

VISVESVARAYA TECHNOLOGICAL UNIVERSITY

“JNANA SANGAMA”, BELAGAVI-590018



**A MOBILE APPLICATION DEVELOPMENT
LABORATORY AND MINI PROJECT (18AIL68)**

REPORT ON

FOOD WASTAGE MANAGEMENT

Submitted in partial fulfillment of the requirements for the award of degree of

Bachelor of Engineering

In

Artificial Intelligence & Machine Learning

By

ANANYA P [1KS20AI003]

Under the guidance of

Mr. Abhilash Bhat

Asst. Prof, Dept. Of CSE

Ms. Sahana Sharma M

Asst. Prof, Dept. Of AIML



KSIT
K S INSTITUTE OF TECHNOLOGY

**Department of Artificial Intelligence & Machine Learning
K.S. INSTITUTE OF TECHNOLOGY**

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109

K.S. INSTITUTE OF TECHNOLOGY

#14, Raghuvanahalli, Kanakapura Main Road, Bengaluru-560109

Department of Artificial Intelligence & Machine Learning



CERTIFICATE

This is to certify that Mini Project work entitled **“FOOD WASTAGE MANAGEMENT”** is carried out by **Ms. ANANYA P** bearing USN **1KS20AI003** bonafide student of **K.S. Institute of Technology** in the partial fulfillment for the award of the **Bachelor of Engineering in Artificial Intelligence & Machine Learning** of the **Visvesvaraya Technological University, Belagavi**, during the year 2022-23. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of Mini Project work prescribed for the said degree for the Sixth semester.

Ms. Anu Mathews
Asst. Professor & Incharge-HOD,
AIML Department

Dr. Dilip Kumar K
Principal/Director, KSIT

Mr. Abhilash Bhat
Asst. Prof, Dept. Of CSE

Ms. Sahana Sharma M
Asst. Prof, Dept. Of AIML

Name of the Examiners

- 1.
- 2.

Signature with date

ACKNOWLEDGEMENT

I take this opportunity to thank everyone involved in the successful implementation of this mini project. I would like to thank the college for providing me an opportunity to work on the mini project.

I take this opportunity to express my sincere gratitude to my college **K.S. Institute of Technology**, Bengaluru for providing the environment to work on this mini project.

I would like to express my gratitude to our **MANAGEMENT**, K.S. Institute of Technology, Bengaluru, for providing a very good infrastructure and all the support provided for carrying out this mini project work in college.

I would like to express my gratitude to **Dr. K.V.A Balaji**, CEO, K.S. Institute of Technology, Bengaluru, for his valuable guidance.

I would like to express my gratitude to **Dr. Dilip Kumar K**, Principal/Director, K.S. Institute of Technology, Bengaluru, for his continuous support.

I like to extend my gratitude to **Ms. Anu Mathews**, Asst. Professor and Incharge-Head, Department of Artificial Intelligence & Machine Learning, for providing very good facilities and all the support provided in carrying out this Mini Project successfully.

I also like to thank my Mini Project Coordinators, **Ms. Sahana Sharma M**, Asst. Professor, Department of Artificial Intelligence & Machine Learning and **Mr. Abhilash Bhat**, Asst. Professor, Department of Computer Science and Engineering for their help and support provided to carry out the Mini Project work successfully.

I am also thankful to the teaching and non-teaching staff of Artificial Intelligence & Machine Learning Department, KSIT for the help provided in completing this Mini Project.

ANANYA P

1KS20AI003

ABSTRACT

This project is on food wastage reduction through donation. A huge number of world's produced food get wasted everyday however there are one person in every seven remains hungry. So we have built an application to reduce the wastage of food.

Restaurants or other users can donate surplus food by using our application. Authorized organization can collect the food through this application and distribute them to hunger people. There are some charity organizations who are doing voluntary work to take the wasted food from restaurants & distribute it to the hunger people manually.

This application will make their work easier than before and inspire more organization to do this noble work. Food wastage is not only a symptom of craving or contamination, but also of various financial problems. In view of rapid shifts in propensities and way of life, the exclusive expectation of living has resulted in the loss of food.

TABLE OF CONTENTS

Sl. No:	CONTENTS	Page No:
1.	INTRODUCTION	
	1.1 Overview	6
	1.2 Problem statement	6
	1.3 Objective	6
	1.4 Android Studio	6
	1.5 Android Studio Architecture	6
	1.6 Methodology	7
	1.7 Project Structure	7
2.	SYSTEM REQUIREMENT	
	2.1 Introduction	8
	2.2 Functional Requirements	8
	2.3 Android Studio	8
	2.4 Android studio Emulator	9
	2.5 Java System requirement	9
	2.6 Objectives	9
3.	SYSTEM DESIGN	
	3.1 Use case Model	10
	3.2 Flow chart	
4.	IMPLEMENTATION	
	4.1 Java	11
	4.2 Overview of files in app	11
	4.3 XML Implementation	12
	4.4 Java Implementation	19
5.	SCREENSHOT	
	5.1 Homepage	32
	5.2 Donor page	32
	5.3 Receiver page	33
	5.4 How to use page	34
	5.5 Contact us	
6.	CONCLUSION	35
7.	FUTURE SCOPE AND ENHANCEMENT	36
8.	REFERENCES	37

CHAPTER 1

INTRODUCTION

1.1 Overview:

As per the knowledge the technology is going advanced and growing day by day. Over main motto is to help needy people. The idea behind over project can be use by many people who wish to donate things to needy organizations Also, many organizations like to ask for various things required by them such as clothes, food grains, books utensils. In this mobile app, we have tried to reduce food wastage by giving waste food to people or organization who need it. The needy will add to a request, in case of any leftover food donor have.

1.2 Problem Statement:

The problem addressed by the language translator app is the need for a convenient and efficient way to translate text between different languages. The app aims to provide a user-friendly interface for inputting text, selecting source and target languages, and displaying the translated text.

1.3 Objective:

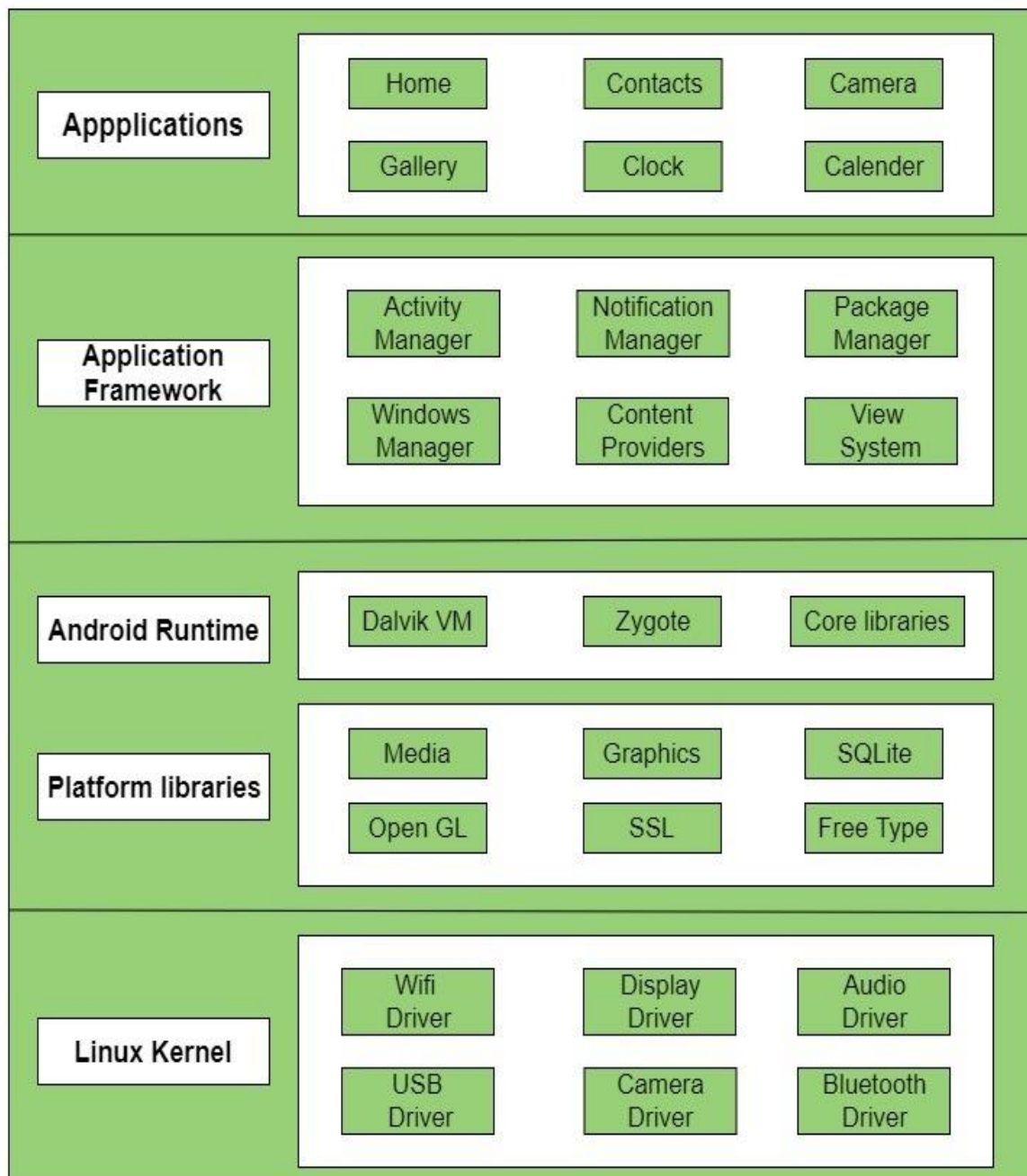
The objective of the language translator app is to simplify the process of language translation and make it accessible to users on their Android devices. The app aims to provide accurate translations and support a wide range of languages.

1.4 Android Studio:

Android Studio is the Integrated Development Environment (IDE) used for developing Android applications. It provides various tools and features to streamline the app development process, including code editing, debugging, and testing capabilities.

1.5 Android Studio Architecture:

- User Interface (UI) Layer: Deals with the presentation and interaction with the user.
- Application Layer: Contains the logic and functionality of the app.
- Android Framework: Provides a set of libraries and APIs for developing Android apps.
- Operating System: The underlying Android operating system that manages the device's resources.



1.6 Methodology:

The methodology used in developing the language translator app is not explicitly mentioned in the provided code. However, based on the code structure, it follows a traditional Android app development approach. It utilizes Firebase Natural Language and Translate APIs for language translation functionality.

1.7 Project Structure:

The project structure of the language translator app may include the following components:

- MainActivity.java: The main activity class that handles user interactions and contains the app's main logic.
- activity_main.xml: The XML layout file defining the UI components of the main activity.
- Spinner: UI components for selecting source and target languages.
- EditText: UI component for entering the source text.
- ImageView: UI component for triggering speech-to-text conversion.
- Button: UI component for initiating the translation process.
- TextView: UI component for displaying the translated text.
- String arrays: Arrays containing language names for the source and target languages.
- Helper methods: Methods for language code conversion and text translation.

CHAPTER 2

SYSTEM REQUIREMENT

2.1 Introduction:

The Food wastage management app has certain system requirements to ensure optimal performance and functionality. The minimum required version of the Android operating system is Android 7.0 Nougat or higher. It is generally recommended to have a device with a minimum of 2GB of RAM for optimal performance. The Food wastage management app requires an internet connection to access the Firebase Natural Language and Translate APIs. Ensure that the device has a stable internet connection, either through Wi-Fi or mobile data, for the app to function properly. The app supports multiple input methods for ease of use. Users can input text using the device's keyboard or utilize speech recognition for converting spoken words into text. To utilize speech recognition, the device should have a microphone or support external microphones.

2.2 Functional Requirements:

The functional requirements of the Food wastage management app:

- The app should be compatible with Android devices running a specific minimum version of the Android operating system.
- It should support input methods such as typing and speech recognition.
- The app should provide a user-friendly interface for selecting source and target languages.
- It should be able to connect to the internet for accessing Firebase Natural Language and Translate APIs.
- The translated text should be displayed accurately and clearly to the user.

2.3 Android Studio:

The system requirements for Android Studio, used for developing this application:

- The minimum version of Java Development Kit (JDK) required for Android Studio.
- The supported operating systems Windows, macOS, or Linux.
- The recommended RAM and storage space for smooth operation.

2.4 Android Studio Emulator:

The requirements for running the app on the Android Studio Emulator:

- The system requirements for the emulator, including the minimum 8 GB RAM and i5 processor.
- The Android Virtual Device (AVD) configurations: AVD with android version 7 or above.

2.5 Java System Requirement:

- The minimum version of Java Runtime Environment (JRE) 11 and Java Development Kit (JDK) 11.0.15 is required to execute the language translator app.

2.6 Objectives:

- Develop a Food wastage management app that runs on Android devices meeting the specified system requirements.
- Ensure the app functions smoothly on different screen sizes and orientations.
- Optimize the app's performance to provide quick and accurate translations.
- Implement error handling and user feedback mechanisms for a seamless user experience.

CHAPTER 3

SYSTEM DESIGN

3.1 Use Case Diagram:

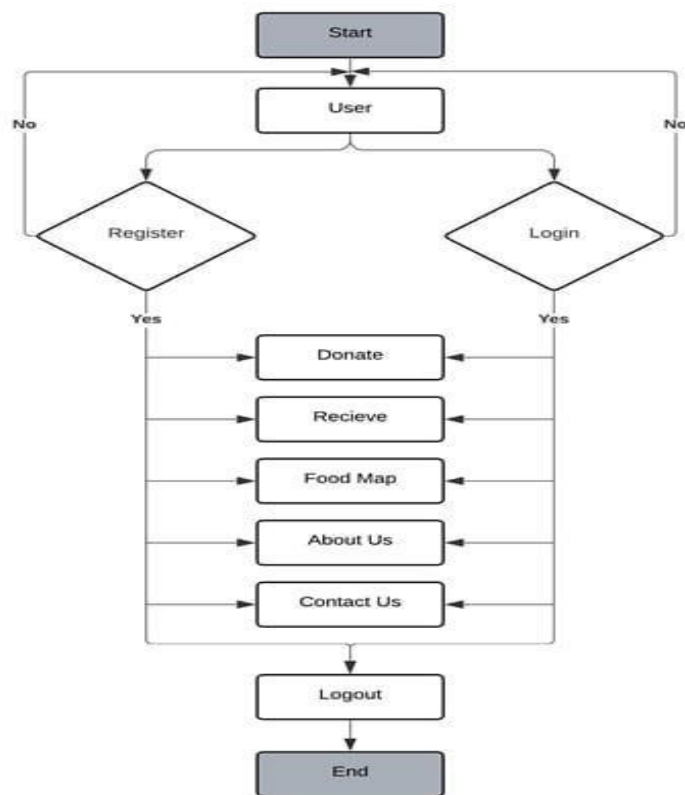
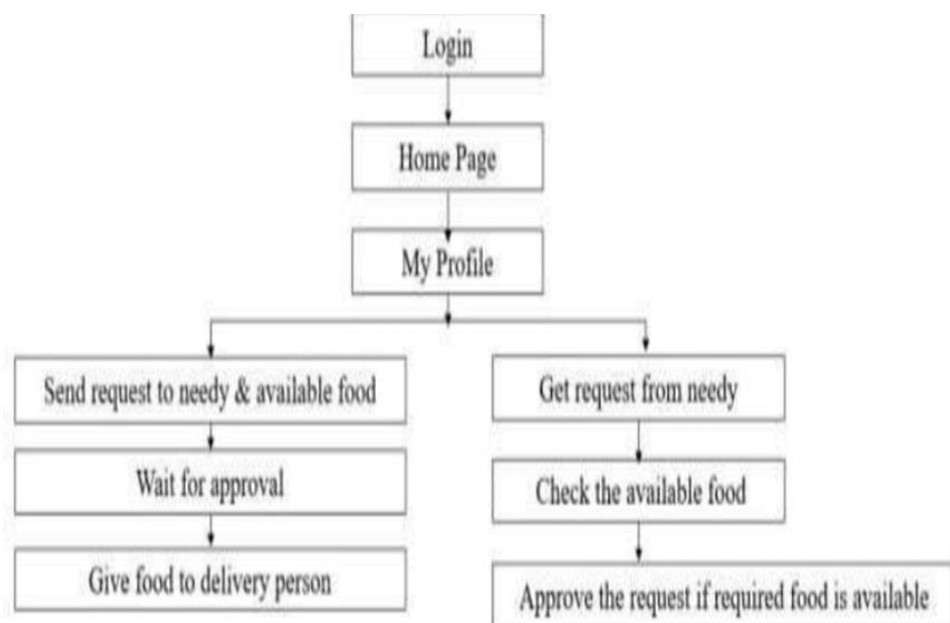


Fig 3.1.1 Represents use case diagram for Food wastage management app

Data Flow Diagram



CHAPTER 4

IMPLEMENTATION

4.1 Java:

Java is the primary programming language used for developing the Food wastage management app. It is a widely-used and versatile language that provides the necessary tools and libraries for Android app development. Java is used to implement the logic, user interactions, and translation functionality of the app.

4.2 Overview of Files in the App:

The Food wastage Management app consists of multiple files, including:

- MainActivity.java: This file contains the main activity class that serves as the entry point and handles the user interface interactions and translation logic.
- activity_main.xml: This XML file defines the layout and appearance of the main activity's user interface components.
- spinner_item.xml: This XML file defines the custom layout for the spinner items used in the language selection.
- Drawable resources: These are image or icon files used for various UI elements in the app.

4.3 XML Implementation:

The XML implementation involves defining the app's user interface layout using XML markup. The activity_main.xml file contains the layout for the main activity screen, which includes various UI elements such as spinners, text inputs, buttons, and text views. The XML files define the visual structure and appearance of the app's user interface.

4.3.1 Important features of the XML code:

1. Constraint Layout: The root element of the layout file is a ConstraintLayout, which is a flexible layout manager that allows you to create complex layouts with relative positioning of views.

2. TextView: It represents a text view widget that displays the text. It is styled with a black color and a font size of 20sp.

3. ImageView: It represents an image view widget that displays an image. It has a width and height of 70dp and is centered horizontally. It is tinted with a black color.

4. LinearLayout: It is a horizontal layout container that holds two Spinners and an ImageView. The weightSum attribute is set to 3, indicating that the child views will be divided into three equal parts.

5. TextInputLayout: It is a container that wraps a TextInputEditText, providing functionality such as floating labels and error handling. There are two instances of TextInputLayout in the code.

6. TextInputEditText: It is an editable text field where the user can enter or view text. The first TextInputEditText is used for entering the source text, while the second one is used for displaying the translated text, both have a height of 200dp.

7. TextView: This TextView displays the text "OR" and is styled with a black color and a font size of 20sp.

4.3.2 XML Code:

```
<?xml version="1.0" encoding="utf-8"?>
<layout xmlns:android="http://schemas.android.com/a pk/res/android"
    xmlns:app="http://schemas.android.com/a pk/res-auto"
    xmlns:tools="http://schemas.android.com/tools">

    <data>
        <variable
            name="food"
            type="bd .ed u . Ri fat. Food Donation App. entities. Food" />
    </data>
```

```
<android.support.design.widget.CardView
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:elevation="25dp"
    android:padding="5dp"
    app:cardBackgroundColor="#65546CF6"
    app:cardCornerRadius="30dp">
```

```
    <android.support.design.widget.ConstraintLayout
        android:layout_width="match_parent"
        android:layout_height="234dp"
        android:background="#72546CF6">
```

```
        <TextView
            android:id="@+id/name_tv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="30dp"
            android:layout_marginTop="26dp"
            android:fontFamily="@font/nunito_sans_light"
            android:text="@{food.name}"
            android:textColor="#CBFFFFFF"
            android:textSize="22dp"
            android:textStyle="bold"
            app:layout_constraintStart_toStartOf="parent"
            app:layout_constraintTop_toTopOf="parent" />
```

```
        <TextView
            android:id="@+id/phone_tv"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_marginStart="30dp"
```

```
android: layout _margin Top="10dp"
android: font Family="@font/nunito_sans_light"
android: text="@ {food . phone}"
android: text Color="#CBFFFFFF"
android: text Size="20dp"
android: text Style="bold"
app: layout _constraint Start _to Start Of="parent"
app: layout _constraint Top _to Bottom Of="@+id/name_tv" />
```

<Text View

```
android: id="@+id/male_tv"
android: layout _width="wrap_content"
android: layout _height="wrap_content"
android: layout _margin Start="30dp"
android: layout _margin Top="10dp"
android: font Family="@font/nunito_sans_light"
android: text="@ {food .gender}"
android: text Color="#CBFFFFFF"
android: text Size="20dp"
android: text Style="bold"
app: layout _constraint Start _to Start Of="parent"
app: layout _constraint Top _to Bottom Of="@+id/phone_tv" />
```

<Text View

```
android: id="@+id/age_tv"
android: layout _width="wrap_content"
android: layout _height="wrap_content"
android: layout _margin Start="50dp"
android: layout _margin Top="10dp"
android: font Family="@font/nunito_sans_light"
android: text="@ {String. value Of(food. age)}"
android: text Alignment="center"
android: text Color="#CBFFFFFF"
```

```
android: text Size="20dp"
android: text Style="bold"
app: layout_ constraint Start_ to End Of="@+id/male_ tv"
app: layout_ constraint Top_ to Bottom Of="@+id/phone_ tv" />
```

<Button

```
android: id="@+id/call B t n"
android: layout _width="wrap _content"
android: layout _height="wrap _content"
android: layout_ margin End ="110dp"
android: background="@drawable/b t n _ round"
android: font Family="@font/nun it o _ sans _light"
android: text="Request!"
android: text Color="@color/white"
android: text Size="16sp"
android: text Style="bold"
app: background Tint="#99FFFFFF"
app: layout _ constraint Bottom_ to Bottom Of="parent"
app: layout_ constraint End _to End Of="parent"
app: layout_ constraint Top _to Top Of="parent"
app: layout _constraint Vertical _bias="0.894" />
```

<Text View

```
android: id="@+id/food grp B t n"
android: layout _width="wrap _content"
android: layout _height="wrap _content"
android: layout_ margin Top="24dp"
android: layout_ margin End="48dp"
android: text="@ { food. Food Group}"
android: text Color="#CBFFFFFF"

android: text Size="25dp"
android: text Style="bold"
```



```
        app: layout_constraintEnd_toEndOf="parent"
        app: layout_constraintTop_toTopOf="parent" />
    </android.support.design.widget.ConstraintLayout>

    </android.support.design.widget.CardView>
</layout>
```

4.4 Java Implementation:

The Java implementation includes writing code in the MainActivity.java file to handle user interactions, implement translation logic, and connect with Firebase Natural Language and Translate APIs. The code implements event listeners for spinners, button clicks, and speech recognition. It also includes methods for translating text, handling permissions, and displaying the translated text.

4.4.1 Important features of java code:

- 1. Food Donation Management:** Design a module to facilitate the donation of excess or leftover food. This module should allow users to enter details about the food, such as its type, quantity, expiration date, and location. It should also include functionality to search and view available donations.
- 2. Expiration Date Tracking:** Incorporate functionality to track and manage the expiration dates of donated food items. This can involve sending notifications to users and organizations when food is approaching its expiration date or has expired.
- 3. Reporting and Analytics:** Implement reporting and analytics features to provide insights into the waste food management process. Generate reports on the quantity of donated food, successful pickups and deliveries, and other relevant metrics. This information can help organizations improve their waste management strategies.

4. Data Storage and Database Management: Implement a robust and efficient database system to store and manage data related to food donations, user profiles, pickup requests, and other relevant information. Consider using a relational database management system (such as MySQL or PostgreSQL) based on the specific requirements of the application.

5. Security Measures: Implement appropriate security measures to protect user data and ensure the confidentiality and integrity of the system. This includes input validation, data encryption, protection against common web vulnerabilities (such as cross-site scripting and SQL injection), and adherence to secure coding practices.

10. Scalability and Performance Optimization: Design the code with scalability in mind to accommodate a growing user base and increasing volumes of data. Optimize performance by using efficient algorithms, caching mechanisms, and appropriate indexing strategies for database queries.

A waste food management application can be designed to allow users to input their name, phone number, and the number of food plates available for donation. This information will be visible to all users of the application. Individuals who are interested in receiving the food can contact the donor using the provided contact number. Once the food is picked up, the donor's details will be deleted from the application to ensure privacy and prevent further contact.

4.4.2 Java Code:

```
package bd. Edu . Ri fat. Food Donation App;

import android. app. Activity;
import android. O s. Bundle;

import android x. Annotation . Non Null;
import android x. fragment. app. Fragment;
import android x. fragment. app. Fragment Result Listener;
import android x. lifecycle. View Model Provider;
import android x. navigation. Navigation;

import android. view. Layout Inflater;
import android. view. View;
```

```
import android. view. View Group;
import android. widget. Adapter View;
import android. widget. Array Adapter;
import android. widget. Radio Button;
import android. widget. Radio Group;
import android. widget. Toast;

import java. util. List;
import java. util. regex. Pattern;

import bd. Edu . Ri fat. Food Donation App. data. Data generate;
import bd. Edu . Ri fat. Food Donation App. Data binding. Fragment Second Binding;
import bd . Edu . Ri fat. Food Donation App. entities. Food;
import bd . Edu. Ri fat. Food Donation App. View models. Food View Model;

public class Second Fragment extends Fragment {

    private String gender = Food. MALE;
    private List<String> food Gr up = Data generate. Food Group;
    private String food Gr up Name;
    private Fragment Second Binding bind in g;
    private Food View Model view Model;
    public static final String REQUEST_KEY="DATE_TIME_KEY";
    public static final String DATE_KEY="DATE_KEY";
    public static final String TIME_KEY="TIME_KEY";

    public Second Fragment() {
        // Required empty public constructor
    }

    @Override
    public View on Create View(Layout Inflater inflater, View Group container,
        Bundle saved Instance State) {
        binding = Fragment Second Binding .inflate(inflater);
        view Model=new View Model Provider(require Activity()).get(Food View Model.
class);

        init Adapter();

        //Spinner Method
```

```
binding. Food Sp. Set On Item Selected Listener(new Adapter View. On Item Selected
Listener() {
    @Override
    public void on Item Selected(Adapter View<?> adapter View, View view, int i, long
l) {

        food Gr up Name = adapter View. Get Item At Position(i).to String();
        //Toast. Make Text(get Activity(), course Title, Toast. LENGTH_SHORT).show();

    }

    @Override
    public void on Nothing Selected(Adapter View<?> adapter View) {

    }
});
//Radio Group Method
binding. radio Group . set On Checked Change Listener(new Radio Group. On Checked
Change Listener() {
    @Override
    public void on Checked Changed(Radio Group radio Group, int i) {
        final Radio Button r b = radio Group. Find View By Id(i);
        gender = r b. get Text() .to String();

    }
});

Get Child Fragment Manager ().set Fragment Result Listener(REQUEST_KEY, this,
new Fragment Result Listener() {
    @Override
    public void on Fragment Result (@NonNull String request Key, @NonNull Bundle
result) {

        if (result. Contains Key (DATE_KEY)) {
            final String date=result. Get String (DATE_KEY);
            binding. date Btn. Set Text(date);
        }

    }
});
binding. date Btn. Set On Click Listener (view -> {
    new Date Picker Dialog Fragment (). Show (get Child Fragment Manager (), null);
});
```

```
binding. save Btn. Set On Click Listener(new View. On Click Listener() {
    @Override
    public void on Click(View view) {

        final String Name = binding. name Et. Get Text().to String();
        final double age = Double. Parse Double(binding. age Et. Get Text().to String());
        final String phone = binding. phone Et .get Text().to String();
        final Food student = new Food(Name, age, phone, gender, food Gr up Name);

        boolean check = validate(Name, age, phone);
        if(check == true)
        {
            Toast. Make Text(get Activity(), "DONATION SUCCESSFUL!", Toast.
LENGTH_SHORT).show();
            View Model. Add Food Donar (student);

Navigation.findNavController(view).navigate(R.id.action_secondFragment_to_fifthFragme
nt);
        }
        else
        {
            Toast.MakeText(getActivity(),"InvalidData",Toast.LENGTH_SHORT).show();
        }
        //Toast. Make Text(get Activity(), student Name, Toast. LENGTH_SHORT).show();
    }
});
// Inflate the layout for this fragment
return binding. get Root();
}
private Boolean validate(String Name, double age, String phone) {
    if (Name. length() == 0) {
        binding. name Et. Request Focus();
        binding .name Et. Set Error("FIELD SHOULD NOT BE EMPTY");
        return false;
    } else if (!Name. matches("[A-Za-z]+")) {
        binding. name Et. Request Focus();
        binding. name Et. Set Error("INVALID NAME");
        return false;
    } else if (age < 5 || age > 2000) {
        binding. age Et . request Focus ();
        binding. age Et. Set Error ("MAX: 2000 PLATES AND MIN: 5 PLATES CAN BE
DONATED");
    }
}
```

```
        return false;
    } else if (! phone. matches("[6-9]{1}[0-9]{9}")) {
        binding. phone Et. Request Focus ();
        binding. phone Et. Set Error ("INVALID PHONE NUMBER");
        return false;
    }
    else
    {
        return true;
    }
}
```

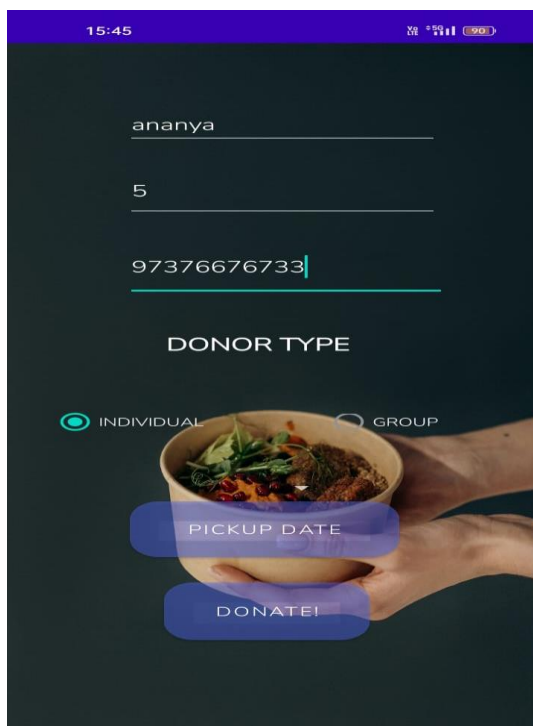
CHAPTER 5

SCREENSHOTS

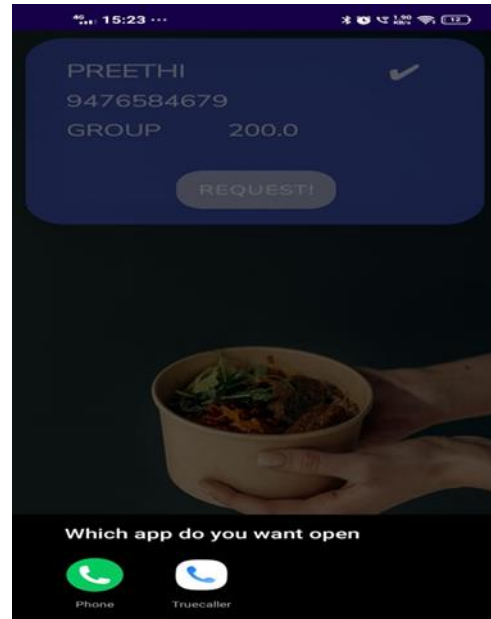
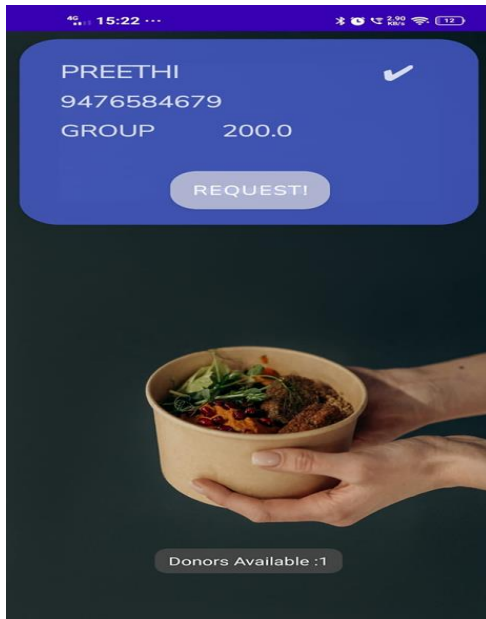
5.1 HOME PAGE:



5.2 DONOR PAGE:



5.3 RECEIVER PAGE:



5.4 HOW TO USE:



HOW TO USE?



FeedZie aims on eliminating hunger issue of India. We believe strongly in our motto that every individual, irrespective of their caste and age, must not sleep with an empty stomach. FeedZie has partnered with restaurants, organizations and individuals like you who will help us in feeding hygenic food to the ones in need.



like you who will help us in feeding hygenic food to the ones in need.

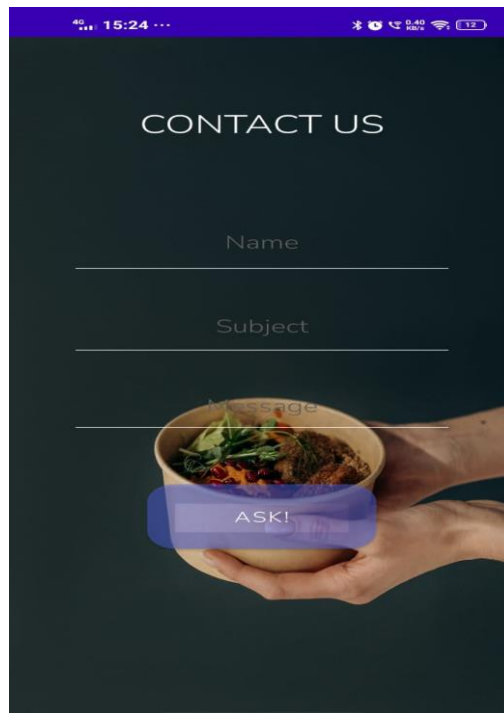
INSTRUCTIONS

Step 1: If your are a donor, add your details under the donor section

Step 2: The donor can select the number of plates. By number of plates, the donor means that he is donating a food item like rice, wheat or such cereals, pulses, fruits or meat covering one full plate.

Step 3: If you are some group/ organization that is interested in distributing foods, hop onto the receiver section and request for donation through phone calls

5.5 CONTACT US



CONTACT US

Name

Subject

Message

ASK!

CONCLUSION

We would like to conclude that our project shall aim at helping the needy by connecting them with the donors by using the NGOs as an intermediary who shall do their job aided by the application that we shall provide them. This application is an internet-based application that provides all organizations/poor people with a platform for donating leftover food. Food waste management is the process through which food and other agricultural products are recovered or redirected for human consumption, animal feed, industrial uses, or environmental benefits. At last I want to conclude that food and health both are related to each other. our health depends upon what food we eat and how much we eat. therefore we should be careful while eating.

\

FUTURE SCOPE AND ENHANCEMENT

This was our scheme of Project Structure about “Food Donation App” developed for Android based operating systems supported in Java language. The Progress of this method took tons of efforts from us. We think this method gave tons of gratification to all or any folks. However each chore isn't supposed to be flawless throughout this development pitch smooth enhancement could also be conceivable during this application. It can be said that if this application will reach to all the people of India than it is going to bring joy in life of many people as some will feel happy by donating food, clothes and books and the people who will receive these items will also feel ephorates. This application can play a major role to help India become more developed in coming future by making all the citizen of India happy and prosperous.

REFERENCES

1. <https://github.com/topics/android-project>
2. Review in Food Wastage Reduction Through Donation Application. June 2020
DOI:10.17148/IJIREEICE.2020.8611. Authors: Sankar Vt K.S.R. College of Arts and Science
3. A Review of Empirical Applications on Food Waste Prevention & Management; March 2018
Project: CoachAI: A Conversational-UI Assisted Ecoaching platform For Health & Wellbeing Authors: Ahmed Fadhil Università degli Studi di Trento, Fondazione Bruno Kessler
4. Zero Hunger: Smart Food Donation System using IoT Juhi Patil, Gayatri More, Pooja Mahale, Nikita Harale and Vijaylaxmi Bittal Department of Computer Science and Engineering SVKM's Institute of Technology Dhule, MH, India
5. <https://www.lovelycoding.org/waste-foodmanagement-system/>
6. <https://ieeexplore.ieee.org/document/9641624>
7. https://ewastekuldeep.com/?gclid=CjwKCAjwvGUBhAzEiwASUMm4lkhZpzoad2XPCz_16uBETctbIw2nXirfJVSSEY7UtRcivHfXZfhoCLasQAvD_Bw
8. Food Waste Loss and Donation U.S. Environmental Protection Agency Office of Resource Conservation and Recovery May 2013
9. <https://nevonprojects.com/waste-foodmanagement-donation-app>