

**VISVESVARAYA TECHNOLOGICAL UNIVERSITY
JNANASANGAMA, BELAGAVI – 590018**



**Mobile Application Development
Mini Project on
“SANTRIPTI”
(Food Donation Application)**

Submitted in partial fulfillment of the requirement of the Sixth Semester

**Bachelor of Engineering
In
COMPUTER SCIENCE AND ENGINEERING**

Submitted By

BHUMIKA R BHARADWAJ

1B019CS023

Under the Guidance of

**Prof.Mamatha B
Assistant Professor, Dept. of CSE**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Brindavan College of Engineering

**DWARAKANAGAR, BAGALUR MAIN ROAD, YELAHANKA, BANGALORE-63,
2021-2022**

Brindavan College of Engineering

Department of Computer Science and Engineering



CERTIFICATE

This is to certify that the **Mobile Application Development Laboratory (18CSMP68)** Mini Project work entitled “**SANTRIPTI (Food Donation Application)**” is a bonafide work carried out by **BHUMIKA R BHARADWAJ 1BO19CS023** in partial fulfillment for the requirements of Sixth Semester, **Bachelor of Engineering in Computer Science and Engineering of Visvesvaraya Technological University, Belagavi** during the year **2021-2022**. It is certified that all corrections and suggestions indicated for the internal assessment have been incorporated in the report. This Mini Project report has been approved as it satisfies the academic requirements in respect to the work prescribed for the Bachelor of Engineering degree.

.....
Prof.Mamatha B
Assistant Professor
Department of CSE

.....
Head of Department
Prof. Meena Kumari K S
Department of CSE

External Examiners

Name of Examiners

1 . _____

2 . _____

Signature with Date

1. _____

2 . _____

ABSTRACT

Wasting food is a common problem in our society. Food waste management is crucial since it can improve our environmental and economic sustainability. We have identified the use of mobile technology to reduce food waste management and built an android mobile application that allows individual user or restaurants to donate and share their foods and leftovers with people in need. In order to reduce that food wastage problem through android application we planned to do this project. In this project the guest can login & enter their Location, amount of food and type of food available. Then a simple notification is given to the agent. After seeing the notification, the agent among that location can login & can gather the details of the donor. The donor can hold an account in this application & whenever there is food wastage, he can login and enter the details of food and location. The agent can also hold an account and can retrieve the details. After retrieving the details, the agent can collect food from the donor and can redistribute to the orphans or others. This project is food redistribution is an enormously successful social innovation that tackles food waste and food poverty. The user's details are maintained confidential because it maintains a separate account for each user

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompanies the successful completion of any task would be incomplete without mentioning the people who made it possible. With deep gratitude, we acknowledge all those guidance and encouragement, which served as beacon of light and crowned our efforts with success. We thank each one of them for their valuable support.

I express my sincere thanks to **Dr. B.R. Shetty, Chairman**, Brindavan College of Engineering, Bangalore, for providing necessary facilities and motivation to carry out Mini project work successfully.

I express my sincere thanks to **Dr. Rajashekar Patil, Principal**, Brindavan College of Engineering, Bangalore, for providing necessary facilities and motivation to carry out Mini project work successfully.

I express my heartfelt gratitude and humble thanks to **Professor Meena Kumari K S, Head of Department**, CSE, Brindavan College of Engineering, for the constant encouragement and help to carry out Mini Project work successfully.

I would also like to express my humble thanks to Mini Project guides, **Assistant Professor Mamatha B** CSE, Bangalore for guiding and having facilitated us to complete our Mini project work successfully.

I would like to mention my special thanks to all the faculty member of Computer Science and Engineering Department, Brindavan College of Engineering for their invaluable support and guidance.

BHUMIKA R BHARADWAJ 1BO19CS023

TABLE OF CONTENTS

Sl. No	Titles	Page. No
1.	Abstract	i
2	Acknowledgement	ii
3	Table of Contents	iii

Chapter No	Titles	Page. No
1	Introduction	1
	1.1 Overview	1
	1.2 Problem Statement	2
	1.3 Mobile Application Development Importance	2
2	System Requirement Specifications	4
	2.1 Software Requirement	4
	2.2 Hardware Requirement	4
	2.3 Software Tools used	5
3	System Analysis	7
	3.1 Feasibility Report	7
	3.2 Technical Feasibility	7
	3.3 Economical Feasibility	8
4	System Design	9
	4.1. System Architecture	9
	4.2. Sequence Diagram	10
5	Implementation	12
	5.1. Java	12
	5.2. Source Code	12
	Main Activity XML	12
	Main Activity Java	16
6	Snapshots	19
	Conclusion	22
	Future Enhancement	23
	References	24

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Mobile application development is the process to making software for smart phones and digital assistants, most commonly for Android and iOS. The software can be preinstalled on the device, downloaded from a mobile app store or accessed through a mobile web browser. The programming and markup languages used for this kind of software development include Java, Swift, C# and HTML5.

Mobile app development is rapidly growing. From retail, telecommunications and e-commerce to insurance, healthcare and government, organizations across industries must meet user expectations for real-time, convenient ways to conduct transactions and access information. Today, mobile devices and the mobile applications that unlock their value are the most popular way for people and businesses to connect to the internet. To stay relevant, responsive and successful, organizations need to develop the mobile applications that their customers, partners and employee's demand. Yet mobile application development might seem daunting. Once you've selected the OS platform or platforms, you need to overcome the limitations of mobile devices and usher your app all the way past the potential hurdles of distribution. Fortunately, by following a few basic guidelines and best practices, you can streamline your application development journey. We can start explaining mobile development, which is not about building phone apps, though it is a huge part of it.

Mobile development presents a reasonably distinctive chance for a one-person development team to build an actual, usable, significant app end-to-end during a comparatively short period. However, mobile apps development represents more than just a chance for the solo-developer to create their own project as it is arguably the longer term of development, as mobile devices are getting larger and bigger parts of our lives.

1.2. PROBLEM STATEMENT

Food Donation Project System is a mission to end hunger and no wasting of food to make a hungry-free world. According to the latest survey, 1.3 billion tons of food is thrown as waste every year. Additionally, one-third of the food consumed is stated as leftovers. The focus of this project is to reduce the amount of food wasted and being used to the needy people. Therefore, an android-based application is developed by which a person can donate food with their capacity and at the same time the application lets the organization to put their request on their requirements. The basic prerequisite to use this Food Donation Project

1.3. MOBILE APPLICATION DEVELOPMENT NEED & IMPORTANCE

App development is essential today for an online business. If you want to boost your sales using technology, app development is truly recommended for everyone. Here listed 4 reasons that you need to know how much it is notable.

- **Accessibility from Variety of Platforms**

You may gain access to nearly every type of online platforms by developing your apps. Develop apps will help reach into marketplaces as far-reaching areas via Google Play, Blackberry, as well as Apple App Store, Symbian, and other internet marketplaces and through social media web sites just like Facebook or Myspace, Twitter, among others. Besides having the ability to mail data to clients, app growth possesses additional exclusive capabilities, which include coupon codes, evaluation of functions, and also force announcements. Your visitors within the quickest feasible valuable time, and also obtain an immediate answer, which assists you examine the advertising tool.

- **Targeted Audience**

It's simple to get obsessed with app ideas. I hear fantastic ideas daily, but generally they're strategies that originate around a work or function, instead of a particular target audience. The audience is very close to an afterthought, merely crucial while making the advertising 13 times improving the effectiveness of their procedures. Can decrease their producing expenditures, which make your company environment friendly. Threatened functions offer customers the ability of handling their companies and never have to get the cost of employing an additional worker for carrying this out train.

Internet based functions will not need customers to install these types of on their hard disk drives that lead to the decrease of memory space. Moreover, any specific up-to-date variations can be found immediately for the customers. The dependable and also efficiently created internet-based functions are created to make sure that they are suitable for all of the different internet browsers, working devices, and also equipment.

- **To engage with customers everywhere**

You can involve with your all types of customers everywhere by developing your apps system. Modern and updated apps perform multiple tasks in business as well as other site too. You should develop your apps system that is really user friendly and easy to access so that customer can contact anytime from anywhere they are.

CHAPTER 2

SYSTEM REQUIREMENT SPECIFICATION

2.1. SOFTWARE REQUIREMENTS

Software requirements deal with defining software resource requirements and prerequisites that need to be installed on a computer to provide optimal functioning of an application.

The following are the software requirements for the application:

- Operating System: Windows 10
- Development Environment: Android Studio 4.2
- API: Java Development Kit (JDK) 7
- Core Language: Java, XML for Front-end.

2.2. HARDWARE REQUIREMENTS

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware.

- CPU: Intel processor with support for Intel VT-x (Intel 64)
- Cores: Dual-Core (Quad-Core recommended)
- RAM: minimum 8GB (>8GB recommended)
- Secondary Storage: 250GB hard disk space plus at least 1GB for Android SDK,
- Emulator System images, and caches.
- Screen resolution: 1366 x 800

2.3. SOFTWARE TOOLS USED

- **Android Studio**

In recent times, Android became the world's most popular operating system for various reasons. As an Android programmer, I want to share what the Android Studio is? Android Studio is an IDE for Google Android Development launched on 16th May 2013, during Google's I/O 2013 event. Android Studio contains all the Android tools to design, test, debug, and profile your application. The Android Studio uses Gradle to manage your project, a Build Automation Tool.

For developing your first app, you need to download Android Studio for your preferred platform (Windows®, Mac OS X, or Linux) from the Android developer's site. Android Studio can develop and test your application on either a real device or an emulator.

Android Studio can be installed on Windows operating systems, OSX and Linux and is recommended by Google itself that the hardware must have at least 4 GB of memory and 1GB of free hard disk space, but we recommend that you have more memory because it was noted that Android Studio is still a little slow. You must have Java installed on the machine via the JDK (Java Development Kit), not the JRE, as it is usually installed, once to develop on Android is necessary for all Java development classes to be present on the machine.

Android Studio has many exciting features that can help you to develop your Android application like:

- Powerful code editor with smart editing and code re-factoring.
- Emulator to show your code output in various resolutions, including Nexus 4, Nexus 7, Nexus 10, and many other android phones.
- Gradle based build support.
- Maven Support.
- Template-based wizards.
- Dracula Theme Environment to enjoy your coding experience.
- You can experience all the awesome features by using Android Studio in-hand.

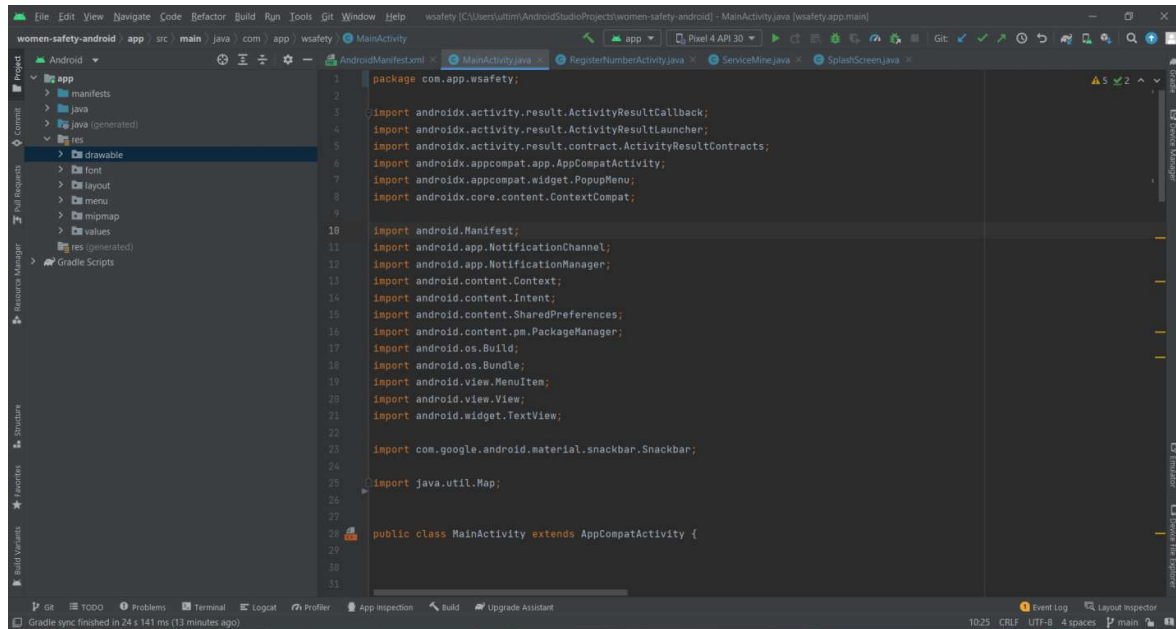


Figure 2.3.1: The Android Studio main window.

- **The toolbar** lets you carry out a wide range of actions, including running your app and launching Android tools.
- **The navigation bar** helps you navigate through your project and open files for editing. It provides a more compact view of the structure visible in the Project window.
- **The editor window** is where you create and modify code. Depending on the current file type, the editor can change.
- **The tool window bar** runs around the outside of the IDE window and contains the buttons that allow you to expand or collapse individual tool windows.
- **The tool windows** give you access to specific tasks like project management, search, version control, and more.
- **The status bar** displays the status of your project and the IDE itself, as well as any warnings or messages.

CHAPTER 3

SYSTEM ANALYSIS

3.1 Feasibility Report:

Feasibility Study is a high-level capsule version of the entire process intended to answer a number of questions like: What is the problem? Is there any feasible solution to the given problem? Is the problem even worth solving? Feasibility study is conducted once the problem is clearly understood. Feasibility study is necessary to determine that the proposed system is Feasible by considering the technical, Operational, and Economical factors. By having a detailed feasibility study the management will have a clear-cut view of the proposed system. The following feasibilities are considered for the project in order to ensure that the project is viable and it does not have any major obstructions.

Feasibility study encompasses the following things:

- Technical Feasibility
- Economic Feasibility

In this phase, we study the feasibility of all proposed systems, and pick the best feasible solution for the problem.

3.2 Technical Feasibility:

In this step, we verify whether the proposed systems are technically feasible or not. i.e., all the technologies required to develop the system are available readily or not. Technical Feasibility determines whether the organization has the technology and skills necessary to carry out the project and how this should be obtained.

Our project is technically feasible because, all the technology needed for our project is readily available.

3.3 Economical Feasibility:

In this step, we verify which proposal is more economical. We compare the financial benefits of the new system with the investment. The new system is economically feasible only when the financial benefits are more than the investments and expenditure.

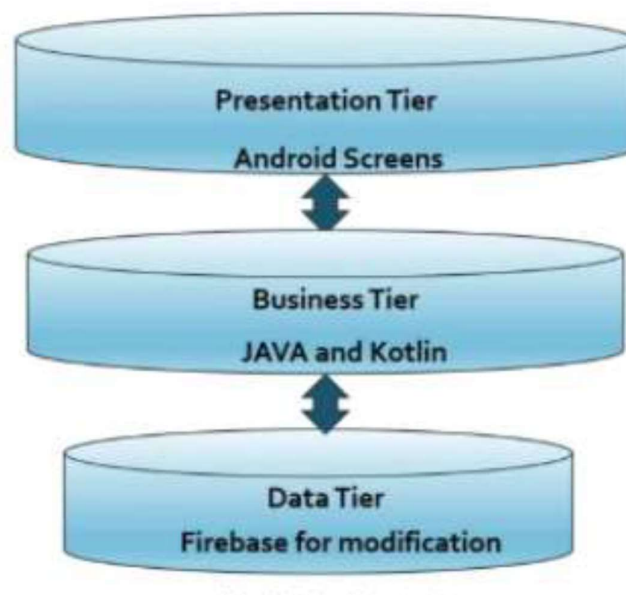
Economic Feasibility determines whether the project goal can be within the resource limits allocated to it or not. It must determine whether it is worthwhile to process with the entire project or whether the benefits obtained from the new system are not worth the costs. Our project is economically feasible because the cost of development is very minimal when compared to financial benefits of the application.

CHAPTER 4

SYSTEM DESIGN

4.1 System Architecture:

The system will be developed using 3-tier architecture:



Presentation Layer (UI): Presentation layer contains pages like .asp or windows form where data is presented to the user or input is taken from the user.

Business Access Layer (BAL) or Business Logic Layer: BAL contains business logic, validations or calculations related with the data, if needed.

Data Access Layer (DAL): DAL contains methods that helps business layer to down and caching is sufficient, the Presentation tier can process Web requests using the cache connect the data and perform required action, might be returning data or manipulating data (insert, update, delete etc.).

We selected 3-tier architecture because the three important modules like the UI, logic and database are independent of each other and are clearly defined. Also modifying any one tier will not affect the other.

Along with it we get the following benefits for using 3-tier architecture:

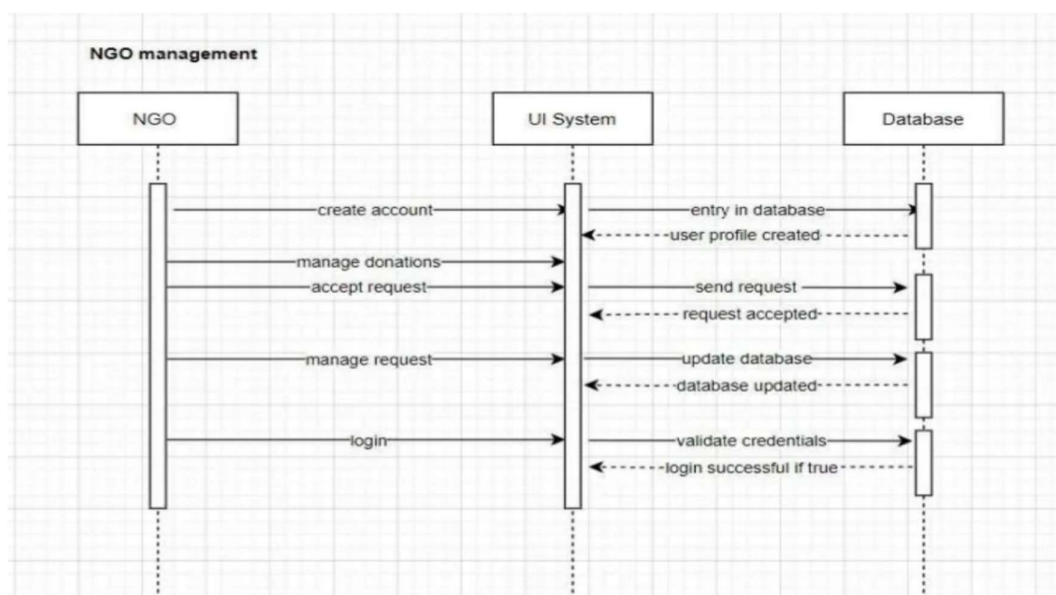
Scalability: Each tier can scale horizontally. For example, you can load-balance the Presentation tier among three servers to satisfy more Web requests without adding servers to the Application and Data tiers.

Performance: Because the Presentation tier can cache requests, network utilization is minimized, and the load is reduced on the Application and Data tiers. If needed, you can load-balance any tier.

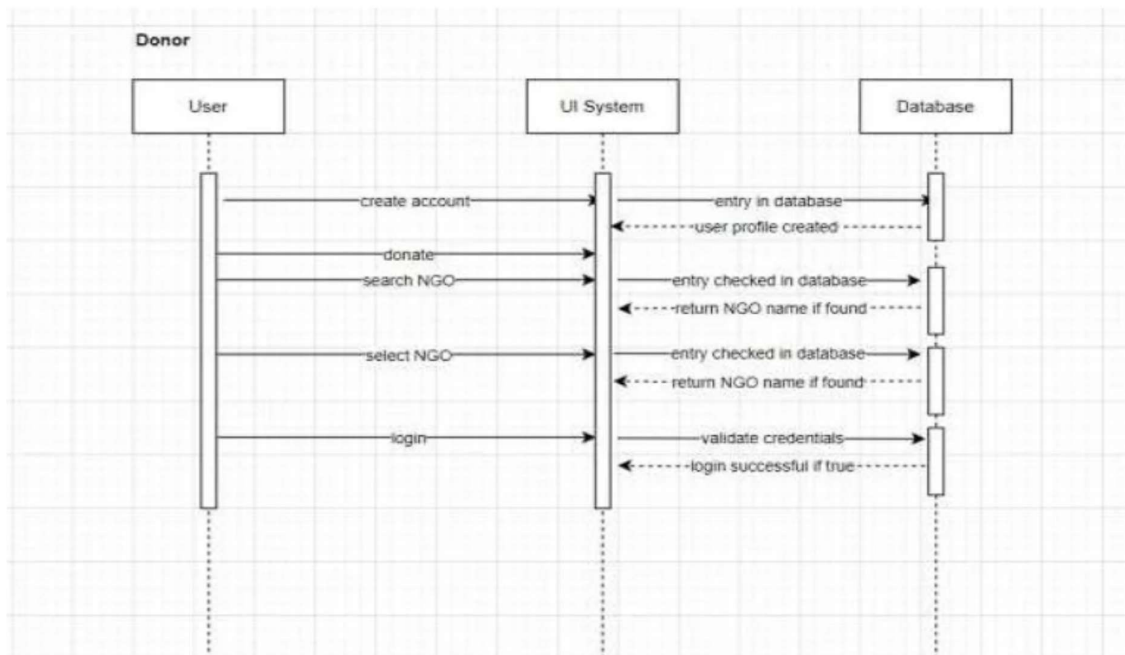
Availability: If the Application tier server is down and caching is sufficient, the Presentation tier can process Web requests using the cache.

4.2 Sequence Diagram:

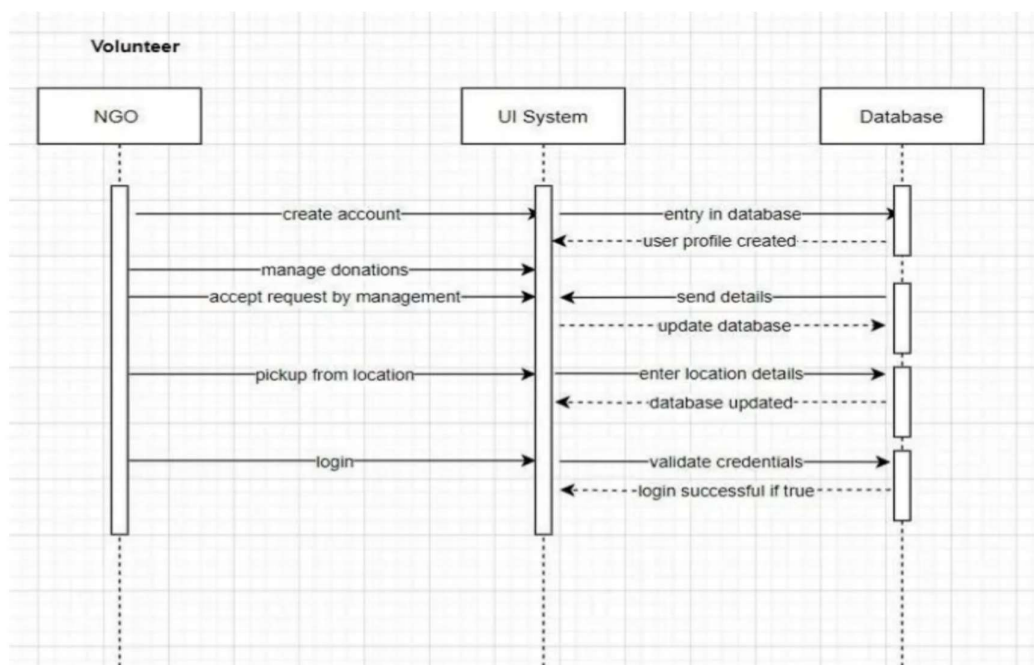
4.2.1 Sequence Diagram for NGO Management:



4.2.2 Sequence Diagram for Donor:



4.2.3 Sequence Diagram for Receiver/Volunteer:



CHAPTER 5

IMPLEMENTATION

5.1. JAVA

Java is a high-level, class-based, object-oriented programming language that is designed to have as few implementation dependencies as possible. It is a general-purpose programming language intended to let application developers *write once, run anywhere* (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of the underlying computer architecture. The syntax of Java is similar to C and C++, but has fewer low-level facilities than either of them. The Java runtime provides dynamic capabilities (such as reflection and runtime code modification) that are typically not available in traditional compiled languages. As of 2019, Java was one of the most popular programming languages in use according to GitHub, particularly for client-server web applications, with a reported 9 million developers.

5.2. SOURCE CODE

XML:

```
<?xmlversion="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:layout_height="match_parent"
  android:background="@color/black"
  tools:context=".About">
  <LinearLayout
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical"
    tools:ignore="ScrollViewSize">
    <TextView
```

```
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="@string/about"
android:fontFamily="@font/opensans"
android:textSize="26sp"
android:textColor="@color/colorPrimaryDark"
android:layout_margin="20dp"/>

<ImageView
    android:layout_width="200dp"
    android:layout_height="200dp"
    android:layout_gravity="center_horizontal"
    android:layout_marginTop="18dp"
    android:layout_marginBottom="28dp"
    android:src="@drawable/sed"
    tools:ignore="ContentDescription"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_margin="10dp"
    android:fontFamily="@font/nunitosans_light"
    android:text="@string/aboutus"
    android:textColor="@color/colorPrimary"
    android:justificationMode="inter_word"
    android:textSize="20sp"/>

<androidx.cardview.widget.CardView
    android:id="@+id/instagram"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:layout_margin="10dp"
    android:clickable="true"
    app:cardBackgroundColor="@color/gray"
    app:cardCornerRadius="12dp"
    app:cardElevation="5dp"
    tools:ignore="KeyboardInaccessibleWidget">

    <ImageView
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_gravity="center_vertical"
        android:layout_marginLeft="40dp"
        android:layout_marginRight="10dp"
        android:src="@drawable/instagram"
        tools:ignore="ContentDescription"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
        android:layout_gravity="center"
        android:fontFamily="@font/opensans"
        android:text="Instagram"
        android:textColor="@color/colorPrimary"
        android:textSize="25sp"/>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:id="@+id/facebook"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:layout_margin="10dp"
    android:clickable="true"
    app:cardBackgroundColor="@color/gray"
    app:cardCornerRadius="12dp"
    app:cardElevation="5dp"
    tools:ignore="KeyboardInaccessibleWidget">

    <ImageView
        android:layout_width="40dp"
        android:layout_height="40dp"
        android:layout_gravity="center_vertical"
        android:layout_marginLeft="40dp"
        android:layout_marginRight="10dp"
        android:src="@drawable/facebook"
        tools:ignore="ContentDescription"/>

    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:fontFamily="@font/opensans"
        android:text="Facebook"
        android:textColor="@color/colorPrimary"
        android:textSize="25sp"

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView
    android:id="@+id/twitter"
    android:layout_width="match_parent"
    android:layout_height="60dp"
    android:layout_margin="10dp"
    android:clickable="true"
    app:cardBackgroundColor="@color/gray"
    app:cardCornerRadius="12dp"
    app:cardElevation="5dp"
```

```
tools:ignore="KeyboardInaccessibleWidget">

<ImageView
    android:layout_width="40dp"
    android:layout_height="40dp"
    android:layout_gravity="center_vertical"
    android:layout_marginLeft="40dp"
    android:layout_marginRight="10dp"
    android:src="@drawable/twitter"
    tools:ignore="ContentDescription"/>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:fontFamily="@font/opensans"
    android:text="Twitter"
    android:textColor="@color/colorPrimary"
    android:textSize="25sp"/>
</androidx.cardview.widget.CardView>

<TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_gravity="center"
    android:fontFamily="@font/opensans_light"
    android:text="Version 1.0.0"
    android:textColor="@color/colorPrimary"
    android:textSize="20sp"/>
</LinearLayout>
```

JAVA:

```
package com.example.santripti;

import android.content.Intent;
import android.os.Bundle;
import android.view.View;

import androidx.appcompat.app.AppCompatActivity;
import androidx.cardview.widget.CardView;

import com.google.firebase.auth.FirebaseAuth;

public class MainActivity extends AppCompatActivity {

    CardView donate, receive, logout, foodmap, about, contact, mypin, history;
    FirebaseAuth fAuth;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        donate = findViewById(R.id.cardDonate);
        receive = findViewById(R.id.cardReceive);
        logout = findViewById(R.id.cardLogout);
        foodmap = findViewById(R.id.cardFoodmap);
        mypin = findViewById(R.id.cardMyPin);
        history = findViewById(R.id.cardHistory);
        about = findViewById(R.id.cardAboutus);
        contact = findViewById(R.id.cardContact);

        fAuth = FirebaseAuth.getInstance();
        if(fAuth.getCurrentUser() == null){
            Intent intent = new Intent(MainActivity.this, landingpage.class);
            intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |
Intent.FLAG_ACTIVITY_CLEAR_TASK | Intent.FLAG_ACTIVITY_NEW_TASK);
            startActivity(intent);
        }

        donate.setOnClickListener(new View.OnClickListener ()
        {
            @Override
            public void onClick(View v) {
                Intent intent = new Intent(getApplicationContext(), Donate.class);
                startActivity(intent);
            }
        })
    }
}
```

```
});  
receive.setOnClickListener(new View.OnClickListener ()  
{  
    @Override  
    public void onClick(View v) {  
        Intent intent = new Intent(getApplicationContext(), Receive.class);  
        startActivity(intent);  
    }  
});  
foodmap.setOnClickListener(new View.OnClickListener ()  
{  
    @Override  
    public void onClick(View v) {  
        Intent intent = new Intent(getApplicationContext(), FoodMap.class);  
        startActivity(intent);  
    }  
});  
about.setOnClickListener(new View.OnClickListener ()  
{  
    @Override  
    public void onClick(View v) {  
        Intent intent = new Intent(getApplicationContext(), About.class);  
        startActivity(intent);  
    }  
});  
mypin.setOnClickListener(new View.OnClickListener ()  
{  
    @Override  
    public void onClick(View v) {  
        Intent intent = new Intent(getApplicationContext(), MyPin.class);  
        startActivity(intent);  
    }  
});  
history.setOnClickListener(new View.OnClickListener ()  
{  
    @Override  
    public void onClick(View v) {  
        Intent intent = new Intent(getApplicationContext(), UserdataActivity.class);  
        startActivity(intent);  
    }  
});  
contact.setOnClickListener(new View.OnClickListener ()  
{  
    @Override  
    public void onClick(View v) {  
        Intent intent = new Intent(getApplicationContext(), Contact.class);  
        startActivity(intent);  
    }  
});
```

```
});  
logout.setOnClickListener(new View.OnClickListener ()  
{  
    @Override  
    public void onClick(View v) {  
        FirebaseAuth.getInstance().signOut();  
        Intent intent = new Intent(MainActivity.this, landingpage.class);  
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP |  
Intent.FLAG_ACTIVITY_CLEAR_TASK | Intent.FLAG_ACTIVITY_NEW_TASK);  
        startActivity(intent);  
    }  
});  
}  
}
```

CHAPTER 6

SNAPSHOTS

APP FRONT COVER:



WELCOME PAGE:

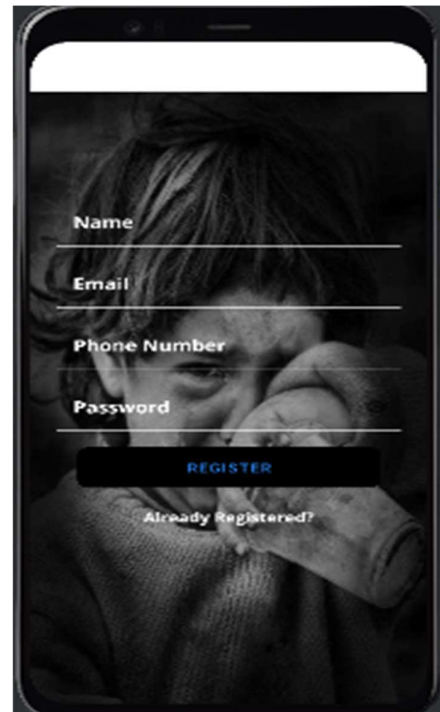


This is the front Cover page of the application home screen which will be the first page visible and the welcome page where we enter into the application.

LOGIN PAGE:

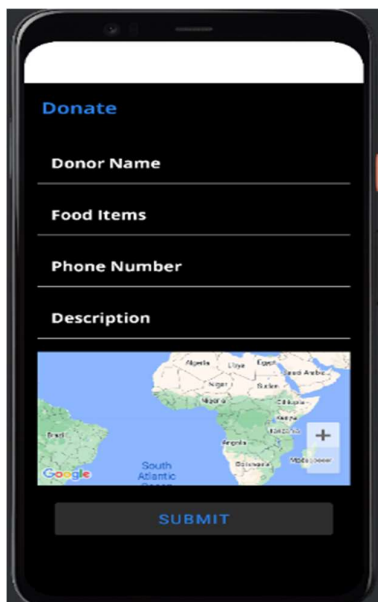
A screenshot of the login page of a mobile application. The background is a dark, high-contrast image of a smiling child. The page features a white header bar. Below the header, there are two input fields labeled 'Email' and 'Password'. A blue 'LOGIN' button is positioned below the password field. At the bottom, there is a link that says 'Not Registered! Click here to Register'.

REGISTER:

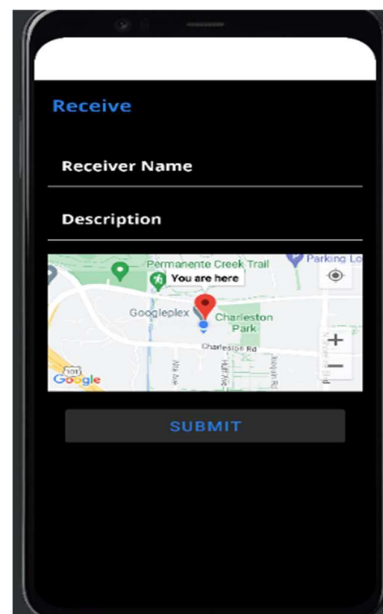
A screenshot of the register page of a mobile application. The background is a dark, high-contrast image of a child's face. The page features a white header bar. Below the header, there are four input fields labeled 'Name', 'Email', 'Phone Number', and 'Password'. A blue 'REGISTER' button is positioned below the password field. At the bottom, there is a link that says 'Already Registered?'.

This is the login and register page where one has to register first to login into the dash board.

DONATE:

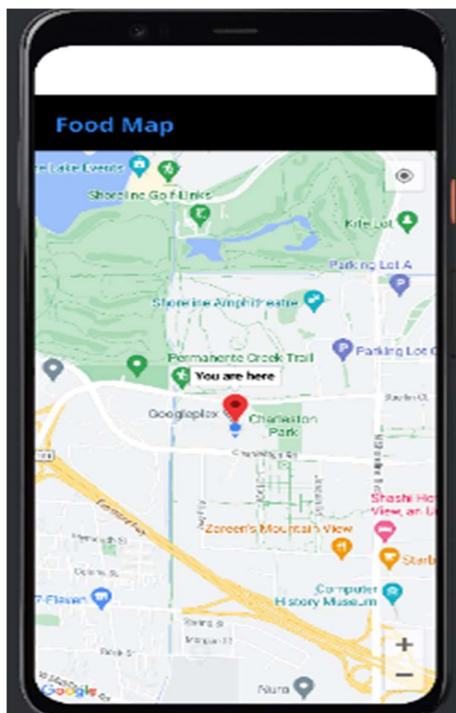
A screenshot of the donate page of a mobile application. The page has a dark background with a white header bar. Below the header, there is a blue 'Donate' title. The form includes input fields for 'Donor Name', 'Food Items', 'Phone Number', and 'Description'. Below the 'Description' field is a map of Africa. At the bottom, there is a blue 'SUBMIT' button.

RECEIVE:

A screenshot of the receive page of a mobile application. The page has a dark background with a white header bar. Below the header, there is a blue 'Receive' title. The form includes input fields for 'Receiver Name' and 'Description'. Below the 'Description' field is a map showing a location in Charleston, South Carolina. At the bottom, there is a blue 'SUBMIT' button.

The donate page allows to donate the food, where receive page can receive it.

FOODMAP:

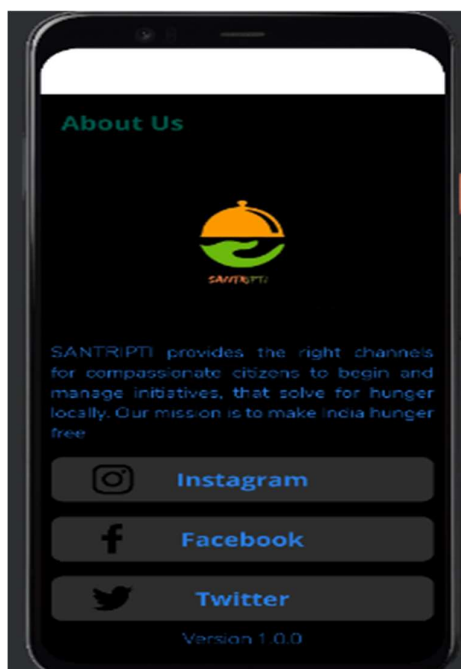


HISTORY:

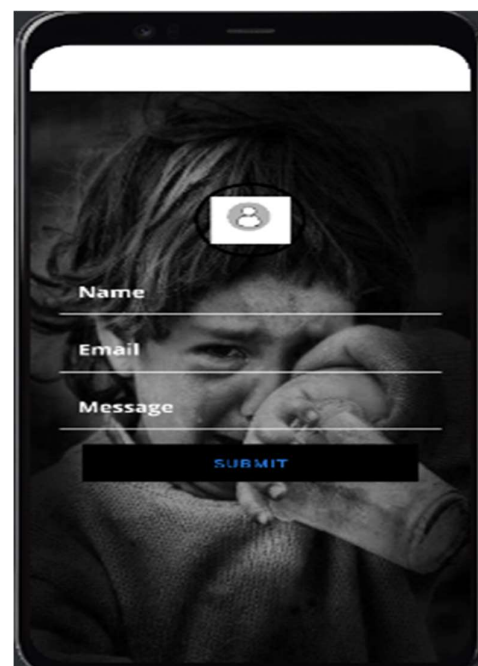


The foodmap page can locate the address of the donor and the receiver , the history page shows the lists of donors and reciever

ABOUT US:



CONTACT US:



The about us page has media information which directs us to different social media handles.

CONCLUSION

Food waste is one of the issues currently facing the planet as a whole. Necessary steps should be taken to stop food waste, otherwise the people of the world will suffer from food scarcity. If you can save food from being wasted, we can feed the hungry people who has no food to eat. We have built “Food Share” which is an android application to reduce food wastage through donation. Any restaurant or individual user who has food surplus, can donate food through our application. Our registered charity organization’s volunteer will collect the food and distribute them to hungry people. We believe that food wastage will be reduced through our project. The surplus food can be donated by our project and also poor or hungry people will get food who have not enough food to eat. So, our project can make a huge difference in the society by saving food from being wasted. Our goal is to establish a link between restaurants or individual users who has food surplus and charity organizations who has volunteer to collect the excess food.

The Development of this system takes a lot of efforts from us. We think this system gave a lot of satisfaction to all of us. Though every task is never said to be perfect in this development field even more improvement may be possible in this application. We learned so many things and gained a lot of knowledge about development field. We hope this will prove fruitful to us.

FUTURE SCOPE

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. Our application shall aim to mitigate issues like lack of awareness among donors, lack of transparency in the donation process and thus acts as a bridge between the people in need. India is a developing nation and problems such as hunger and other issues are still prevalent to a large degree. We shall try to contribute our best by connecting the people in need with the providers and donors. We shall try and expand our application scope to other platforms such as IOS and also shall try to expand our reach and the amount of help we provide.

REFERENCES

- [1] A. Ciaght and A. Villafiorita, “Beyond food sharing: Supporting food wastage reduction using ICT”, <http://esatjournals.net/ijret/2016v05/i04/IJRET20160504058.pdf>, 2016.

- [2] K. Raut, N. Shah and A. Thorat, “Food donation portal”, <http://ijarcet.org/wpcontent/uploads/IJARCET-VOL-5-ISSUE-4-906908.pdf>, 2015.

- [3] D. Shah, A. Ansari and R. Sharma, “Helping hands”, <http://ijsrd.com/Article.php?manuscript=IJSRDV4I110485>, 2016.