

**Visvesvaraya Technological University
Belagavi, Karnataka-590 018**



**A MINI PROJECT REPORT
On**

PROMOPAL

Submitted
in partial fulfilment requirements for the credit of the Course
on
MOBILE APPLICATION DEVELOPMENT
18CSMP68
by

DARSHAN N SHETTY 4CB20CS028

Under the Guidance of
Mrs. Aruna Kumari G K
Assistant Professor



**Department of Computer Science and Engineering
Canara Engineering College, Benjanapadavu
2022-2023**

CANARA ENGINEERING COLLEGE

BENJANAPADAVU, BANTWAL- 574219 – KARNATAKA

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



CERTIFICATE

This is to certify that **DARSHAN N SHETTY (4CB20CS028)** have successfully completed the mini project work on ‘**PromoPal**’ and submitted in partial fulfillment of the requirements of the Course on **Mobile Application Development(18CSMP68)** prescribed by the **Visvesvaraya Technological University** during the academic year **2022-2023**. It is verified that all corrections/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 mini project work prescribed by Bachelor of Engineering Degree.

Course Instructor
Mrs. Aruna Kumari G K
Assistant Professor

H.O.D
Dr. Sunil Kumar B L
Associate Professor

Name of the Examiners

Signature with Date

1.

2.

ACKNOWLEDGEMENT

I am indebted to our **Principal, Dr. Ganesh V. Bhat** and management of **Canara Engineering College** for providing an environment with all facilities that helped us in completing our mini project.

I am extremely grateful to **Dr. Sunil Kumar B L, Head of Computer Science & Engineering Department** for his moral support and encouragement.

I wish to express our sincere gratitude to our Course Instructor, **Mrs. Aruna Kumari G K, Assistant Professor, Computer Science & Engineering Department**, for the guidance and suggestions.

I thank all the teaching and non-teaching staff of the department for their kind help in completing this mini project.

Last but not the least; I would like to add some personal note. If there is a driving force that kept us going, and what has not changed it is the constant support and blessing of our parents, family and friends. There is no doubt, in spite of my strenuous efforts error might remain in mini project. Naturally, I alone take full responsibility for any lack of clarity, occasional erratum or inexactness that may occur.

DARSHAN N SHETTY 4CB20CS028

ABSTRACT

The "PromoPal" Android app, developed using Android Studio, is a self-promotion application designed to simplify the process of sending promotional messages via SMS and email. The app starts with a captivating splash screen that provides a concise description of its purpose and functionalities. On the home screen, users are presented with two distinct cards labeled "SMS" and "Email," representing the messaging options. Selecting either card directs users to the corresponding page where they input recipient details and create compelling promotional messages. The SMS page prompts users to enter the mobile number and other relevant information, while the Email page requires email address and message details. Both pages offer customization options like selecting names, specifying positions, and immediate start. After completing the forms, users can preview the auto-generated messages for accuracy. They then have the flexibility to choose between sending the message via SMS or email based on their desired communication channel. The PromoPal app streamlines self-promotion by providing customizable templates, easy input forms, and message previews, enhancing the user's promotional messaging experience.

TABLE OF CONTENTS

CONTENTS	PAGE NO.
1. INTRODUCTION	
1.1 Introduction to Android Application	01
1.2 Introduction to Android Studio	01
1.3 Project Description	02
2. REQUIREMENT SPECIFICATION	
2.1 Hardware Requirements	03
2.2 Software Requirements	03
3. DESIGN	
3.1 Architectural Diagram/Flowchart	04
3.2 XML Code	05
4. IMPLEMENTATION DETAILS	
4.1 List of classes imported and their use in the application.	18
4.2 List of important library functions used and their descriptions.	19
4.3 Java Code with proper comment lines	20
4.4 Manifest File	35
5. RESULTS AND DISCUSSION	
5.1 Sample Output (Screenshots) with brief discussion	35
6. CONCLUSION & FUTURE WORK	38
REFERENCES	39

CHAPTER 1

INTRODUCTION

1.1 Introduction to Android Application

Android Application is a software designed to run on an Android device or emulator. The term also refers to an APK file which stands for Android package. This file is a Zip archive containing app code, resources, and meta information. Android apps can be written in Kotlin, Java, and C++ and are run inside Virtual Machine. The official development environment is Android Studio. The official development environment is Android Studio. The compiled code along with any data and resource files required by the application is bundled by the apt tool into an Android package, an archive file marked by an .apk suffix. This file is the vehicle for distributing the application and installing it on mobile devices; it's the file users download to their devices. All the code in a single .apk file is considered to be one application. By default, every application runs in its own Linux process, with a unique Linux user id. Each process has its own virtual machine (VM).

1.2 Introduction to Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps, such as:

- A flexible Gradle-based build system.
- A fast and feature-rich emulator.
- A unified environment where you can develop for all Android devices.
- Apply Changes to push code and resource changes to your running app without restarting your app.
- Code templates and GitHub integration to help you build common app features and import sample code.
- Extensive testing tools and frameworks.
- Lint tools to catch performance, usability, version compatibility, and other problems.
- C++ and NDK support.

- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

1.3 Project Description

PromoPal is a compact and user-friendly Android app developed using Android Studio, designed specifically to streamline and simplify the process of self-promotion. With PromoPal, users can effortlessly create and send customized promotional messages through both SMS and email channels, saving time and effort in reaching out to potential clients, employers, or contacts.

The app boasts an intuitive interface that guides users through each step of the promotional message creation process. Starting with a captivating splash screen, PromoPal greets users and provides a concise overview of its purpose and functionalities, setting the tone for a positive user experience. Upon transitioning to the home screen, users are presented with two distinct cards: "SMS" and "Email." These cards represent the two messaging options available within the app. By selecting either card, users are seamlessly directed to the corresponding page, tailored for composing either an SMS or an email message.

On the SMS page, users are prompted to enter the recipient's mobile number and various other details essential for creating a compelling promotional message. The form provides a range of options, allowing users to select the recruiter's name, their own name, indicate the desired position (such as "junior"), and specify an immediate start, among other pertinent information. This level of customization ensures that the generated message is not only personalized but also relevant to the recipient's needs and preferences.

Similarly, on the Email page, users are required to input the recipient's email address and relevant details for the promotional message. The form provides similar customization options, such as selecting the recruiter's name, the sender's name, specifying the position, and indicating immediate availability. By offering these customizable fields, PromoPal allows users to create engaging and tailored email messages that effectively showcase their skills, products, or services.

One of the key features of PromoPal is the ability to preview the auto-generated messages. After completing the input forms for both SMS and email, users can review and preview the composed messages, ensuring their accuracy, appropriateness, and effectiveness. This preview feature empowers users to make any necessary adjustments or refinements to the message content, optimizing their promotional efforts.

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 Hardware Requirements

- Processor: Minimum 1 GHz, or faster processor
- RAM: 4GB (recommend)

2.2 Software Requirements

- Operating System: Windows 10, Version 1909, OS Build 18363.719
- Tool/IDE (with version) used: Android Studio 4.2.2
- Emulator Used: Pixel 3a API 30
- Language used: Java

CHAPTER 3

DESIGN

3.1 Architectural Diagram of the Application

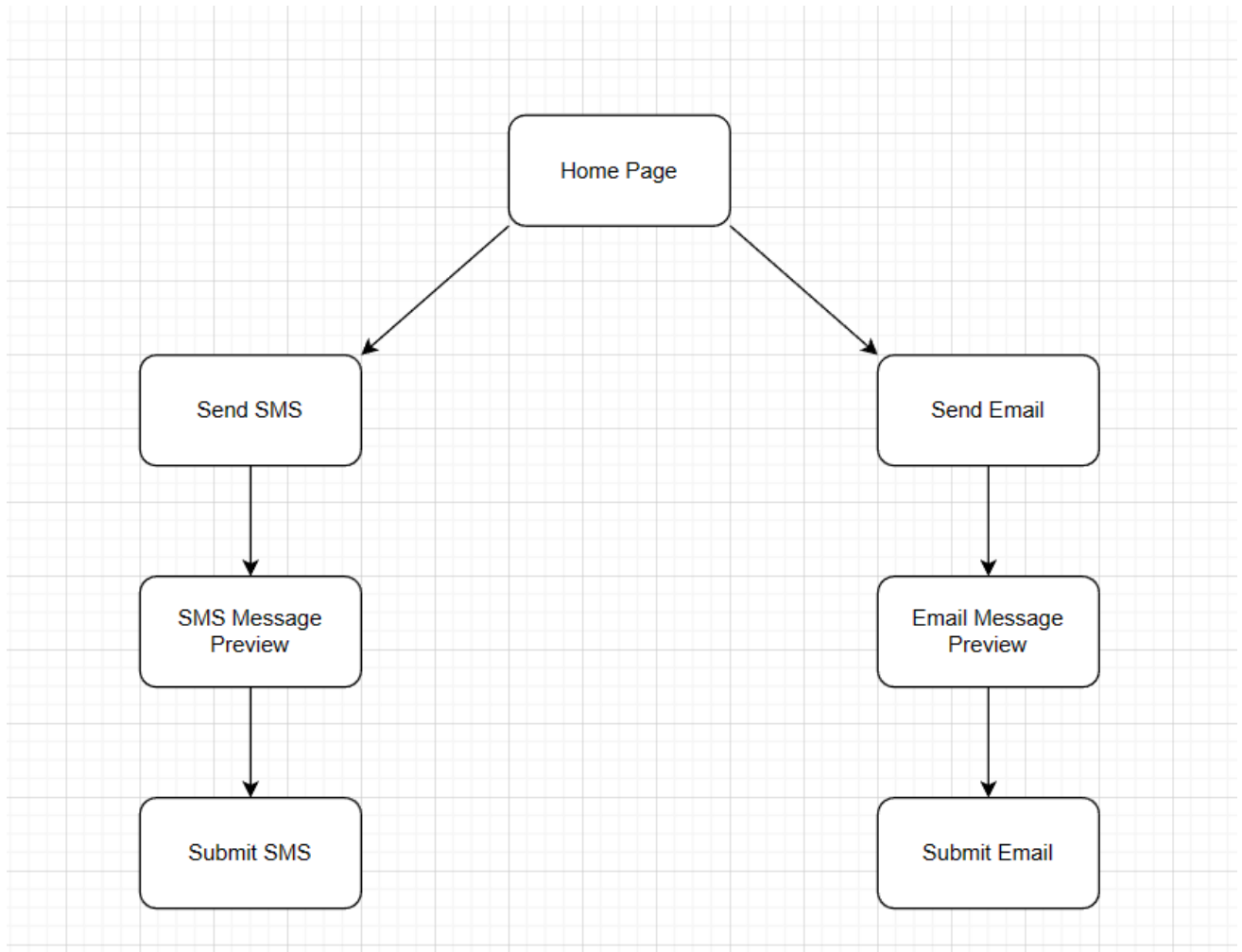


Fig 3.1: Architectural diagram

Explanation:

- Splash Screen: The app starts with a splash screen introducing PromoPal.
- Home Screen: Users see two cards labeled "SMS" and "Email" on the home screen.
- SMS Card: Clicking on the SMS card takes users to a page to enter recipient's mobile number and message details. Users can preview the auto-generated SMS message.
- Email Card: Clicking on the Email card leads users to a page to enter recipient's email address and message details. Users can preview the auto-generated email message.
- SMS Send: From the SMS page, users can send the auto-generated SMS message.
- Email Send: From the Email page, users can send the auto-generated email message.

3.2 XML Code

activity_splash_screen.xml

```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:background="#dee2e6"
    xmlns:android="http://schemas.android.com/apk/res/android" >
    <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=".Splash_Screen">
        <ImageView
            android:id="@+id/logo_id"
            android:layout_width="0dp"
            android:layout_height="400dp"
            android:layout_marginStart="10dp"
            style=""
```

android:layout_marginTop="100dp"

android:layout_marginEnd="10dp"

android:src="@drawable/promopal_high_resolution_logo_black_on_transparent_background"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toTopOf="parent" />

<TextView

android:id="@+id/info_id"

android:layout_width="match_parent"

android:layout_height="wrap_content"

android:layout_marginStart="20dp"

android:layout_marginEnd="20dp"

android:textColor="@color/black"

android:text="PromoPal simplifies professional networking by generating personalized paragraphs from your details, ready to send via SMS or email with just one click"

android:textSize="20sp"

android:layout_marginTop="30dp"

app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintStart_toStartOf="parent"

app:layout_constraintTop_toBottomOf="@id/logo_id"

tools:layout_editor_absoluteX="38dp"

tools:layout_editor_absoluteY="531dp" />

</androidx.constraintlayout.widget.ConstraintLayout>

</ScrollView>

activity_home_page.xml

<?xml version="1.0" encoding="utf-8"?>

<ScrollView 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:background="@drawable/_427030"
android:layout_height="match_parent">

```

```

<androidx.constraintlayout.widget.ConstraintLayout
    android:layout_width="match_parent"

    android:layout_height="wrap_content"
    tools:context=".HomePage">

```

```

<androidx.cardview.widget.CardView
    android:id="@+id/sms_card_view_id"
    android:layout_width="0dp"
    app:cardBackgroundColor="@color/white"
    android:layout_height="250dp"
    android:layout_marginStart="30dp"
    android:layout_marginTop="100dp"
    android:layout_marginEnd="30dp"
    android:layout_marginBottom="50dp"
    app:layout_constraintBottom_toTopOf="@id/email_card_view_id"
    app:layout_constraintEnd_toEndOf="parent"
    app:layout_constraintHorizontal_bias="1.0"
    app:layout_constraintStart_toStartOf="parent"
    app:layout_constraintTop_toTopOf="parent"
    app:layout_constraintVertical_bias="1.0"
    app:layout_goneMarginEnd="30dp">

```

```

<ImageView
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:src="@drawable/images" />

```

```

<TextView
    android:id="@+id/sms_card_heading"
    android:layout_width="match_parent"

```

```
android:layout_height="wrap_content"
```

```
android:layout_gravity="bottom"
```

```
android:gravity="center"
```

```
android:text="Send message via SMS"
```

```
android:background="@color/white"
```

```
android:textColor="#001219"
```

```
android:textSize="24sp" />
```

```
</androidx.cardview.widget.CardView>
```

```
<androidx.cardview.widget.CardView
```

```
    android:id="@+id/email_card_view_id"
```

```
    app:cardBackgroundColor="@color/white"
```

```
    android:layout_width="0dp"
```

```
    android:layout_height="250dp"
```

```
    android:layout_marginStart="30dp"
```

```
    android:layout_marginTop="16dp"
```

```
    android:layout_marginEnd="30dp"
```

```
    app:layout_constraintBottom_toBottomOf="parent"
```

```
    app:layout_constraintEnd_toEndOf="parent"
```

```
    app:layout_constraintHorizontal_bias="1.0"
```

```
    app:layout_constraintStart_toStartOf="parent"
```

```
    app:layout_constraintTop_toBottomOf="@id/sms_card_view_id"
```

```
    app:layout_constraintVertical_bias="1.0"
```

```
    app:layout_goneMarginEnd="30dp">
```

```
<ImageView
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent"
```

```
    android:src="@drawable/email_gmail_mail_service_mailing_online_service_icon_1320194987766966945" />
```

```
<TextView
```

```
    android:id="@+id/email_card_heading"
```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_gravity="bottom"
        android:gravity="center"
        android:text="Send message via EMAIL"
        android:background="@color/white"
        android:textColor="#001219"
        android:textSize="24sp"/>
    </androidx.cardview.widget.CardView>
</androidx.constraintlayout.widget.ConstraintLayout>
</ScrollView>

```

activity_sms_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView 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"
    android:background="@drawable/v915_techi_017_d">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="48dp">

        <com.google.android.material.textfield.TextInputLayout
            style="@style/outline_box_style"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:hint="@string/hint_Recruiter_name"
            android:textColorHint="@color/black">

            <com.google.android.material.textfield.TextInputEditText
                android:id="@+id/edit_text_name"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:inputType="textCapWords"
        android:textColor="@color/black"
        android:textColorHint="@color/black" />
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    style="@style/outline_box_style"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/hint_Recruiter_number"
    android:textColorHint="@color/black">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/edit_text_number"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:inputType="phone"
        android:textColor="@color/black"
        android:textColorHint="@color/black" />
</com.google.android.material.textfield.TextInputLayout>
<com.google.android.material.textfield.TextInputLayout
    style="@style/outline_box_style"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/hint_my_display_name"
    android:textColorHint="@color/black">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/edit_text_my_display_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"

```

```

        android:inputType="textCapWords"
        android:textColor="@color/black"
        android:textColorHint="@color/black" />

```

```
</com.google.android.material.textfield.TextInputLayout>
```

```
<CheckBox
```

```

    android:id="@+id/check_box_id"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:text="@string/include_junior"
    android:textColor="@color/black" />

```

```
<Spinner
```

```

    android:id="@+id/spinner_id"
    android:layout_width="match_parent"
    style="@style/spinnerstyle"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:textColor="#000000"
    tools:listitem="@android:layout/simple_list_item_1" />

```

```
<CheckBox
```

```

    android:id="@+id/immediate_start_id"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:text="@string/immediate_start"
    android:textColor="@color/black" />

```

```
<com.google.android.material.textfield.TextInputLayout
```

```

    style="@style/outline_box_style"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"

```

```

        android:hint="@string/available_start_date"
        android:textColorHint="@color/black">

        <com.google.android.material.textfield.TextInputEditText
            android:id="@+id/start_date_id"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="10dp"
            android:inputType="number|date"
            android:textColor="@color/black"
            android:textColorHint="@color/black" />
    </com.google.android.material.textfield.TextInputLayout>
    <Button
        android:id="@+id/button_id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="25dp"
        android:text="@string/preview_message" />
</LinearLayout>
</ScrollView>

```

activity_email_main.xml

```

<?xml version="1.0" encoding="utf-8"?>
<ScrollView 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"
    android:background="@drawable/rm222_mind_14">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical"
        android:padding="48dp">

```

```

<com.google.android.material.textfield.TextInputLayout
    style="@style/outline_box_style"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/hint_Recruiter_name"
    android:textColorHint="@color/black">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/edit_text_name"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:inputType="textCapWords"
        android:textColor="@color/black" />
</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    style="@style/outline_box_style"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/Recruiter_email"
    android:textColorHint="@color/black">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/edit_text_email"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:inputType="textEmailAddress"
        android:textColor="@color/black"
        android:textColorHint="@color/black"/>
</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout
    style="@style/outline_box_style"

```

```

        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="@string/hint_my_display_name"
        android:textColorHint="@color/black">
<com.google.android.material.textfield.TextInputEditText
    android:id="@+id/edit_text_my_display_name"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:inputType="textCapWords"
    android:textColor="@color/black"
    android:textColorHint="@color/black" />
</com.google.android.material.textfield.TextInputLayout>

```

```

<CheckBox
    android:id="@+id/check_box_id"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:text="@string/include_junior"
    android:textColor="@color/black" />

```

```

<Spinner
    android:id="@+id/spinner_id"
    android:layout_width="match_parent"
    style="@style/spinnerstyle"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:textColor="@color/black"
    tools:listitem="@android:layout/simple_list_item_1" />

```

```

<CheckBox
    android:id="@+id/immediate_start_id"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"

```

```

        android:layout_margin="10dp"
        android:text="@string/immediate_start"
        android:textColor="@color/black" />

<com.google.android.material.textfield.TextInputLayout
    style="@style/outline_box_style"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:hint="@string/available_start_date"
    android:textColorHint="@color/black">

    <com.google.android.material.textfield.TextInputEditText
        android:id="@+id/start_date_id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="10dp"
        android:textColor="@color/black"
        android:textColorHint="@color/black" />

</com.google.android.material.textfield.TextInputLayout>
<Button
    android:id="@+id/button_id"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="25dp"
    android:backgroundTint="#161b33"
    android:text="@string/preview_message" />
</LinearLayout>
</ScrollView>

```

activity_sms_preview.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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="@drawable/vertical_background_template_web_banner_design_your_
creative_ideas_with_space_text_7954_6596"
        android:orientation="vertical"
        android:padding="45dp">
        <androidx.cardview.widget.CardView
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            app:cardBackgroundColor="#d8e2dc"
            app:cardCornerRadius="10dp">

        <TextView
            android:id="@+id/text_view_id"
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_margin="12dp"
            android:fontFamily="sans-serif-medium"
            android:textColor="#55286f"
            android:textSize="20sp"
            tools:text="this is my sample text to test the cardview" />
        </androidx.cardview.widget.CardView>
        <Button
            android:layout_width="match_parent"
            android:layout_height="wrap_content"
            android:layout_marginTop="20dp"
            android:id="@+id/send_message_button_id"
            android:text="@string/send_message" />
    </LinearLayout>

```

activity_email_preview.xml

```

<?xml version="1.0" encoding="utf-8"?>
<LinearLayout 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="@drawable/vertical_background_template_web_banner_design_your_
creative_ideas_with_space_text_7954_6596"
        android:orientation="vertical"
        android:padding="45dp">
<androidx.cardview.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    app:cardBackgroundColor="#d8e2dc"
    app:cardCornerRadius="10dp">
    <TextView
        android:id="@+id/text_view_id"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_margin="12dp"
        android:fontFamily="sans-serif-medium"
        android:textColor="#55286f"
        android:textSize="20sp"
        tools:text="this is my sample text to test the cardview" />
    </androidx.cardview.widget.CardView>
    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginTop="20dp"
        android:id="@+id/send_message_button_id"
        android:text="@string/send_message" />
    <Button
        android:layout_width="match_parent"
        android:id="@+id/attachPdfButtonId"
        android:layout_height="wrap_content"
        android:layout_marginTop="50dp"
        android:text="Attach Resume"
    />
</LinearLayout>

```

CHAPTER 4

IMPLEMENTATION

4.1 List of classes imported and their purpose in the application.

- **AppCompatActivity** : Base class for activities that wish to use some of the newer platform features on older Android devices.
- **Intent** : An intent is an abstract description of an operation to be performed. It can be used to navigate from one activity to another or from a fragment to an activity. Data can also be sent along with the intent.
- **Array List** : is a part of collection framework and is present in java.util package. It provides us with dynamic arrays in Java. Though, it may be slower than standard arrays but can be helpful in programs where lots of manipulation in the array is needed. This class is found in java.util package.
- **List**: List in Java provides the facility to maintain the ordered collection. It contains the index- based methods to insert, update, delete and search the elements. It can have the duplicate elements also. We can also store the null elements in the list.
- **Uri**: The URI class in Java is used for creating, parsing, and manipulating Uniform Resource Identifiers. It provides methods to access URI components, resolve URIs, and relativize URIs.
- **Binding**: ViewBinding is a feature in Android that generates a binding class for each XML layout file in your project. It allows for efficient and type-safe access to views, replacing the traditional findViewById() method calls.
- **Toast**: A toast is a view containing a quick little message for the user. The toast class helps you create and show those.

4.2 List of important library functions used and their descriptions.

- **getText()** : To get the text entered in the EditText views.
- **toString()** : It's a method in java's Object class, which is the superclass of every java object. It is meant for returning a textual representation of an object.
- **setOnClickListener()** : OnClickListener and wires the listener to the button using setOnClickListener. As the result, the system executes the code you write in onClick after the user presses the button.
- **setAnimations()**: To set animations in Android, you can use the setAnimation() method on a View or the overridePendingTransition() method on an Activity. These methods allow you to specify custom animations for enter, exit, and transition between activities or view changes.
- **Inflate()**: used to inflate the layout inflater.
- **getIntent()**: used to receive the intent data.
- **startActivity()**: used to start the intent.
- **Serializable()**: In Android, Serializable is an interface that allows objects to be serialized and deserialized. It enables the conversion of an object's state into a byte stream, which can be stored or transmitted, and later reconstructed back into an object. This is commonly used for passing objects between activities or fragments, or for persisting data in storage.

4.3 Java Code with proper comment lines.

SplashScreen.java

```
package com.example.promopal;
import android.content.Intent;
import android.graphics.Typeface;
import android.os.Bundle;
import android.os.Handler;
import android.view.WindowManager;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import androidx.appcompat.app.AppCompatActivity;
import com.example.promopal.databinding.ActivitySplashScreenBinding;

public class SplashScreen extends AppCompatActivity {
    private ActivitySplashScreenBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivitySplashScreenBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        getWindow().setFlags(WindowManager.LayoutParams.FLAG_FULLSCREEN,
            WindowManager.LayoutParams.FLAG_FULLSCREEN);

        setAnimations();

        new Handler().postDelayed(() -> {
            Intent intent = new Intent(SplashScreen.this, HomePage.class);
            startActivity(intent);
            finish(); // if not given, user will come back to splash screen
        }, 5000);
    }
}
```

```
private void setAnimations() {
    Animation topAnimation = AnimationUtils.loadAnimation(this, R.anim.top_animation);
    Animation bottomAnimation = AnimationUtils.loadAnimation(this,
R.anim.bottoom_animation);

    binding.logoId.setAnimation(topAnimation);
    binding.infoId.setAnimation(bottomAnimation);
}
```

HomePage.java

```
package com.example.promopal;
import android.content.Intent;
import android.graphics.Typeface;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import com.example.promopal.databinding.ActivityHomePageBinding;

public class HomePage extends AppCompatActivity {
    private ActivityHomePageBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityHomePageBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
        setOnClickListeners();
    }

    private void setOnClickListeners() {
        binding.smsCardViewId.setOnClickListener(v -> {
            Intent intent = new Intent(this, SmsMain.class);
            startActivity(intent);
        });

        binding.emailCardViewId.setOnClickListener(v -> {
```

```
        Intent intent = new Intent(this, EmailMain.class);
        startActivity(intent);
    });
}}
```

EmailMain.java

```
package com.example.promopal;
import android.content.Intent;
import android.os.Bundle;
import android.widget.AdapterView;
import android.widget.Toast;
import androidx.appcompat.app.AppCompatActivity;
import com.example.promopal.databinding.ActivityEmailMainBinding;

public class EmailMain extends AppCompatActivity {
    private ActivityEmailMainBinding binding;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityEmailMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());

        ArrayAdapter<CharSequence>
        spinnerAdapter= ArrayAdapter.createFromResource(this, R.array.job, R.layout.spinner_item
        );
        spinnerAdapter.setDropDownViewResource(R.layout.spinner_dropdown_list);
        binding.spinnerId.setAdapter(spinnerAdapter);
        binding.buttonId.setOnClickListener(v -> onPreviewButtonClicked());
    }

    private void onPreviewButtonClicked() {
        String name = binding.editTextName.getText().toString().trim();
        String email = binding.editTextEmail.getText().toString().trim();
        String displayName = binding.editTextMyDisplayName.getText().toString().trim();
        String availableStartDate = binding.startDateId.getText().toString().trim();
    }
}
```

```

    if (name.isEmpty() || email.isEmpty() || displayName.isEmpty()) {

        Toast.makeText(this, "Fill all the details", Toast.LENGTH_LONG).show();

        } else if(binding.immediateStartId.isChecked() && !availableStartDate.isEmpty()){

            Toast.makeText(this, "Both Immediate Start and Available Start date selected",
            Toast.LENGTH_LONG).show();

        } else if (!isValidEmail(binding.editTextEmail.getText().toString())) {

            Toast.makeText(this, "Invalid Email format", Toast.LENGTH_LONG).show();

        } else {

            MessageEmail message = new MessageEmail(
                binding.editTextName.getText().toString(),
                binding.editTextEmail.getText().toString(),
                binding.editTextMyDisplayName.getText().toString(),
                binding.startDateId.getText() != null ? binding.startDateId.getText().toString() : null,
                binding.checkBoxId.isChecked(),
                binding.spinnerId.getSelectedItem() != null ?
                binding.spinnerId.getSelectedItem().toString() : null,
                binding.immediateStartId.isChecked()
            );

            Intent previewIntent = new Intent(this, EmailPreview.class);
            previewIntent.putExtra("Message", message);
            startActivity(previewIntent);
        }
    }

    private boolean isValidEmail(String email) {
        // Define your email validation pattern
        String pattern = "^[A-Za-z0-9+_.-]+@[A-Za-z0-9.-]+$";
        return email.matches(pattern);
    }
}

```

SmsMain.java

```

package com.example.promopal;

import android.content.Intent;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

```

```
import android.widget.ArrayAdapter;
import android.widget.Spinner;

import android.widget.Toast;
import com.example.promopal.databinding.ActivitySmsMainBinding;
public class SmsMain extends AppCompatActivity {
    private ActivitySmsMainBinding binding;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivitySmsMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
        binding.buttonId.setOnClickListener(v -> onPreviewClicked());
        ArrayAdapter<CharSequence> spinnerAdapter =
ArrayAdapter.createFromResource(this, R.array.job, R.layout.spinner_item );
        spinnerAdapter.setDropDownViewResource(R.layout.spinner_dropdown_list);
        binding.spinnerId.setAdapter(spinnerAdapter);
    }
    private void onPreviewClicked() {
        String name = binding.editTextName.getText().toString().trim();
        String number = binding.editTextNumber.getText().toString().trim();
        String displayName = binding.editTextMyDisplayName.getText().toString().trim();

        if (name.isEmpty() || number.isEmpty() || displayName.isEmpty()) {
            Toast.makeText(this, "Fill all the details", Toast.LENGTH_LONG).show();
        }
        else if (!isValidNumber(binding.editTextNumber.getText().toString())) {
            Toast.makeText(this, "Invalid number format", Toast.LENGTH_LONG).show();
        }
        else {
            MessageSms message = new MessageSms(
                binding.editTextName.getText().toString(),
                binding.editTextNumber.getText().toString(),
                binding.editTextMyDisplayName.getText().toString(),
                binding.startDateId.getText() != null ?
```

```
binding.startDateId.getText().toString() : null,
        binding.checkBoxId.isChecked(),
        binding.spinnerId.getSelectedItem() != null ?
binding.spinnerId.getSelectedItem().toString() : null,
        binding.immediateStartId.isChecked()
    );
    Intent previewIntent = new Intent(this, SmsPreview.class);
    previewIntent.putExtra("Message", message);
    startActivity(previewIntent);
}

}

private boolean isValidNumber(String number) {
    // Define your number validation pattern
    String pattern = "[0-9]{10}$"; // Assumes 10-digit numbers
    return number.matches(pattern);
}}
```

EmailPreview.java

```
package com.example.promopal;
import android.content.Intent;
import android.net.Uri;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.widget.Toast;

import com.example.promopal.MessageEmail;
import com.example.promopal.databinding.ActivityEmailPreviewBinding;

import java.io.File;

public class EmailPreview extends AppCompatActivity {
    public MessageEmail message;
    public String messagePreviewText;
    public ActivityEmailPreviewBinding binding;
    public static final int REQUEST_CODE_PICK_PDF = 1;
```

```

public static final int REQUEST_CODE_SEND_EMAIL =2;

public static int sent = 0;

@Override

protected void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);

    binding = ActivityEmailPreviewBinding.inflate(getLayoutInflater());

    setContentView(binding.getRoot());

    MessageDisplay();

    sendButton();

    attachPdfButton();

}

private void MessageDisplay() {

    message = (MessageEmail) getIntent().getSerializableExtra("Message");

    messagePreviewText = "Hello " + message.getContactName() + "\n\n" +

        "My name is " + message.getMyDisplayName() + " and I am " +

message.getFullDesc() + "\n\n" +

        "I have a portfolio of Projects to show for my work.\n\n" +

        "I am able to start a new position " + message.getAvailability() + "\n\n" +

        "Please get in touch if there is a suitable role for me.\n\n" +

        "Thank you and best regards.";

    binding.textViewId.setText(messagePreviewText);

}

private void sendButton() {

    binding.sendMessageButtonId.setOnClickListener(v -> {

        Intent intent = new Intent(Intent.ACTION_SENDTO);

        String[] emailadd = { message.getContactEmail()};

        intent.putExtra(Intent.EXTRA_EMAIL, emailadd);

        intent.putExtra(Intent.EXTRA_SUBJECT, "Requesting for Internship opportunity");

        intent.putExtra(Intent.EXTRA_TEXT, messagePreviewText);

        intent.setData(Uri.parse("mailto:"));

        startActivityForResult(Intent.createChooser(intent, "Send Email"),

REQUEST_CODE_SEND_EMAIL);

    });

}

private void attachPdfButton() {

```

```

binding.attachPdfButtonId.setOnClickListener(v -> {
    Intent intent = new Intent(Intent.ACTION_OPEN_DOCUMENT);
    intent.addCategory(Intent.CATEGORY_OPENABLE);
    intent.setType("application/pdf");

    startActivityResult(intent, REQUEST_CODE_PICK_PDF);
});
}
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (requestCode == REQUEST_CODE_PICK_PDF && resultCode == RESULT_OK) {
        if (data != null) {
            Uri pdfUri = data.getData();
            if (pdfUri != null) {
                attachPdfToEmail(pdfUri, resultCode, requestCode);
            }
        }
    }
    private void attachPdfToEmail(Uri pdfUri, int resultCode, int requestCode) {
        Intent emailIntent = new Intent(Intent.ACTION_SEND);
        emailIntent.putExtra(Intent.EXTRA_EMAIL, new
String[]{message.getContactEmail()});
        emailIntent.putExtra(Intent.EXTRA_SUBJECT, "Requesting for Internship
opportunity");
        emailIntent.putExtra(Intent.EXTRA_TEXT, messagePreviewText);
        emailIntent.setType("application/pdf");
        emailIntent.putExtra(Intent.EXTRA_STREAM, pdfUri);
        startActivity(Intent.createChooser(emailIntent, "Send Email"));
        mailcheck(requestCode, resultCode);
    }
    private void mailcheck(int requestCode, int resultCode)
    {
        if(requestCode == REQUEST_CODE_SEND_EMAIL)
        {
            if (resultCode == RESULT_OK)

```

```

        Toast.makeText(EmailPreview.this, "Email Sent", Toast.LENGTH_LONG).show();
    else
        Toast.makeText(EmailPreview.this, "Email not Sent ",
Toast.LENGTH_LONG).show();
    }
}
}

```

SmsPreview.java

```

package com.example.promopal;
import android.content.Intent;
import android.net.Uri;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Bundle;
import android.view.View;
import com.example.promopal.databinding.ActivitySmsPreviewBinding;
public class SmsPreview extends AppCompatActivity {
    private ActivitySmsPreviewBinding binding;
    private MessageSms message;
    private String messagePreviewText;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivitySmsPreviewBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
        MessageDisplay();
        SetButton();
    }
    private void MessageDisplay() {
        message = (MessageSms) getIntent().getSerializableExtra("Message");
        messagePreviewText = "Hello " + message.getContactName() +
            "\n\nMy name is " + message.getMyDisplayName() +
            " and I am " + message.getFullDesc() + "." +
            "\n\nI have a portfolio of Projects to show for my work." +
            "\n\nI am able to start a new position " + message.getAvailability() +
            ".\n\nPlease get in touch if there is a suitable role for me." +

```

```

        "\n\nThank you and best regards.";
        binding.textViewId.setText(messagePreviewText);
    }
    public void SetButton() {
        binding.sendMessageButtonId.setOnClickListener(v -> {
            Intent intent = new Intent(Intent.ACTION_SENDTO);
            intent.setData(Uri.parse("smsto:" + message.getContactNumber()));
            intent.putExtra("sms_body", messagePreviewText);
            startActivity(intent);
        });
    }
}

```

MessageEmail.java

```

package com.example.promopal;
import java.io.Serializable;
public class MessageEmail implements Serializable {
    private String contactName;
    private String contactEmail;
    private String MyDisplayName;
    private String StartDate;
    private boolean includeJunior;
    private String jobTitle;
    private boolean ImmediateStart;
    public MessageEmail(String contactName, String contactEmail, String MyDisplayName,
String StartDate, boolean includeJunior, String jobTitle, boolean ImmediateStart) {
        this.contactName = contactName;
        this.contactEmail = contactEmail;
        this.MyDisplayName = MyDisplayName;
        this.StartDate = StartDate;
        this.includeJunior = includeJunior;
        this.jobTitle = jobTitle;
        this.ImmediateStart = ImmediateStart;
    }
    public String getContactName() {
        return contactName;
    }
}

```

```
}  
public String getContactEmail() {  
    return contactEmail;  
}  
public String getMyDisplayName() {  
    return MyDisplayName;  
}  
public String getStartDate() {  
    return StartDate;  
}  
public boolean isIncludeJunior() {  
    return includeJunior;  
}  
public String getJobTitle() {  
    return jobTitle;  
}  
public boolean isImmediateStart() {  
    return ImmediateStart;  
}  
public String getFullDesc() {  
    return includeJunior ? "a junior " + jobTitle : "an " + jobTitle;  
}  
public String getAvailability() {  
    return ImmediateStart ? " immediately " : " from " + StartDate;  
}  
}
```

MessageSms.java

```
package com.example.promopal;  
import java.io.Serializable;  
public class MessageSms implements Serializable {  
    private String contactName;  
    private String contactNumber;  
    private String MyDisplayName;  
    private String StartDate;  
    private boolean includeJunior;
```

```

private String jobTitle;
private boolean ImmediateStart;
public MessageSms(String contactName, String contactNumber, String MyDisplayName,
String StartDate, boolean includeJunior, String jobTitle, boolean ImmediateStart) {
    this.contactName = contactName;
    this.contactNumber = contactNumber;
    this.MyDisplayName = MyDisplayName;
    this.StartDate = StartDate;
    this.includeJunior = includeJunior;
    this.jobTitle = jobTitle;
    this.ImmediateStart = ImmediateStart;
}
public String getContactName() {
    return contactName;
}
public String getContactNumber() {
    return contactNumber;
}
public String getMyDisplayName() {
    return MyDisplayName;
}
public String getStartDate() {
    return StartDate;
}
public boolean isIncludeJunior() {
    return includeJunior;
}
public String getJobTitle() {
    return jobTitle;
}
public boolean isImmediateStart() {
    return ImmediateStart;
}
public String getFullDesc() {
    return includeJunior ? "a junior " + jobTitle : "an " + jobTitle;
}

```

```
}  
public String getAvailability() {  
    return ImmediateStart ? " immediately " : " from " + StartDate;  
}  
}
```

4.4 Manifest File code

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"
        />
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"
        android:maxSdkVersion="32" />

    <application
        android:allowBackup="true"
        android:dataExtractionRules="@xml/data_extraction_rules"
        android:fullBackupContent="@xml/backup_rules"
        android:icon="@drawable/promopal_high_resolution_logo_black_on_white_background"
        android:label="PromoPal"

        android:roundIcon="@drawable/promopal_high_resolution_logo_black_on_white_backgrou
nd"
        android:supportsRtl="true"
        android:theme="@style/Theme.PromoPal"
        tools:targetApi="31">
        <activity
            android:name=".EmailPreview"
            android:exported="false" />
        <activity
            android:name=".EmailMain"
            android:exported="false" />
        <activity
            android:name=".HomePage"
            android:exported="false" />
        <activity
            android:name=".SplashScreen"
            android:exported="true">
```

```

<intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
</intent-filter>

<meta-data
    android:name="android.app.lib_name"
    android:value="" />
</activity>
<activity
    android:name=".SmsPreview"
    android:exported="false"
    android:label="Message Preview"
    android:parentActivityName=".SmsMain" />
<activity
    android:name=".SmsMain"
    android:exported="false" />
</application>
</manifest>

```

CHAPTER 5

RESULTS AND DISCUSSION

5.1 Screenshots



Figure 5.1: Splash Screen

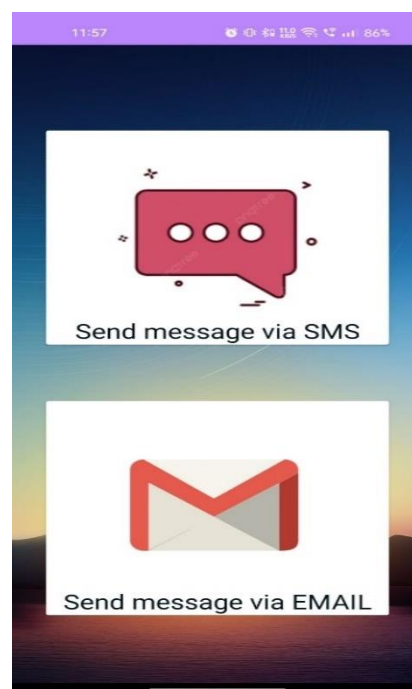


Figure 5.2: Home page

11:57 86%

Recruiter name
Darshan

Recruiter Email
anirudhrao290@gmail.com

My Name
Anirudh Rao

☒ Include junior

UI/UX Designer

☐ immediate start

available start date
02/07/2023

PREVIEW MESSAGE

Figure 5.3: Email Main

11:56 86%

Hello Darshan

My name is Anirudh Rao
and I am a junior UI/UX
Designer.

I have a portfolio of
Projects to show for my
work.

I am able to start a new
position from 02/07/2023.

Please get in touch if there
is a suitable role for me.

Thank you and best
regards.

Send message

Attach Resume

Figure 5.4: Email Preview

11:55

Recruiter name
Harish

Recruiter number
8762723647

My Name
Anirudh Rao

☒ Include junior

Full Stack Developer

☒ immediate start

available start date

PREVIEW MESSAGE

Figure 5.5: SMS Main

11:55

Hello Harish

My name is Anirudh Rao
and I am a junior Full Stack
Developer.

I have a portfolio of
Projects to show for my
work.

I am able to start a new
position immediately .

Please get in touch if there
is a suitable role for me.

Thank you and best
regards.

Send message

Figure 5.6: SMS Preview

CHAPTER 6

CONCLUSION & FUTURE ENHANCEMENTS

Through the project we came to know how to use and implement an android app. We were able to understand the importance of designing a good and clean interface for android applications. The app's user-friendly interface allows users to input recipient details, customize promotional messages, and preview auto-generated messages before sending them. PromoPal simplifies the process of self-promotion, enabling users to effectively communicate their message and enhance their promotional efforts. In the Future PromoPal could be enhanced by integrating additional communication channels such as social media platforms and messaging apps. Advanced customization options like different message templates, font styles, colors, and multimedia elements could be added. Implementing analytics and tracking features would enable users to measure campaign effectiveness. These future enhancements would make PromoPal a more versatile and powerful tool for self-promotion.

Future Enhancements for this application would be

- Additional Communication Channels
- Advanced Customization
- Integration with Contact Management

REFERENCES

The books and the other sources we referred while doing this project are as follows:

Referred Books

- Google Developer Training **Android Developer Fundamentals Course – Concept Reference** Google Developer Training Team, 2017

Websites Referred

- <https://www.youtube.com>
- <https://www.geeksforgeeks.org/>
- <https://developer.android.com/>