICATE**

This is to certify that the project work entitled **“HOTEL BOOKING SYSTEM”** is a bonafied work carried out by **Nikhitha R Karanth (4PM20CS064), Noor Ayesha Naaz (4PM20CS065), Trupthi K(4PM20CS119) and Vaishnavi H S(4PM20CS120)** in partial fulfillment for the 6th Semester of Computer Science and Engineering of the Visvesvaraya Technological University, Belgaum during the academic year 2023. It is certified that all corrections and suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of Project Work prescribed for the said degree.

**Mr. Ranjan V Dr. Arjun U**

Assistant Professor, Dept of CS&E. HOD, Dept of CS&E

PESITM, Shivamogga PESITM, Shivamogga

**Name of the examiners Signature with Date**

**1)……………………………**

**2)……………………………**

**ACKNOWLEDGEMENT**

While presenting this MOBILE APPLICATION DEVELOPMENT mini project on “**Hotel Booking System”,** we feel that it is our duty to acknowledge the help rendered to us by various persons.

Firstly we thank god for showering his blessings on us. We are grateful to our institution City Engineering College for providing us a congenial atmosphere to carry out the project successfully. We would like to express our heartfelt gratitude to**, Dr. Chaitanya Kumar M.V.,** Principal, PESITM, Shimoga, for extending his support.

We would also like to express our heartfelt gratitude to **Dr. Arjun U,** HOD, Computer Science and Engineering whose guidance and support was truly valuable.

We are very grateful to our **Mr. Ranjan V**, and our co guides, Department of Computer Science, for their able guidance and valuable advice at every stage of our project which helped me in the successful completion of our project.

We would also have indebted to our parent and friends for their continued moral and material support throughout the course of project and helping me in finalize the presentation.

Our hearty thanks to all those have contributed bits, bytes and words to accomplish this project.

|  |  |
| --- | --- |
| **Nikhitha R Karanth** | **(4PM20CS064)** |
| **Noor Ayesha Naaz** | **(4PM20CS065)** |
| **Trupthi K** | **(4PM20CS119)** |
| **Vaishnavi H S** | **(4PM20CS120)** |

**DECLARATION**

We student of 6th semester BE, Computer Science and Engineering department hereby declare that project work entitled “Hotel Booking System” has been carried out by us at PESITM, Shimoga and submitted in partial fulfillment of the course requirement for the award of the degree of Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belgaum, during the academic year 2020-2021.

We also declare that, to the best of our knowledge and belief, the work reported here does not form the part of dissertation on the basis of which a degree or award was conferred on a earlier occasion on this by any other student.

Date:

Place: Shimoga

|  |  |
| --- | --- |
| **Nikhitha R Karanth** | **(4PM20CS064)** |
| **Noor Ayesha Naaz** | **(4PM20CS065)** |
| **Trupthi K** | **(4PM20CS119)** |
| **Vaishnavi H S** | **(4PM20CS120)** |

**ABSTRACT**

The Hotel Booking System mobile app is a comprehensive solution that focuses on enhancing the hotel booking experience. With a user-friendly interface, the app allows guests to easily search for available rooms, view detailed descriptions and amenities, and make reservations seamlessly. The app's robust backend ensures real-time room availability updates, reducing the risk of double bookings and providing accurate information to both guests and hotel staff. By integrating a secure payment gateway, the app facilitates hassle-free transactions, making the booking process convenient and efficient.

For hotel staff, the app provides a centralized dashboard to manage room inventory, bookings, and guest information. It offers a comprehensive overview of reservations, enabling staff to efficiently allocate rooms, track occupancy rates, and plan for maintenance or cleaning tasks. With instant notifications and alerts, the app keeps hotel staff informed about new bookings, modifications, and cancellations, allowing for swift responses and optimal room management.

Overall, the Hotel Management System mobile app optimizes the booking process, improving guest satisfaction and increasing operational efficiency for hotel owners and staff. By streamlining reservations, providing real-time updates, and offering convenient payment options, the app aims to deliver a seamless booking experience, ultimately contributing to the success of the hotel.

**TABLE OF CONTENTS**

**Chapter1** INTRODUCTION 3

* 1. [Introduction to android development 3](#_TOC_250000)

1.2 Feature of Android

1.3 Project Goal or Objective

**Chapter2** REQUIREMENTS SPECIFICATION 8

* 1. Software and Hardware Requirements 8
  2. About Android Studio 8

3.3 Programming Languages used in Android development

**Chapter3** SOFTWARE DESIGN 9

* 1. Design

4.1.1[User interface 9](#_bookmark4)

4.1.2[Layout editor 10](#_bookmark5)

4.1.3 [Add a textbox 11](#_bookmark6)

4.1.4 [Add a button 11](#_bookmark7)

4.1.5 [Change UI string 12](#_bookmark8)

4.2 Implementation 4.2.1 Flow diagram

4.2.2 Function Used in Java

4.2.3 Source Code

**Chapter4** SNAP SHOTS 17

**Chapter5** CONCLUSION AND FUTURE SCOPE 20

**Chapter8** BIBILIOGRAPHY 21

[8.1 References 2](#_bookmark12)1

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **Figure No.** | **Figure Name** | **Page No.** |
| 1 | Android Devices | 3 |
| 2 | Welcome To Android Studio | 4 |
| 3 | Android Studio Main Window | 6 |
| 4 | Functional Requirements | 7 |
| 5 | User Interface | 9 |
| 6 | Layout Editor | 10 |
| 7 | Add A Textbox | 11 |
| 8 | Add A Button | 11 |
| 9 | Change UI String | 12 |
| 10 | Interface1 | 17 |
| 11 | Interface2 | 17 |
| 12 | Android Text Bar View | 18 |
| 13 | Output View | 18 |
| 14 | Output Snapshot1 | 19 |
| 15 | Output Snapshot2 | 19 |
|  |  |  |

**Chapter 1**

**INTRODUCTION**

**The Hotel Booking System** mobile app revolutionizes the hotel booking process by offering a user-friendly interface for guests to search for available rooms, view details, and make reservations seamlessly. With real-time updates and a secure payment gateway, guests experience a streamlined and convenient booking experience. Hotel staff benefit from a centralized dashboard for efficient room management, ensuring optimal allocation and enhanced operational efficiency. The app aims to optimize the hotel booking experience, improve guest satisfaction, and elevate the overall effectiveness of hotel management.

* 1. **Introduction to Mobile application development**

**Mobile application development** is the process of creating software applications that run on a mobile device, and a typical mobile application utilizes a network connection to work with remote computing resources. Hence, the mobile development process involves creating installable software bundles (code, binaries, assets, etc.) , implementing backend services such as data access with an API, and testing the application on target devices. There are two dominant platforms in the modern smartphone market. One is the iOS platform from Apple Inc. The iOS platform is the operating system that powers Apple's popular line of iPhone smartphones. The second is Android from Google. The Android operating system is not only used by Google devices but also by many other OEMs to built their own smartphones and other smart devices.



**Figure 1.1 Devices running on Android Operating System**

**1.1.1 ANDROID**

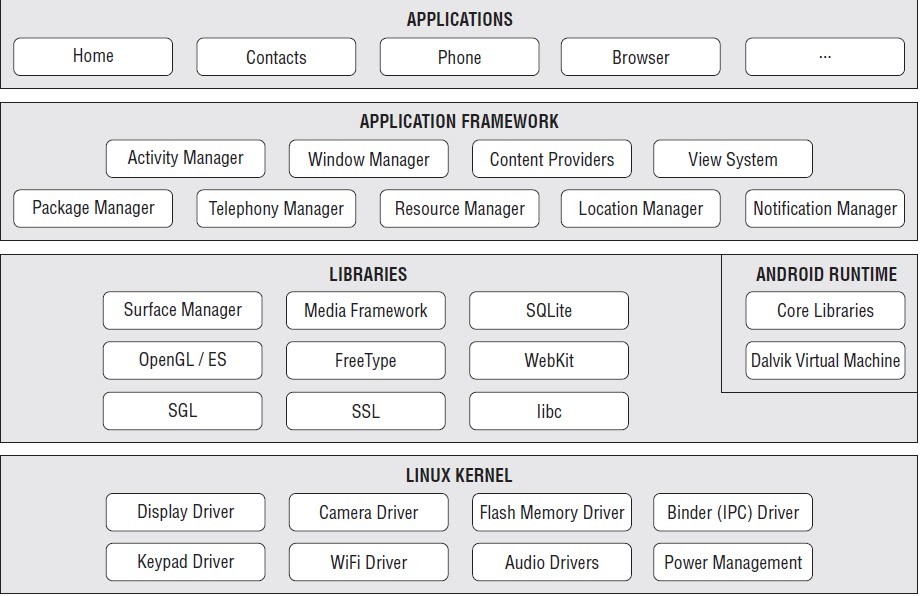
Android is an operating system and programming platform for smart phones and other mobile devices (such as tablets). It includes a software development kit for writing original code and assembling software modules to create apps for Android users. It also provides a marketplace to distribute apps. All together, Android represents an ecosystem for mobile apps. Android is a widely-adopted open source project.

Android Studio is Android's official IDE. It is purpose-built for Android to accelerate your development and help you build the highest-quality apps for every Android device.

**1.1.2 History:**

This was owned by Open Handset Alliance. Google formed a group of hardware, software, and telecommunication companies called the Open Handset Alliance with the goal of contributing to Android development. They then made up of 84 companies.The first beta version of the Android Software Development Kit (SDK) was released by Google in 2007 where as the first commercial version, Android 1.0, was releasedin September 2008.On June 27, 2012, at the Google I/O conference, Google announced the next Android version,4.1 Jelly Bean. Jelly Bean is an incremental update, with the primary aim of improving the user interface, both in terms of functionality and performance.

**1.1.3 Android Architecture:**



**Figure 1.2 Android Architecture**

**Android Applications:**

Your apps live at this level, along with core system apps for email, SMS messaging, calendars, Internet browsing, or contacts. Android applications are usually developed in the Java language using the Android Software Development Kit. Once developed, Android applications can be packaged easily and sold out either through a store such as GooglePlay, SlideME, Opera Mobile Store, Mobango, F-droid and the Amazon Appstore

**Java API Framework/Android Framework**

All features of Android are available to developers through application programming interfaces (APIs) writtenin the Java language. Like:

* + - View System: used to build an app's UI, including lists, buttons, and menus.
    - Resource Manager: used to access to non-code resources such as localized strings, graphics, and layout files.
    - Notification Manager: used to display custom alerts in the status bar.
    - Activity Manager: that manages the lifecycle of apps, etc.

**Libraries and Android Runtime**

Each app runs in its own process andwith its own instance of the Android Runtime, which enables multiple virtual machines on low-memory devices. Android also includes a set of core runtime libraries. Many core Android system components and services are built from native code that require native libraries written in C and C++. These native libraries are availableto apps through the Java API framework.

**Linux Kernel**

At it’s base, Android runs on a Linux kernel for interacting with the device’s processor, memory, etc. Thus an Android device can beseen as a Linux computer.

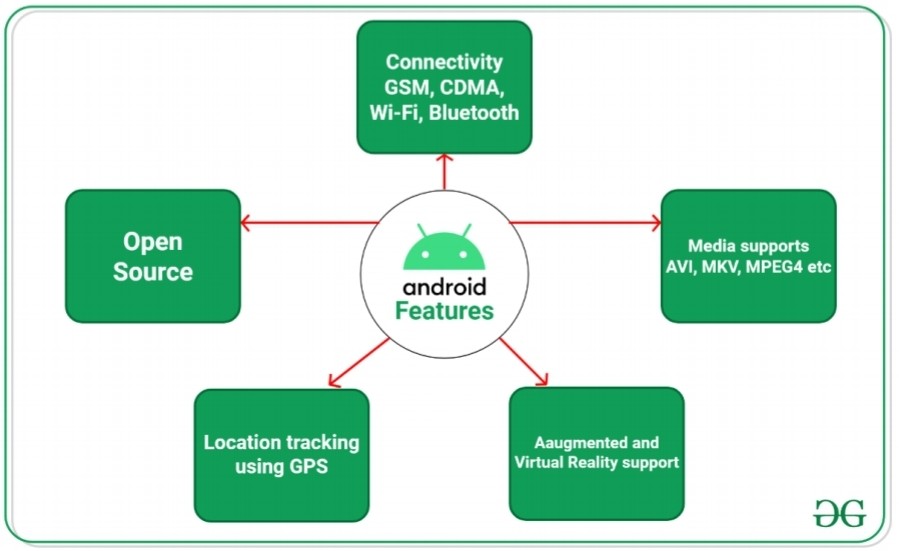
**1.2 Features of Android**

Android is a powerful open-source operating system that open-source provides immense features and some of these are listed below.

* Android Open Source Project so we can customize the OS based on our requirements.
* Android supports different types of connectivity for GSM, CDMA, Wi-Fi, Bluetooth, etc. for telephonic conversation or data transfer.
* Using wifi technology we can pair with other devices while playing games or using other applications.
* It contains multiple APIs to support location-tracking services such as GPS.
* We can manage all data storage related activities by using the file manager.
* It contains a wide range of media supports like AVI, MKV, FLV, MPEG4, etc. to play or record a variety of audio/video.
* It also supports different image formats like JPEG, PNG, GIF, BMP, MP3, etc.
* It supports multimedia hardware control to perform playback or recording using a camera and microphone.
* Android has an integrated open-source Web Kit layout based web browser to support

User Interface like HTML5, CSS3.

* Android supports multi-tasking means we can run multiple applications at a time and can switch in between them.
* It provides support for virtual reality or 2D/3D Graphics.



**Figure 1.3 Features of Android**

**1.4 Project Goal**

The goal of the Hotel Booking System mobile app project is to develop a comprehensive solution that simplifies the hotel booking process for guests while optimizing operational efficiency for hotel staff. By providing a user-friendly interface, real-time updates on room availability, and secure payment integration, the app aims to enhance the overall guest experience, improve room management, and streamline the operations of hotel owners and staff. Ultimately, the project seeks to deliver a seamless and convenient booking system that maximizes guest satisfaction and increases the profitability of the hotel

**CHAPTER 2**

**REQUIREMENT SPECIFICATION**

**2.1 Software and Hardware Requirements:**

Basic system requirements for installing Android Studio:

* Operating System Version

Microsoft Windows Microsoft Windows 7/8/10 (32- or 64-bitm

* Random Access Memory (RAM)

4 GB RAM minimum; 8 GB RAM recommended.

* Free digital storage

2 GB of available digital storage minimum, 4 GB Recommended (500 MB for IDE 1.5 GB for Android SDK and emulator system image).

* Minimum required JDK Version

Java Development Kit 8

* Minimum screen resolution 1280 x 800.

**2.2 About Android Studio**

Android Studio is exclusively designed for developing Android applications. It consists of all Android SDK tools to design, develop, maintain, test, debug and publish our app. The IDE is designed very efficiently which makes the developer’s job easy. It also supports the IntelliJ IDE, the main idea behind this IDE is that it automatically senses the variables, methods, classes, built-in functions or it could be anything else when we press the first letter of it. Say, suppose we declared few variables or methods that starts with an ‘S’, it automatically senses everything that starts with an ‘S’ and makes suggestions. It also supports Git as a version control system to maintain the app changes and push them into github. All java files, layout files (for design) are integrated into a single project easily. After the completion of project, the whole application could be put as an .APK (Android Package) file, in which we can run that APK file in any device and use the application. Other main tools include Android SDK, ADB, and Gradle Build.

**2.2.1Android Software Development Kit (SDK):**

One of the main tools used in developing android applications, as it packages many core features into one SDK and it can be used in the application easily. This helps us to avoid writing lot of code, and building applications faster.

**2.2.2 Android Debug Bridge (ADB):**

Android SDK uses ADB tool as a connection device which allows us to connect the Android Devices or Emulator with the machine via USB. After developing or while developing applications, we can connect with the device to check how the application runs. Later, we can debug and run the applications.

**2.2.3 Gradle Build:**

Gradle Scripts are the recent feature that is added to Android Studio. It is basically an automated build system which is used to automate the various phases involved in designing an application that includes design, development, test, debug, and publish. We need to configure the project and modules by mentioning all the supported jar files, SDK’s, version name, level, compiled SDK version, build tools version. to ensure that the developed app is compatible with the testing device/emulator. Gradle is also similar to Ant and Maven which helps in maintaining java projects (repositories).

**2.2.4 Android Device Monitor:**

If we want to access all the hidden files that are generated when we run the application, we can use the monitor. We can select any project and explore the files that are related to that project. But, as they are hidden files, we need root permissions to access them. Suppose, if we run the app in device, we need to root the device and run commands in adb shell to get permissions.

**2.2.5 SDK Manager:**

It is one of the main tools to maintain the updates of all the installed components required to run the project. It also notifies us when the project is not compatible with device or any other compatibility issues and to download any component that is required.

**2.2.6 AVD Manager:**

It is used to create virtual devices of any desired API level to support higher level SDK’s incase our device does not support. Using emulators to test the application is difficult as it might be little slower when compared to real device.

**2.2.7 SQLite Database:**

Android also supports inbuilt database which is Android SQLite to develop any small applications and perform any CRUD (Create, Update, and Delete) operations. As it is not flexible enough to support substantial number of data, for complex applications we are using other external databases.

**2.3 Programming Languages used in Developing Android Applications**

* + - Java
    - XML

**2.3.1 Java**

As the project is developing an Android Application, the default programming language is Java. All Android applications are built using Java in Android Studio or Eclipse or both. Java is a popular and widely used language throughout the world. As mentioned in, Java is one of the powerful programming languages like C, C++. developed by Sun Microsystems which has many powerful features as described below. After the development of C, C++, Java has come into evolution by addressing their drawbacks . It is one of the open source projects that could be easily installed in our machine. The language is also easy to learn, understand and implement. Java is used in various kinds of applications like Web, Desktop, Mobile, and Big Data. Many powerful features are supported by Java including various libraries, application services, graphics library for 2D/3D applications. The language is flexible enough to maintain code complexity, test, implementation, integration and support. Apart from these, there are other key features which make Java more special. It is object oriented programming language, one of the important hierarchies in the programming languages which is used to implement real time applications, provides for code reusability, it has a platform independence feature including any virtual machines(Write Once Read Everywhere), as in no need to write the code for different OS as the Java actual code is compiled irrespective of any machine, OS.

You create classes, define methods, instantiate objects, and call methods on those objects. But because you’re working within a framework, there is a set of code that already exists to call specific methods. As a developer, your task will be to fill in what these methods do in order to run your specific application.

Java is a popular programming language, created in 1995.

It is owned by Oracle, and more than **3 billion** devices run Java. It is used for:

* + - Mobile applications (specially Android apps)
    - Desktop applications
    - Web applications
    - Web servers and application servers
    - Games

**1.5 XML**

XML (Extensible Markup Language) is a markup language similar to [HTML](https://developer.mozilla.org/en-US/docs/Glossary/HTML), but without predefined tags to use. Instead, you define your own tags designed specifically for your needs. This is a powerful way to store data in a format that can be stored, searched, and shared. Most importantly, since the fundamental format of XML is standardized, if you share or transmit XML across systems or platforms, either locally or over the internet, the recipient can still parse the data due to the standardized XML syntax.The design goals of XML focus on simplicity, generality, and usability across the Internet. It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.

**XML Properties**

1. XML is a markup language that focuses on data rather than how the data looks.  
   2. XML is designed to send, store, receive and display data. In simple words you can say that XML is used for storing and transporting data.  
   3. XML became a W3C (W3C stands for World Wide Web Consortium, the main international standards

**CHAPTER 3**

**SYSTEM DESIGN AND IMPLEMENTATION**

**3.1 DESIGN**

**USER INTERFACE:**

For a login page in a Hotel Booking System application, you can design a user interface that

includes the following components and features like,

Input Fields: Include several input fields for users to enter their registration information. Common

fields include:

• Email: Provide a field to enter a valid email address.

• Password: Include a password field for users to create a secure password.

and at last login button.

Android provides an XML vocabulary for View Group and View classes, so most of your UI

isdefined in XML files. However, rather than teach you to write XML, this lesson shows you

howto create a layout using Android Studio's Layout Editor. The Layout Editor writes the

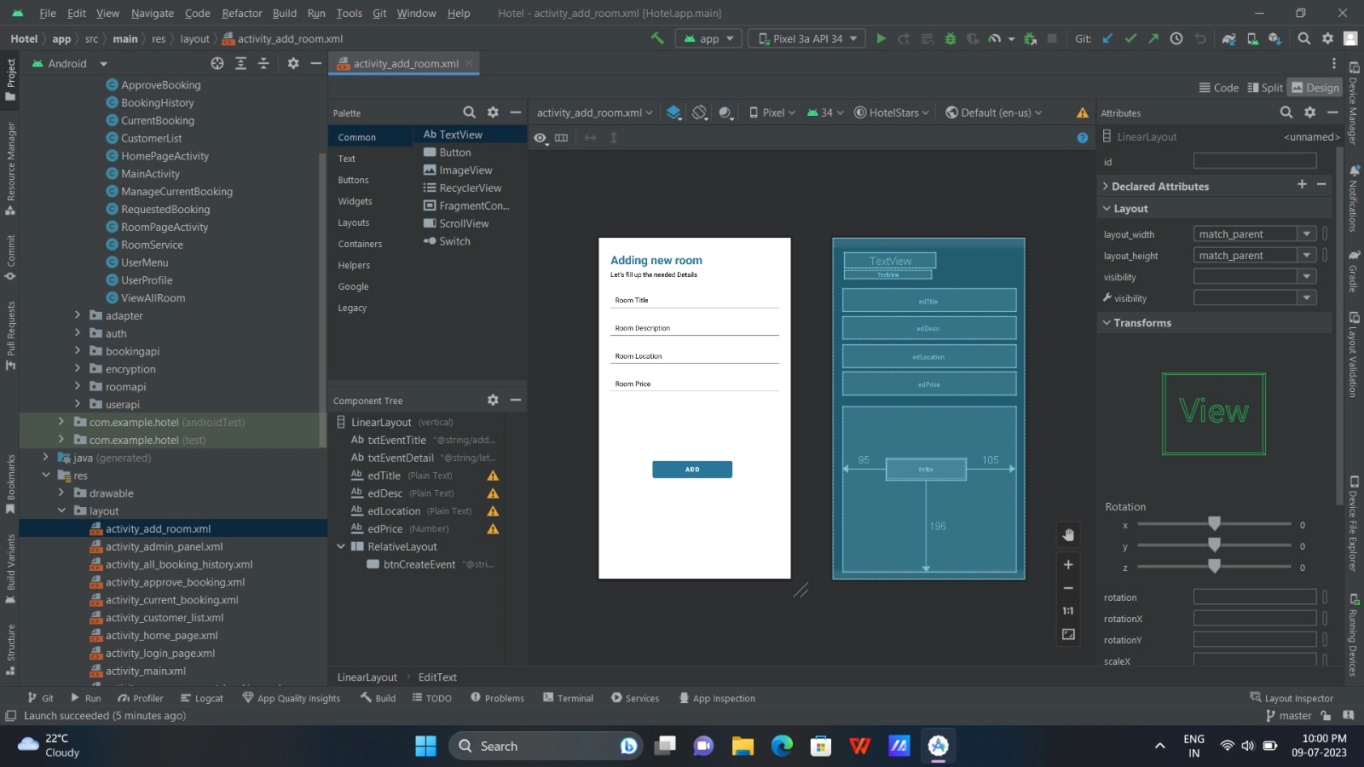
XML for you as you drag and drop views to build



**Fig no 5: User interface**

**3.2 LAYOUT EDITOR:**

1. In the Project window, open app > res > layout > activity\_main.xml.
2. To make room for the Layout Editor, hide the Project window. To do so, select View > Tool Windows > Project, or just click Project on the left side of the Android Studio screen.
3. If your editor shows the XML source, click the Design tab at the top right of the window.
4. Click  (Select Design Surface) and select Blueprint.
5. Click  (View Options) in the Layout Editor Toolbar and make sure that Show All Constraints is checked.
6. Make sure Auto connect is off. A tooltip in the toolbar displays  (Enable Auto connection to Parent) when Auto connects is off.
7. Click  (Default Margins) in the toolbar and select. If needed, you can adjust the margins for each view later.
8. Click  (Device for Preview) in the toolbar and select 5.5, 1440 × 2560, 560 dpi (Pixel XL).



**Fig no 6: Layout Editor**

**3.2.1 ADD A TEXTBOX:**

First, you need to remove what's already in the layout. Click Text View in the Component Tree panel and then press the Delete key.

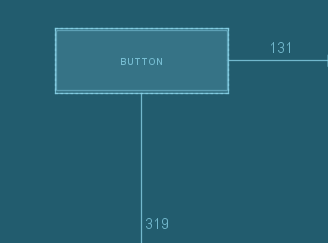


**Fig no 7: Add a textbox.**

1. In the Palette panel, click Text to show the available text controls.
2. Drag the Plain Text into the design editor and drop it near the top of the layout. This is an [Edit Text](https://developer.android.com/reference/android/widget/EditText) widget that accepts plain text input.

**3.2.2 ADD A BUTTON**

1. In the Palette panel, click Buttons.
2. Drag the Button widget into the design editor and drop it near the right side.
3. Create a constraint from the left side of the button to the right side of the text box.



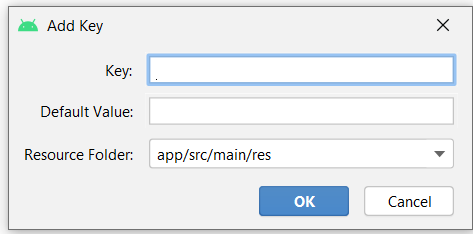
**Fig no 8: Add a Button**

**3.2.3 CHANGE UI STRING**

1. Open the Project window and then open app > res > values > strings.xml.
2. This is a [string resources](https://developer.android.com/guide/topics/resources/string-resource) file, where you can specify all of your UI strings. It allows you to manage all of your UI strings in a single location, which makes them easier to find, update, and localize.
3. Click Open editor at the top of the window. This opens the [Translations Editor,](https://developer.android.com/studio/write/translations-editor) which provides a simple interface to add and edit your default strings. It also helps you keep all of your translated strings organized.

a.Click  (Add Key) to create a new string as the "hint text" for the text box. At this point, the window showed in figure 7 opens. a. Enter "edit message" in the Key field.

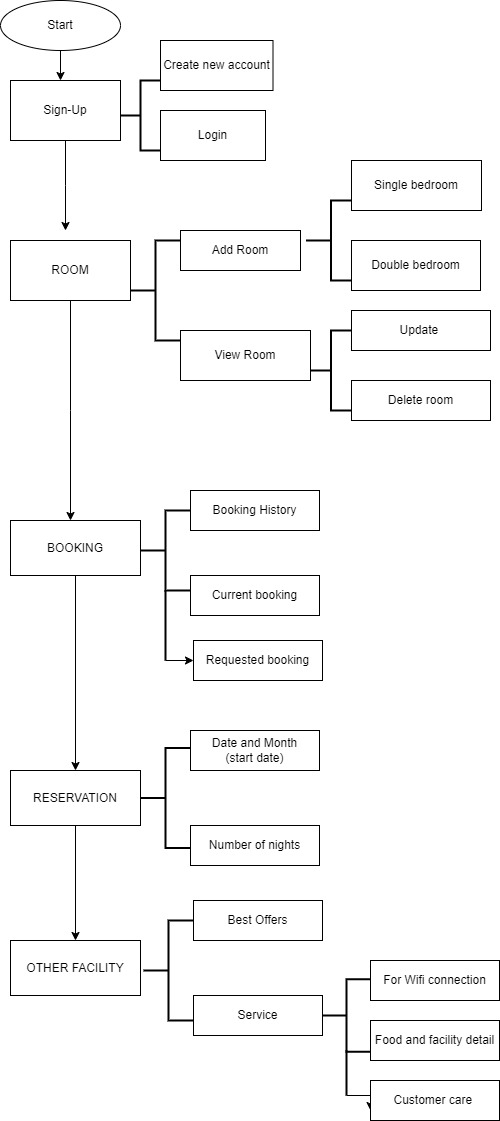
b.Enter "Enter a message" in the Default Value field.

c. click ok.

**Fig no 9: Change UI String**

**3.2 IMPLEMENTATION**

**3.2.1 Flow Chart**



**3.2.2 About the Functions used in Java**

* + - **void onCreate(Bundle savedInstanceState)**

Android provides another elegant way of achieving this. To achieve this, we have to override a method called onSaveInstanceState(). Android platform allows the users to save any instance state. Instance state can be saved in the Bundle. Bundle is passed as argument to the onSaveInstanceState method.

We can load the saved instance state from the Bundle passed as argument to the onCreate method. We can also load the saved instance state in onRestoreInstanceState method. But I will leave that for the readers to figure out.

* + - **void basicNotify(View view)**

View is one of the most general class in Android. It holds references to single piece of UI. See link above. Passing View view as argument in methods in most cases gives you opportunity to call method associated with this view. For example if you click on some elements, view is passed in listener so you can know which view was clicked and what attributes it have. Please read Android documentation.

* + - **void useAppContext()**

useContext is a hook that allows you to access and consume a given Context in a React app.

* + - **void addition\_isCorrect():**

checks for correct addition done by the particular code

* + - **void headsUpNotify(View view)**

There are multiple ways of how the notification should appear on screen and leave the screen. A set of default animator helper classes are provided and can be switched out by overriding config\_headsUpNotificationAnimationHelper.

* + - **expandableNotify(View view)**

enabling the user to get more data out of the notification system. This new ability is called expanded notification, watch the short tutorial below to know more about hoe to expand / collapse your notifications.

* + - **createNotificationChannel(String CHANNEL)**

Channel importance affects the interruption level of all notifications posted in the channel, and you must specify it in the NotificationChannel constructor. Creating an existing notification channel with its original values performs no operation, so it's safe to call this code when starting an app.

By default, all notifications posted to this channel use the visual and auditory behaviors defined by the importance level from the NotificationManagerCompat class

**3.3.3 Source code**

**XML Code:**

|  |  |
| --- | --- |
| |  | | --- | |  | |
| <?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:layout\_width="match\_parent"  android:layout\_height="match\_parent"  android:background="@color/white"  tools:context=".activity.MainActivity">  <com.google.android.material.imageview.ShapeableImageView  android:id="@+id/logo"  android:layout\_width="200dp"  android:layout\_height="200dp"  android:layout\_marginLeft="20dp"  android:layout\_marginTop="60dp"  android:src="@mipmap/ic\_launcher\_foreground"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.473"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toTopOf="parent" />  <com.google.android.material.textview.MaterialTextView  android:id="@+id/text\_about"  android:layout\_width="match\_parent"  android:layout\_height="wrap\_content"  android:layout\_marginTop="50dp"  android:layout\_marginStart="20dp"  android:layout\_marginEnd="30dp"  android:text="@string/just\_get\_your\_luggage\_ready\_your\_fantastic\_room\_is\_ready"  android:textAlignment="center"  android:textStyle="normal"  android:textColor="@color/black"  android:textSize="18sp"  app:layout\_constraintTop\_toBottomOf="@+id/logo" />  <com.google.android.material.button.MaterialButton  android:id="@+id/login\_button"  android:layout\_width="200dp"  android:layout\_height="wrap\_content"  android:layout\_marginTop="30dp"  android:background="@drawable/background"  android:drawableEnd="@drawable/smile"  android:drawableTint="@color/white"  android:text="@string/login"  android:textStyle="bold"  app:backgroundTint="@color/blue"  android:textColor="@color/white"  android:onClick="onLoginActivity"  app:layout\_constraintBottom\_toTopOf="@id/register"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintHorizontal\_bias="0.498"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/text\_about"  app:layout\_constraintVertical\_bias="0.287" />  <com.google.android.material.textview.MaterialTextView  android:id="@+id/register"  android:layout\_width="wrap\_content"  android:layout\_height="40dp"  android:text="@string/newaccount"  android:textSize="18sp"  android:onClick="onRegisterActivity"  android:textColor="@color/blue"  app:layout\_constraintBottom\_toBottomOf="parent"  app:layout\_constraintEnd\_toEndOf="parent"  app:layout\_constraintStart\_toStartOf="parent"  app:layout\_constraintTop\_toBottomOf="@+id/login\_button"  app:layout\_constraintVertical\_bias="0.347" />  </androidx.constraintlayout.widget.ConstraintLayout>  **activity\_admin\_panel.xml :** |

<?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:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".activity.AdminPanel"

android:background="@color/white">

<!--Top Bar-->

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:layout\_constraintTop\_toTopOf="parent"

android:padding="16dp">

<ImageView

android:layout\_width="60dp"

android:layout\_height="60dp"

android:layout\_alignParentStart="true"

android:background="@drawable/grey\_circular\_border"

android:onClick="onLogoutClick"

android:padding="12dp"

android:src="@drawable/logout" />

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_centerInParent="true"

android:gravity="center"

android:orientation="vertical">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/app\_name"

android:textColor="@color/blue"

android:textStyle="bold"

android:textSize="22sp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="4dp"

android:text="@string/lets\_do\_some\_management"

android:textColor="@color/black"

android:textSize="15sp" />

</LinearLayout>

</RelativeLayout>

<GridLayout

android:id="@+id/gridLayout"

android:layout\_width="match\_parent"

android:layout\_height="550dp"

android:layout\_marginBottom="20dp"

android:alignmentMode="alignMargins"

android:columnCount="2"

android:columnOrderPreserved="false"

android:padding="10dp"

android:rowCount="3"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintVertical\_bias="0.0"

app:layout\_editor\_absoluteX="0dp">

<androidx.cardview.widget.CardView

android:layout\_width="0dp"

android:layout\_height="0dp"

android:layout\_rowWeight="1"

android:layout\_columnWeight="1"

android:layout\_marginLeft="16dp"

android:layout\_marginRight="16dp"

android:layout\_marginBottom="16dp"

android:backgroundTint="#E3E2E7"

app:cardCornerRadius="15dp"

app:cardElevation="6dp">

<LinearLayout

android:onClick="add\_roomButton"

android:layout\_width="130dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal|center\_vertical"

android:layout\_margin="16dp"

android:background="#E3E2E7"

android:gravity="center"

android:orientation="vertical">

<ImageView

android:layout\_width="60dp"

android:layout\_height="60dp"

android:layout\_gravity="center\_horizontal"

android:background="@drawable/add"

android:backgroundTint="@color/blue"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"

android:text="@string/add\_room"

android:textColor="@color/blue"

android:textAlignment="center"

android:textSize="16sp"

android:textStyle="bold" />

</LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:layout\_width="0dp"

android:layout\_height="0dp"

android:layout\_rowWeight="1"

android:layout\_columnWeight="1"

android:layout\_marginLeft="16dp"

android:layout\_marginRight="16dp"

android:layout\_marginBottom="16dp"

android:backgroundTint="#E3E2E7"

app:cardCornerRadius="15dp"

app:cardElevation="6dp">

<LinearLayout

android:onClick="view\_roomsButton"

android:layout\_width="130dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal|center\_vertical"

android:layout\_margin="16dp"

android:background="#E3E2E7"

android:gravity="center"

android:orientation="vertical">

<ImageView

android:layout\_width="60dp"

android:layout\_height="60dp"

android:layout\_gravity="center\_horizontal"

android:background="@drawable/room"

android:backgroundTint="@color/blue"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"

android:text="@string/view\_rooms"

android:textColor="@color/blue"

android:textAlignment="center"

android:textSize="16sp"

android:textStyle="bold" />

</LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:layout\_width="0dp"

android:layout\_height="0dp"

android:layout\_rowWeight="1"

android:layout\_columnWeight="1"

android:layout\_marginLeft="16dp"

android:layout\_marginRight="16dp"

android:layout\_marginBottom="16dp"

android:backgroundTint="#E3E2E7"

app:cardCornerRadius="15dp"

app:cardElevation="6dp">

<LinearLayout

android:onClick="requested\_bookingButton"

android:layout\_width="130dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal|center\_vertical"

android:layout\_margin="16dp"

android:background="#E3E2E7"

android:gravity="center"

android:orientation="vertical">

<ImageView

android:layout\_width="60dp"

android:layout\_height="60dp"

android:layout\_gravity="center\_horizontal"

android:background="@drawable/send"

android:backgroundTint="@color/blue"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"

android:text="@string/requested\_booking"

android:textColor="@color/blue"

android:textAlignment="center"

android:textSize="16sp"

android:textStyle="bold" />

</LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:layout\_width="0dp"

android:layout\_height="0dp"

android:layout\_rowWeight="1"

android:layout\_columnWeight="1"

android:layout\_marginLeft="16dp"

android:layout\_marginRight="16dp"

android:layout\_marginBottom="16dp"

android:backgroundTint="#E3E2E7"

app:cardCornerRadius="8dp"

app:cardElevation="6dp">

<LinearLayout

android:onClick="current\_bookingButton"

android:layout\_width="130dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal|center\_vertical"

android:layout\_margin="16dp"

android:background="#E3E2E7"

android:gravity="center"

android:orientation="vertical">

<ImageView

android:layout\_width="60dp"

android:layout\_height="60dp"

android:layout\_gravity="center\_horizontal"

android:background="@drawable/room"

android:backgroundTint="@color/blue"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"

android:text="@string/current\_booking"

android:textAlignment="center"

android:textColor="@color/blue"

android:layout\_gravity="center\_horizontal"

android:textSize="16sp"

android:textStyle="bold" />

</LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:layout\_width="0dp"

android:layout\_height="0dp"

android:layout\_rowWeight="1"

android:layout\_columnWeight="1"

android:layout\_marginLeft="16dp"

android:layout\_marginRight="16dp"

android:layout\_marginBottom="16dp"

android:backgroundTint="#E3E2E7"

app:cardCornerRadius="8dp"

app:cardElevation="6dp">

<LinearLayout

android:onClick="booking\_historyButton"

android:layout\_width="130dp"

android:layout\_height="wrap\_content"

android:layout\_gravity="center\_horizontal|center\_vertical"

android:layout\_margin="16dp"

android:gravity="center"

android:background="#E3E2E7"

android:orientation="vertical">

<ImageView

android:layout\_width="60dp"

android:layout\_height="60dp"

android:layout\_gravity="center\_horizontal"

android:background="@drawable/history"

android:backgroundTint="@color/blue"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"

android:text="@string/booking\_history"

android:textColor="@color/blue"

android:textAlignment="center"

android:textSize="16sp"

android:textStyle="bold" />

</LinearLayout>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:layout\_width="0dp"

android:layout\_height="0dp"

android:layout\_rowWeight="1"

android:layout\_columnWeight="1"

android:layout\_marginLeft="16dp"

android:layout\_marginRight="16dp"

android:layout\_marginBottom="16dp"

android:backgroundTint="#E3E2E7"

app:cardCornerRadius="8dp"

app:cardElevation="6dp">

<LinearLayout

android:onClick="view\_customersButton"

android:layout\_width="130dp"

android:layout\_height="100dp"

android:layout\_gravity="center\_horizontal|center\_vertical"

android:layout\_margin="16dp"

android:gravity="center"

android:background="#E3E2E7"

android:orientation="vertical">

<ImageView

android:layout\_width="60dp"

android:layout\_height="60dp"

android:layout\_gravity="center\_horizontal"

android:background="@drawable/person"

android:backgroundTint="@color/blue"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="8dp"

android:text="@string/view\_customers"

android:textSize="16sp"

android:textAlignment="center"

android:textColor="@color/blue"

android:textStyle="bold" />

</LinearLayout>

</androidx.cardview.widget.CardView>

</GridLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

**JAVA Code:**

**AddRoomActivity.java**

package com.example.hotel.activity;

import androidx.activity.result.ActivityResultLauncher;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import androidx.core.content.ContextCompat;

import android.Manifest;

import android.annotation.SuppressLint;

import android.app.ProgressDialog;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.net.Uri;

import android.os.Bundle;

import android.text.TextUtils;

import android.util.Log;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.Toast;

import com.example.hotel.R;

import com.example.hotel.databinding.ActivityAddRoomBinding;

import com.example.hotel.roomapi.RoomUploadData;

import com.example.hotel.roomapi.RoomViewMessage;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.firestore.FirebaseFirestore;

import com.google.firebase.firestore.SetOptions;

import com.google.firebase.storage.FirebaseStorage;

import com.google.firebase.storage.OnProgressListener;

import com.google.firebase.storage.StorageReference;

import com.google.firebase.storage.StorageTask;

import com.google.firebase.storage.UploadTask;

import com.squareup.picasso.Picasso;

import java.sql.Timestamp;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Locale;

import java.util.Objects;

import javax.net.ssl.SSLSessionBindingEvent;

import com.google.firebase.FirebaseApp;

@SuppressWarnings("deprecation")

public class AddRoomActivity extends AppCompatActivity implements RoomViewMessage {

ActivityAddRoomBinding binding;

private StorageReference storageReference;

ProgressDialog progressDialog;

EditText edTitle, edDesc, edLocation, edPrice;

RoomUploadData roomUploadData;

private StorageTask uploadtask;

Button add;

private static final String TAG = "addRoom";

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

Objects.requireNonNull(getSupportActionBar()).hide(); //This Line hides the action bar

binding = ActivityAddRoomBinding.inflate(getLayoutInflater());

setContentView(binding.getRoot());

add = binding.btnCreateEvent;

edTitle = binding.edTitle;

edDesc = binding.edDesc;

edLocation = binding.edLocation;

edPrice = binding.edPrice;

roomUploadData = new RoomUploadData(this);

storageReference = FirebaseStorage.getInstance().getReference("RoomImages");

progressDialog = new ProgressDialog(this);

progressDialog.setTitle("Uploading file...");

add.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View v) {

checkSignUpDetails();

}

});

}

private void checkSignUpDetails() {

String isAvailable = "yes";

String title = edTitle.getText().toString().trim();

String description = edDesc.getText().toString().trim();

String location = edLocation.getText().toString().trim();

int price =Integer.parseInt(edPrice.getText().toString().trim());

Timestamp timestamp = new Timestamp(System.currentTimeMillis());

String id = timestamp.toString().trim();

if(!TextUtils.isEmpty(title) && !TextUtils.isEmpty(description)){

roomUploadData.onSuccessUpdate(this,id,title,description,isAvailable,location,null, price);

}else{

if(TextUtils.isEmpty(title)){

edTitle.setError("Title is required");

}if (TextUtils.isEmpty(description)){

edDesc.setError("Description is required");

}

if (TextUtils.isEmpty(location)){

edLocation.setError("Location is required");

}

if (TextUtils.isEmpty(edPrice.getText())) {

edPrice.setError("Price is required");

}

}

}

@Override

public void onUpdateFailure(String message) {

Toast.makeText(AddRoomActivity.this, message, Toast.LENGTH\_SHORT).show();

Intent intent = new Intent(AddRoomActivity.this, AdminPanel.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

finish();

}

@Override

public void onUpdateSuccess(String message) {

Toast.makeText(AddRoomActivity.this, message, Toast.LENGTH\_SHORT).show();

Intent intent = new Intent(AddRoomActivity.this, AdminPanel.class);

intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

finish();

}

}

**AdminManageRoom.java**

package com.example.hotel.activity;

import android.annotation.SuppressLint;

import android.app.Activity;

import android.content.Intent;

import android.os.Bundle;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.Nullable;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import com.example.hotel.R;

import com.example.hotel.adapter.ManageRoomsAdapter;

import com.example.hotel.roomapi.RoomFetchData;

import com.example.hotel.roomapi.RoomModel;

import com.example.hotel.roomapi.RoomViewFetchMessage;

import java.util.ArrayList;

public class AdminManageRoom extends Activity implements RoomViewFetchMessage {

private RecyclerView ListDataView;

private ManageRoomsAdapter manageRoomsAdapter;

ArrayList<RoomModel> roomModelArrayList = new ArrayList<>();

private TextView title;

private RoomFetchData roomFetchData;

@SuppressLint("SetTextI18n")

@Override

protected void onCreate(@Nullable Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.listview\_activity);

ListDataView = findViewById(R.id.ListViewRoom);

title = findViewById(R.id.pageTitle);

title.setText("Manage Room Record");

roomFetchData = new RoomFetchData(this,this);

RecyclerViewMethods();

roomFetchData.onSuccessUpdate(this);

}

public void RecyclerViewMethods() {

LinearLayoutManager manager = new LinearLayoutManager(this);

ListDataView.setLayoutManager(manager);

ListDataView.setHasFixedSize(true);

manageRoomsAdapter = new ManageRoomsAdapter(this, roomModelArrayList);

ListDataView.setAdapter(manageRoomsAdapter);

ListDataView.invalidate();

}

@Override

public void onUpdateSuccess(RoomModel message) {

if(message != null){

RoomModel roomModel = new RoomModel(message.getId(),message.getTitle(),message.getDescription(),message.getIsAvailable(),message.getLocation(),

message.getImageUrl(),message.getPrice());

roomModelArrayList.add(roomModel);

}

manageRoomsAdapter.notifyDataSetChanged();

}

@Override

public void onUpdateFailure(String message) {

Toast.makeText(AdminManageRoom.this, message, Toast.LENGTH\_LONG).show();

}

@Override

public void onBackPressed() {

super.onBackPressed();

Intent intent = new Intent(AdminManageRoom.this, AdminPanel.class);

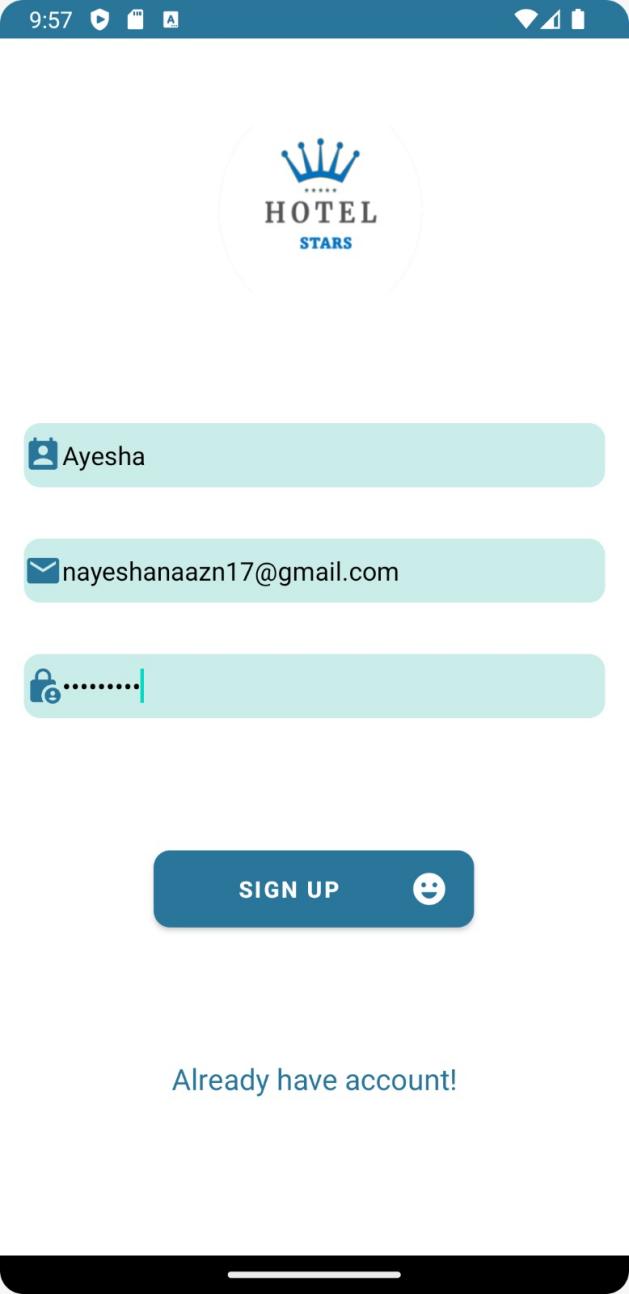
intent.addFlags(Intent.FLAG\_ACTIVITY\_CLEAR\_TASK | Intent.FLAG\_ACTIVITY\_NEW\_TASK);

startActivity(intent);

finish(); }}

**Chapter 4**

**SNAPSHOTS**

****

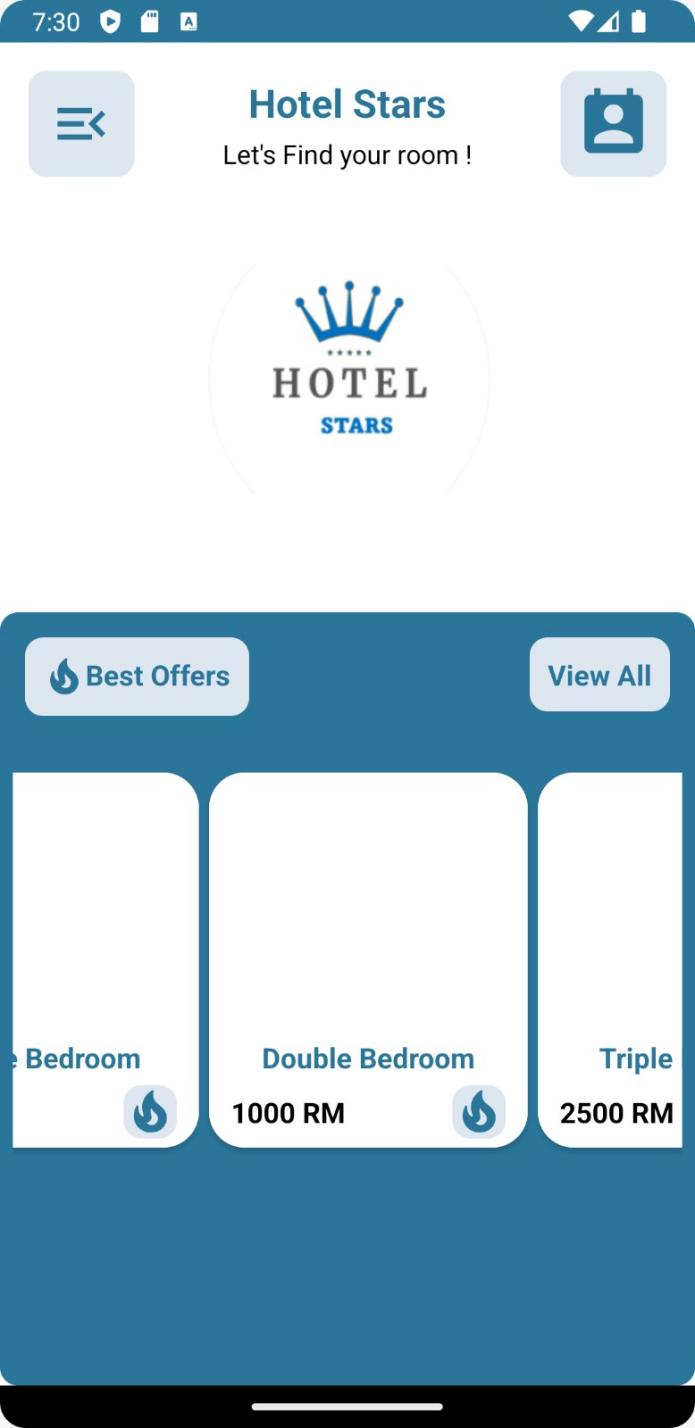
**Figure 6.1 REGISTRATION PAGE**

This figure shows the registration page for both user and admin



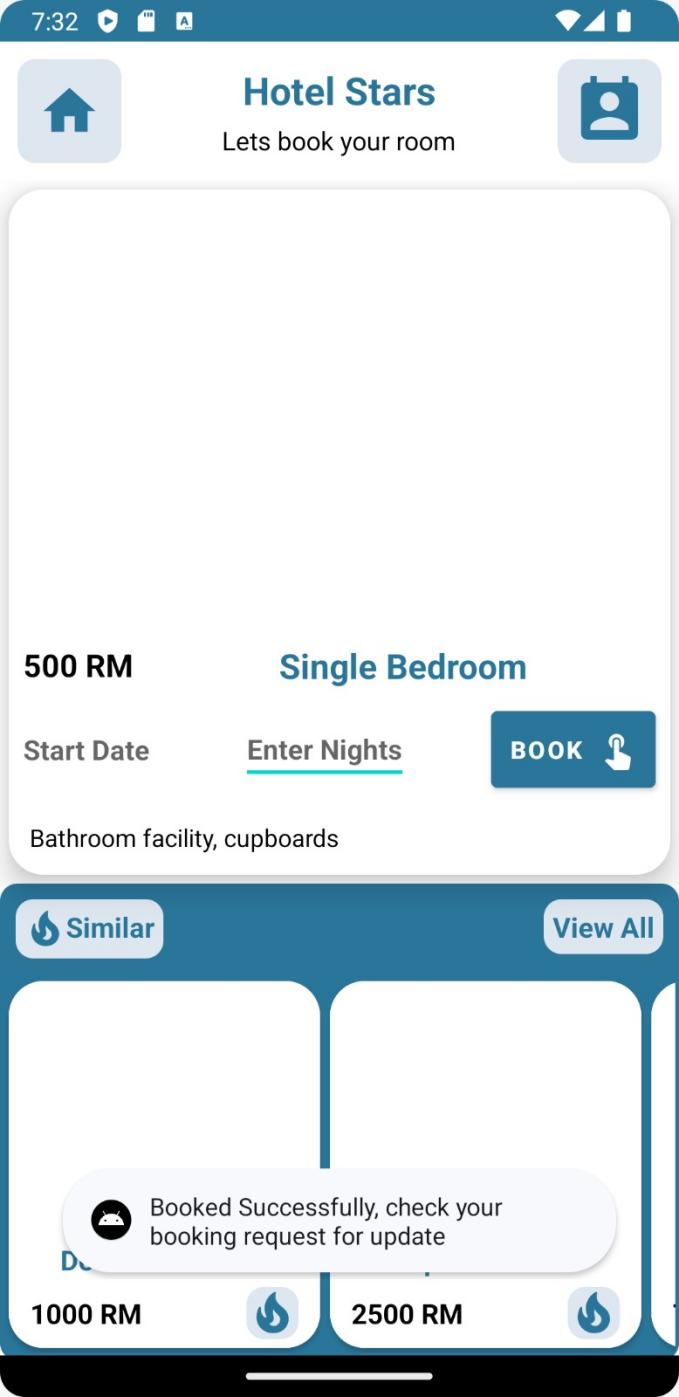
**Figure 6.2 LOGIN PAGE**

This figure shows the login page for both user and admin



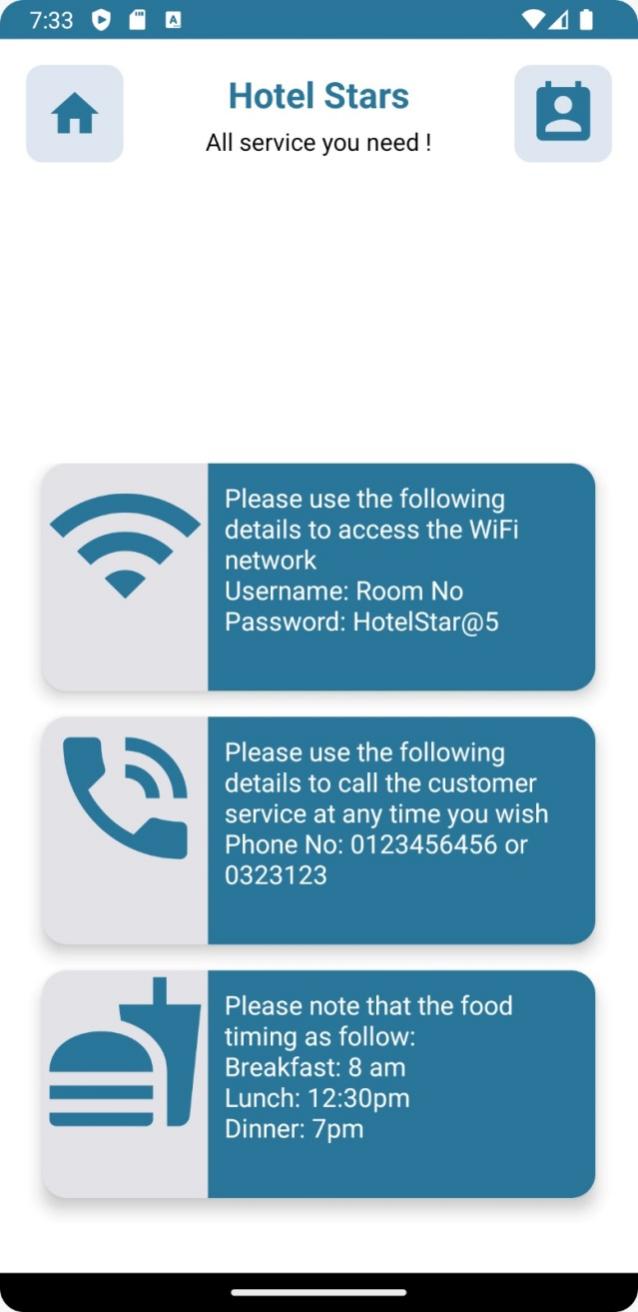
**Figure 6.3 HOME PAGE**

This figure shows the home page for user



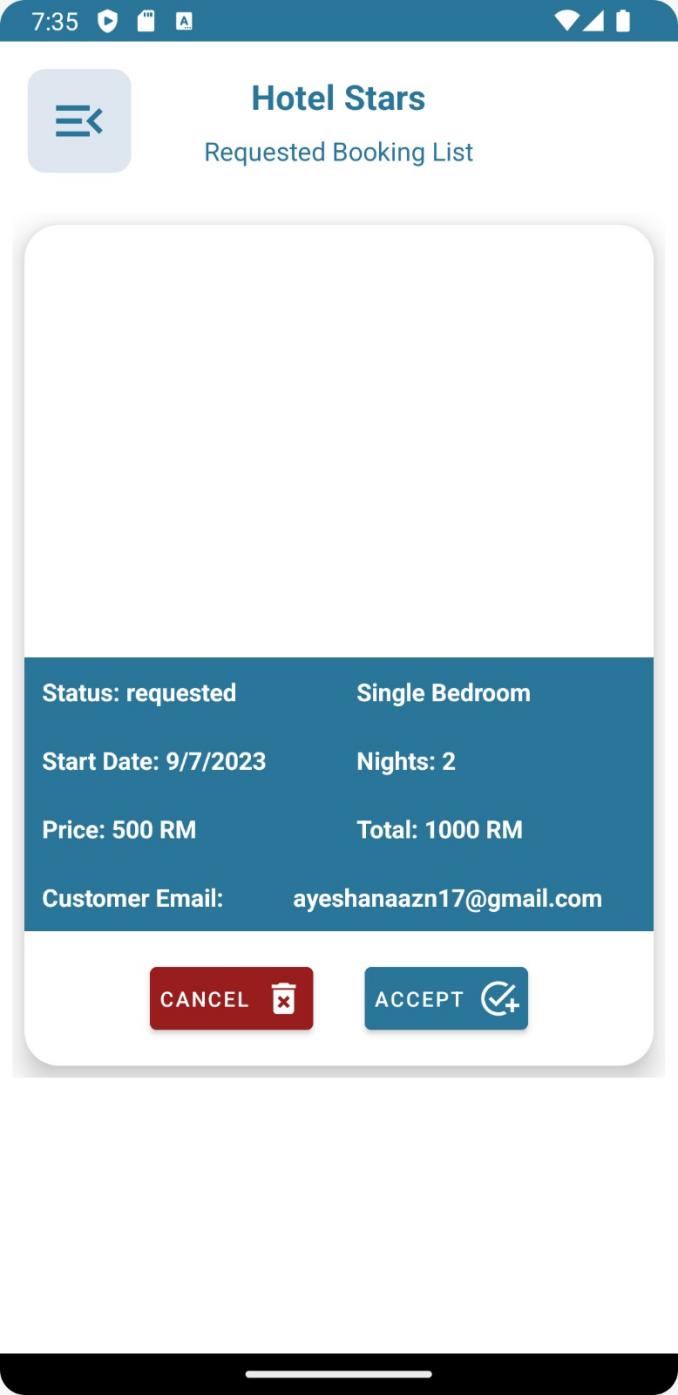
**Figure 6.4 BOOKING ROOM**

This figure shows the Booking room page for user.



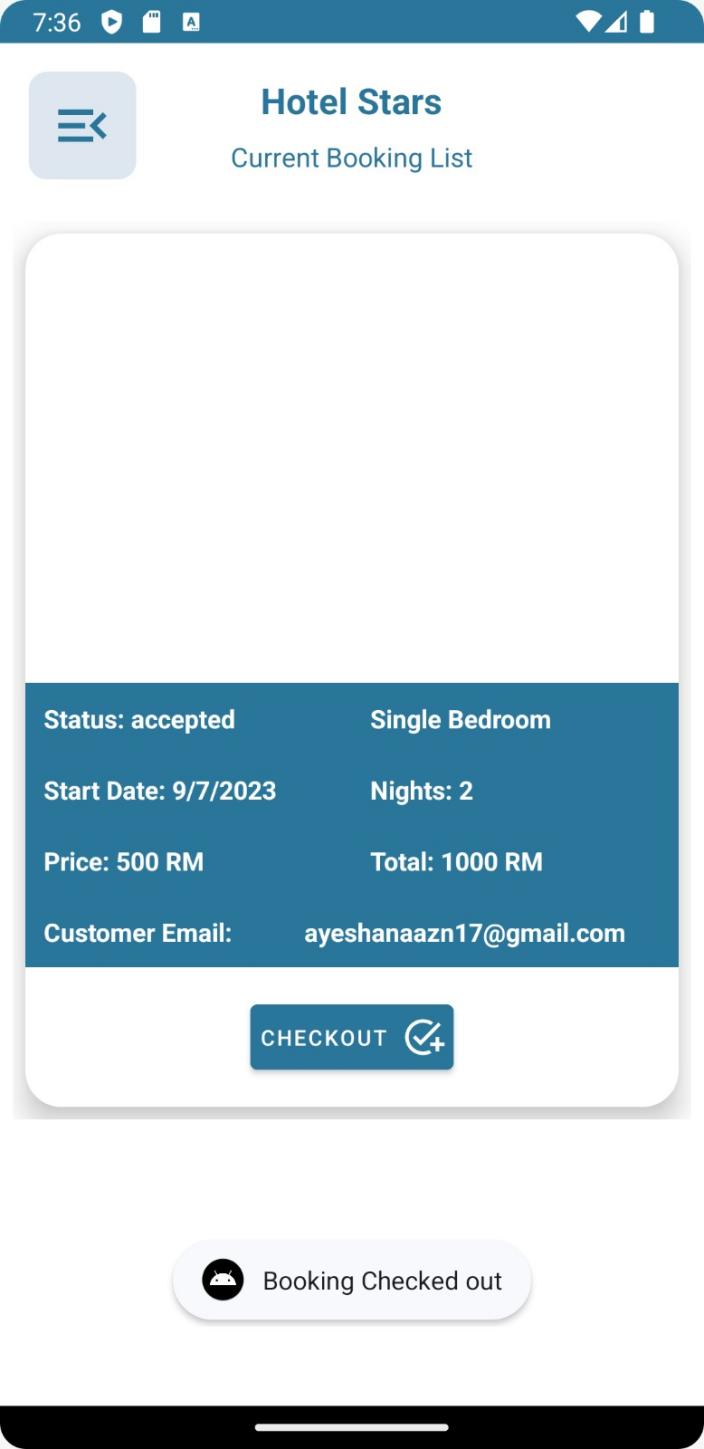
**Figure 6.5 SERVICES AVAILABLE**

This figure shows the room service available for user.



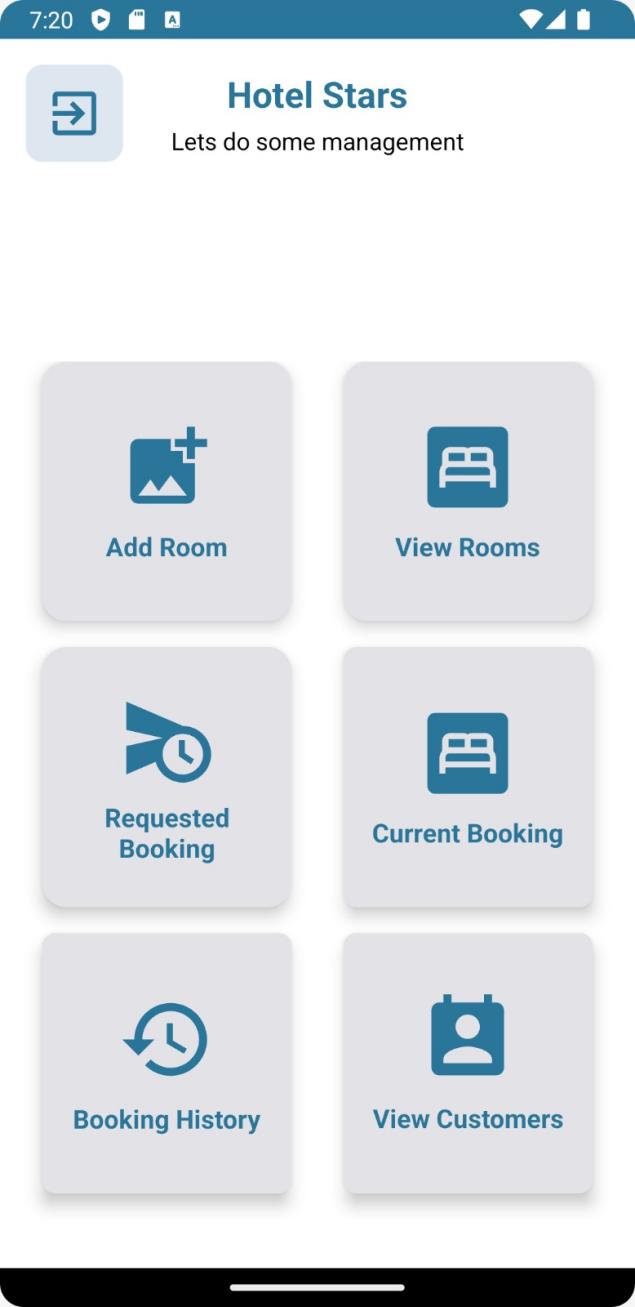
**Figure 6.6 BOOKING HISTORY**

This figure shows the booking history of the user



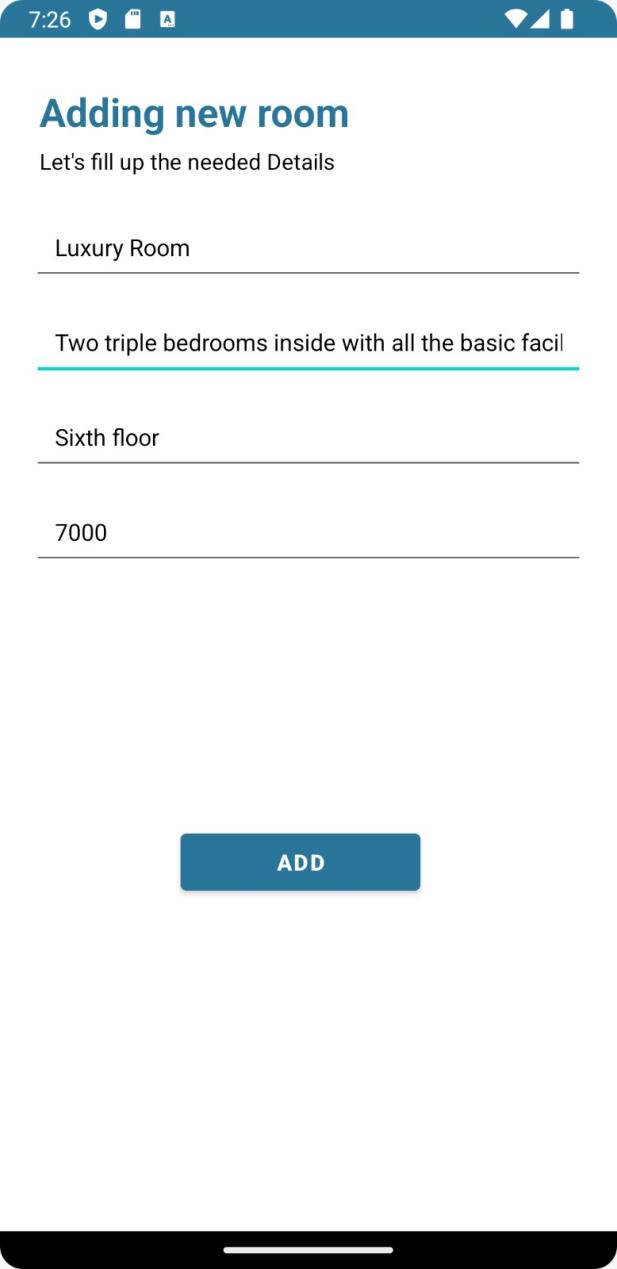
**Figure 6.5 CURRENT BOOKING LIST**

This figure shows the booking list of the user



**Figure 6.8 MANAGING SERVICE**

This figure shows the managing services list for the user

****

**Figure 6.5 ADDING NEW ROOM**

This figure shows the managing services list for the user

**Chapter 5**

**CONCLUSION AND FUTURE SCOPE**

In conclusion, the hotel booking system app provides a seamless and efficient platform for users to book rooms while offering the administrative team the ability to manage and add new rooms. With its user-friendly interface and streamlined process, the app simplifies the booking experience for users, allowing them to easily browse available rooms, select their preferred options, and make reservations. Meanwhile, the app's administrative features empower hotel staff to effortlessly add new rooms to the system, ensuring that the available inventory remains up-to-date. Overall, this hotel booking system app enhances both the user and administrative experiences, facilitating smooth and hassle-free room bookings in the hospitality industry.

.

**Future Scope:**

The future of the hotel booking system app holds exciting possibilities for both users and administrators. Key developments include personalized recommendations based on user preferences, integrated services like restaurant reservations and local activities, virtual reality and augmented reality features for immersive room exploration, smart assistant integration for voice-controlled bookings, blockchain technology for enhanced security and transparency, and sustainability initiatives to promote eco-friendly choices. These advancements aim to provide a more seamless and tailored booking experience while streamlining administrative tasks in the hospitality industry.

**REFERENCES**

1. P. Jain, "What Makes Java a Powerful Programming Language," 11 February 2013.
2. A. Rongala, "Benefits of Java over Other Programming Languages," 7 May 2015.
3. "Java SE Downloads,"
4. www.youtube
5. Github
6. "Android Developers,"