FOOD WASTAGE REDUCTION BY DONATION TO CHARITY INSTITUITIONS

A PROJECT REPORT

Submitted by

MEENA K [REGISTER NO:211417104144]

SHANMATHI S [REGISTER NO:211417104252]

SHOPICA V [REGISTER NO:211417104258]

In partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI - 600123.

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2021

BONAFIDE CERTIFICATE

Certified that this project report **“FOOD WASTAGE REDUCTION BY DONATION TO CHARITY INSTITUTIONS”** is the bonafide work of **“MEENA K [211417104144], SHANMATHI S [211417104252], SHOPICA V [211417104258] ”** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr.S.MURUGAVALLI,M.E.,Ph.D., **HEAD OF THE DEPARTMENT**	**Mrs. M.MAHESWARI** **ASST.PROFESSOR** **SUPERVISOR**
DEPARTMENT OF CSE, PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, POONAMALLEE, CHENNAI- 600 123.	DEPARTMENT OF CSE, PANIMALAR ENGINEERING COLLEGE, NASARATHPETTAI, POONAMALLEE, CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on...........................

INTERNAL EXAMINER	**EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt. C.VIJAYARAJESWARI**, **Thiru. C.SAKTHIKUMAR,M.E.,** and **Tmt. SARANYASREE SAKTHIKUMAR ,B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr. K.Mani, M.E., Ph.D.** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI , M.E., Ph.D.,** for the support extended throughout the project.

We would like to thank our Project Guide **Mrs. M.MAHESWARI** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

MEENA K [211417104144]
SHANMATHI S [211417104252]
SHOPICA V [211417104258]

ABSTRACT

A generous mobile application to alleviate the burden of innocent people, who need food to survive. At the same time, the application also helps to reduce the problem of food waste. Although there are a few mobile applications for help, none of them provide communication between the needy, donors, and food suppliers to solve the food wastage problem. This app provides comparative research between many charitable applications that help the community, defines the app limit, and introduces a mobile charity app.

The results of the feasibility study showed that the system is easy to use and is able to reduce 49% of food waste in the selected testing area. The program will create a standard working portal for hotels/restaurants and charities. The charity can directly contact restaurants with leftover food and a report will be generated that will show how much food is being served by which restaurant and give them reward points. Administrator, Third-party, and premium user are advanced modules. Food Donor can be any organization, institution, or college that wants to donate food and make a new food donation request. Food recipients can be any food aid company looking for food.

A new food donation request will be created on the portal and once the application has approved the request, a notice will be sent to the third-party responsible for delivering the food from the food supplier to the food recipient. Premium users are the ones who donate food on an everyday basis.

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Overview

In an area where many people suffer from malnutrition, food wastage is a disturbing problem. Roads, garbage cans, and garbage dumps provide ample evidence of this. Weddings, carnivals, restaurants, community, and family functions produce so much food. But food shortage still a problem in many areas. Food shortage indicates many economic problems. A high standard of living has led to wastage of food, clothing, etc.. because of the rapid changes in habits and lifestyle. Instead of wasting these resources, we can use them to donate them to various organizations such as orphanages, old age homes, etc. The product is a web-based application that aims to establish links between donors like restaurants, etc.. and charity organizations that are in need of food. Feed the needy is an online-based app that provides a platform for donors with leftover food to donate to all organizations in need of food.

1.2 Problem Definition

The portal is shown to be an effective means of donating food to organizations, etc. over the internet. In highly populated countries like India, food wastage is a big issue. The garbage bins, streets and landfills have ample proof to prove it. Canteens, marriages, restaurants, social and family get-togethers and functions expel out so much food. Food wastage is not only an indication of pollution or hunger, but also of many economic problems. The high standard of living has resulted in the wastage of food. Because of quick changes in habits and lifestyle instead of wasting food, the system can put them in use by donating them to various charities and organizations such as old age homes, orphanages, etc. The product is an internet-based website application that basically aims at charity through donations. Many people, institutes wish to donate things to needy organizations. Also, many organizations wish to ask for various things required by them such as clothes, food grains, utensils, etc. But there is no source available through which they can satisfy their requirements. Thereby, a web-based application has been developed through which people can donate food items as per their capacity and the application also allows organizations to put up their requests, that is items required by them.

CHAPTER 2
LITERATURE SURVEY

1. **TITLE :** Food donation portal

AUTHOR: Naman Taleti, Omkar Surve, Jenil Shah, Shrey Kyal

YEAR : 2015

DESCRIPTION:

The paper Food donation portal, which was published in 2015 summaries in brief the evolution of food donation activities and offers a medium that connects donors with NGOs. An idea for a food donation network is presented and impact on society through this medium is mentioned. The disadvantage in this paper is that there is no GPS service available. That means the system does not allow the organization or charity to find the nearest donor available in the area and they have to manually find the donors.

MERITS:

It offers a medium that connects donors with NGOs. An idea for a food donation network is presented and impact on society through this medium is mentioned.

DEMERITS:

There is no GPS service available. That means the system does not allow the organization or charity to find the nearest donor available in the area and they have to manually find the donors.

REFERENCE LINK: http://ijsrd.com/Article.php?manuscript=IJSRDV4I11048

2. **TITLE :** Helping Hands

AUTHOR: Komal Mandal, Swati Jadhav, Kruti Lakhani

YEAR : 2016

DESCRIPTION:

The paper Helping Hands, published in 2016, a new internet-based application that provides a platform for donating old stuff and leftover food to all needy people/organizations. It provides information about the motivation to come up with such an application, thereby describing the existing donation system and how the proposed product works for the betterment of society. The, disadvantage of this project is that there is no dashboard available that is, at the end of the month the system don't get all the records that how much is donated or received by the receiver.

MERITS:

An internet-based application that provides a platform for donating old stuff and leftover food to all needy people/ organizations. It provides information about the motivation to come up with such an application, thereby describing the existing donation system and how the proposed product works for the betterment of society

DEMERITS:

There is no dashboard available that is, at the end of the month the system don't get all the records that how much is donated or received by the receive

REFERENCE LINK: http://ijarcet.org/wp-content/uploads/IJARCET-VOL-5-ISSUE-4-906-908.pdf

3. **TITLE :** Beyond food sharing: Supporting food waste reduction With ICT's

AUTHOR: Aaron Ciaghi, Adolfo Villafiorita

YEAR : 2016

DESCRIPTION:

The paper Beyond food sharing: Supporting food waste reduction With ICTs', published in 2016, guaranteeing food security is key in improving the quality of life of citizens at all levels of society. The recent economic crisis has increased the number of people living in conditions of food poverty, especially in developed regions. Despite a growing awareness of the importance of reducing waste and managing food surplus, the role of ICTs in this domain is still unclear and rarely documented. This paper describes the use of ICT tools to recover food surplus at different stages of the supply chain and also describes the way forward for an integrated set of ICT tools to reduce waste from producers to households.

MERITS:

This paper describes the use of ICT tools to recover food surplus at different stages of the supply chain and also describes the way forward for an integrated set of ICT tools to reduce waste from producers to households.

DEMERITS:

Despite a growing awareness of the importance of reducing waste and managing food surplus, the role of ICTs in this domain is still unclear and rarely documented.

REFERENCE LINK: https://ieeexplore.ieee.org/document/7580874

4. **TITLE :** Food donations using a forecasting-simulation model

AUTHOR: Isaac A Nuamah, Lauren Davis, Steven Jiang, Nicole Lane

YEAR : 2016

DESCRIPTION:

The paper Food donations using a forecasting-simulation model, published in 2016, presents a methodology to estimate donations for non-profit hunger relief organizations. These organizations are committed to alleviating hunger around the world and depend mainly on the benevolence of donors to achieve their goals. However, the quantity and frequency of donations they receive varies considerably over time which presents a challenge in their fight to end hunger. A simulation model is developed to determine the expected quantity of food donations received per month in a multiware house distribution network. The simulation model is based on a state-space model for exponential smoothing. A numerical study is performed using data from a non-profit hunger relief organization. The results show that good estimation accuracies can be achieved with this approach. Furthermore, non-profit hunger relief organizations can use the approach discussed in this paper to predict donations for proactive planning.

MERITS:

It presents a methodology to estimate donations for non-profit hunger relief organizations. Non-profit hunger relief organizations can use the approach discussed in this paper to predict donations for proactive planning.

DEMERITS:

It can be time-consuming and resource-intensive. Forecasting involves a lot of data gathering, data organizing, and coordination.

REFERENCE LINK: https://ieeexplore.ieee.org/document/7408305

5. **TITLE :** Smartphone Based Waste Food Supply Chain for Aurangabad City Using GIS Location Based and Google Web Services

AUTHOR: Hitesh V. Raut , Swapnil R. Rajput , Dhananjay B. Nalawade , Karbhari V. Kale

YEAR : 2014

DESCRIPTION:

The Paper Smartphone Based Waste Food Supply Chain for Aurangabad City Using GIS Location Based and Google Web Services, published in 2014, describes the client-server GIS and Smartphone application for the hunger free city. At the client side App provide facility to food to the charity for the help of hungry people. Donors enter basic information like latitude and long quantity of waste food and type of waste along with value and contact number. Charities can pick up that waste food and deliver food to hungers. Completion of registration will placed onto server database where charities can store the entries of donor in table format and shows the optimal path between donor locations to nearest charity along with direction. So, wastage food can easily deliver to hungry people within a time.

MERITS:

It describes the client-server GIS and Smartphone application for the hunger free city. It shows the optimal path between donor locations to nearest charity along with direction. So, wastage food can easily deliver to hungry people within a time.

DEMERITS:

As the system primarily uses GIS software, which is expensive and difficult to use.

GIS system is made up of extremely complex map structures and information that may be difficult to integrate with the traditional maps to gain any meaningful information.

REFERENCE LINK: https://ijret.org/volumes/2016v05/i04/IJRET20160504058.pdf

6. **TITLE :** Development of a Sustainable Food Supply Chain by Post Harvest Program

AUTHOR: Craig Rominger, Stan Emert, Kenji Ushimaru

**YEAR : **2012

DESCRIPTION:

The paper Development of a Sustainable Food Supply Chain by Post Harvest Program introduces The Post Harvest Project (PHP), which is a group of committed technology and capital partners dedicated to delivering technological solutions to food waste in the supply chain. Of the many technological solutions being pursued by PHP, an innovative food preservation technology called nanoICE is described in detail. An effort is underway in Ghana to build a series of small community food process plants based on improved cold storage on fishing boats and on-shore facilities for preserving fish, which in an important source of protein.

The aim of PHP's effort in Ghana is to engage local communities to feed people and create opportunities, and improve health through good nutrition.

MERITS:

The Post Harvest Project (PHP), which is a group of committed technology and capital partners dedicated to delivering technological solutions to food waste in the supply chain.

DEMERITS:

As the system uses nano ice technology Nanoparticles can get into the body through the skin, lungs and digestive system. This may help create 'free radicals' which can cause cell damage and damage to the DNA. There is also concern that once nanoparticles are in the bloodstream they will be able to cross the blood-brain barrier.

REFERENCE LINK: https://ieeexplore.ieee.org/document/6387053/similar#similar

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

At present, the need of the system is completed using websites which aren't accessible quicker and supply no awareness about the service to the planet. there's no real interaction between the donor and recipient since everything disclosed by intermediates. one more reason includes there's no active mobile application available in locality.

3.2 PROPOSED SYSTEM

The system will make sure that the surplus food from functions, restaurants, and hotels goes to the hungry people instead of being thrown away resulting in food wastage. The system will act as a communication platform between the Donor and the Recipient. In our system, the Donor and the recipient can locate each other easily.

3.3 FEASIBILITY STUDY

A feasibility study is carried out to select the best system that meets performance requirements. The main aim of the feasibility study activity is to determine that it would be financially and technically feasible to develop the product.

3.3.1 TECHNICAL FEASIBILITY

This is concerned with specifying the software will successfully satisfy the user requirement. Open source and business-friendly and it is truly cross platform, easily deployed and highly extensible.

3.3.2 ECONOMIC FEASIBILITY

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. The enhancement of the existing system doesn’t incur any kind of drastic increase in the expenses.

3.4 HARDWARE REQUIREMENTS

- Processor : Intel Pentium Dual Core 2.00GHz
- Hard disk : 40 GB
- RAM : 2 GB (minimum)

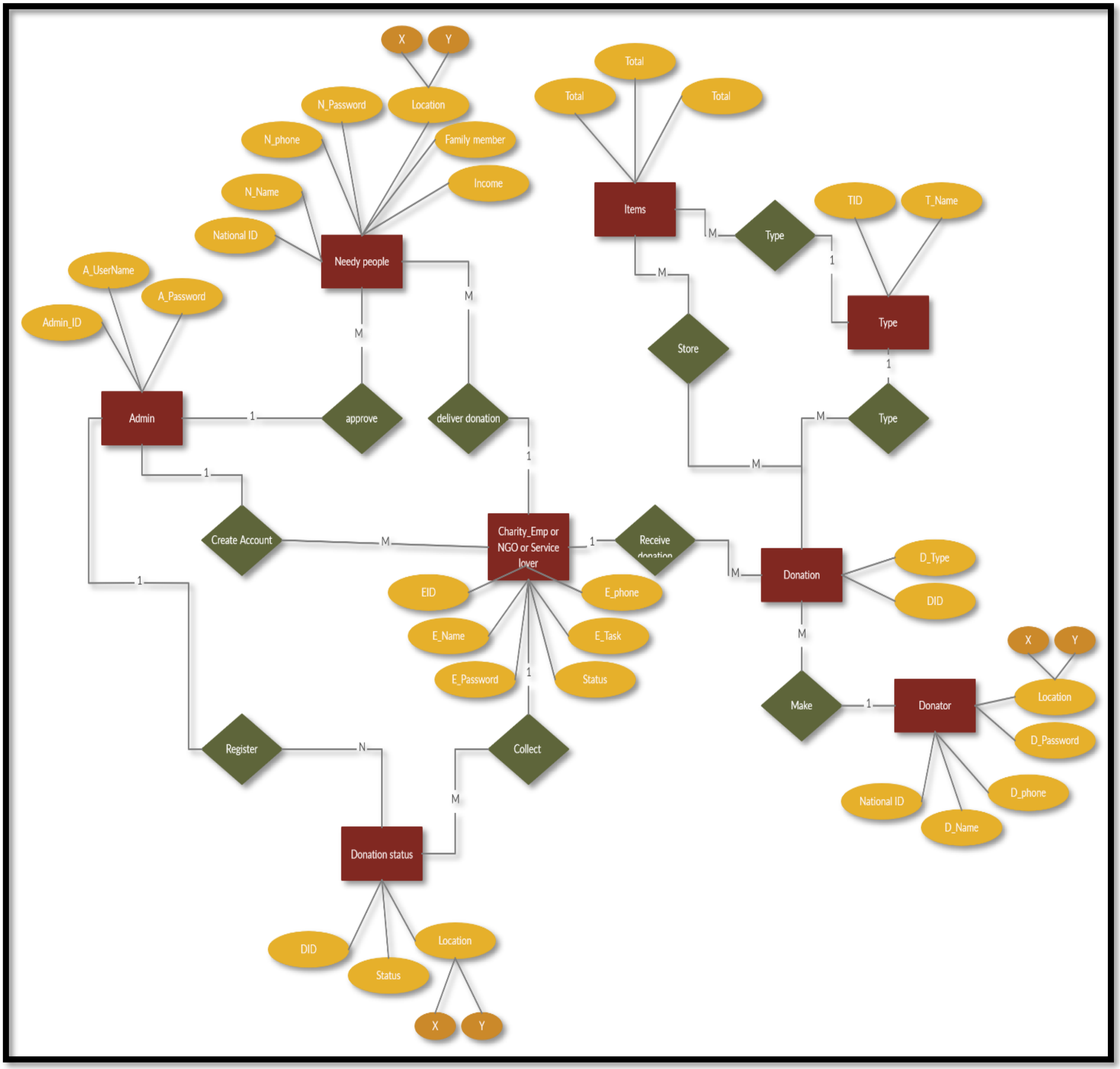
3.5 SOFTWARE REQUIREMENTS

- Android Studio (Current Version)
- Firebase
- JAVA
- Android SDK

CHAPTER 4

SYSTEM DESIGN

4.1. ER diagram



4.2 UML Diagrams

UML stands for Unified Modeling Language. It’s a rich language to model software solutions, application structures, system behavior and business processes. There are 14 UML diagram types to help you model these behaviors. Unified Modeling Language (UML) is a standard visual modeling language intended to be used for

- modeling business and similar processes,
- analysis, design, and implementation of software-based systems

UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems.

UML 1.4.2 Specification explained that process:

- provides guidance as to the order of a team’s activities,
- specifies what artifacts should be developed,
- directs the tasks of individual developers and the team as a whole, and
- offers criteria for monitoring and measuring a project’s products and activities.

UML is intentionally process independent and could be applied in the context of different processes. Still, it is most suitable for use case driven, iterative and incremental development processes. An example of such process is Rational Unified Process (RUP). Some information could be intentionally omitted from the diagram, some information represented on the diagram could have different interpretations, and some concepts of UML have no graphical notation at all, so there is no way to depict those on diagrams. For example, semantics of multiplicity of actors and multiplicity of use cases on use case diagrams is not defined precisely in the UML specification and could mean either concurrent or successive usage of use cases.

Name of an abstract classifier is shown in italics while final classifier has no specific graphical notation, so there is no way to determine whether classifier is final or not from the diagram.

List of UML Diagram Types

There are two main categories; structure diagrams and behavioral diagrams.

Structure Diagrams

Structure diagrams show the things in the modeled system. In a more technical term, they show different objects in a system. Behavioral diagrams show what should happen in a system. They describe how the objects interact with each other to create a functioning system.

Class Diagram

Class diagrams are the main building block of any object-oriented solution. It shows the classes in a system, attributes, and operations of each class and the relationship between each class. In most modeling tools, a class has three parts. Name at the top, attributes in the middle and operations or methods at the bottom. In a large system with many related classes, classes are grouped together to create class diagrams. Different relationships between classes are shown by different types of arrows.

Component Diagram

A component diagram displays the structural relationship of components of a software system. Components communicate with each other using interfaces. The interfaces are linked using connectors.

Deployment Diagram

A deployment diagram shows the hardware of your system and the software in that hardware. Deployment diagrams are useful when your software solution is deployed across multiple machines with each having a unique configuration.

Package Diagram

As the name suggests, a package diagram shows the dependencies between different packages in a system.

Composite Structure Diagram

Composite structure diagrams are used to show the internal structure of a class.

Use Case Diagram

As the most known diagram type of the behavioral UML diagrams, Use case diagrams give a graphic overview of the actors involved in a system, different functions needed by those actors and how these different functions interact. It’s a great starting point for any project discussion because you can easily identify the main actors involved and the main processes of the system.

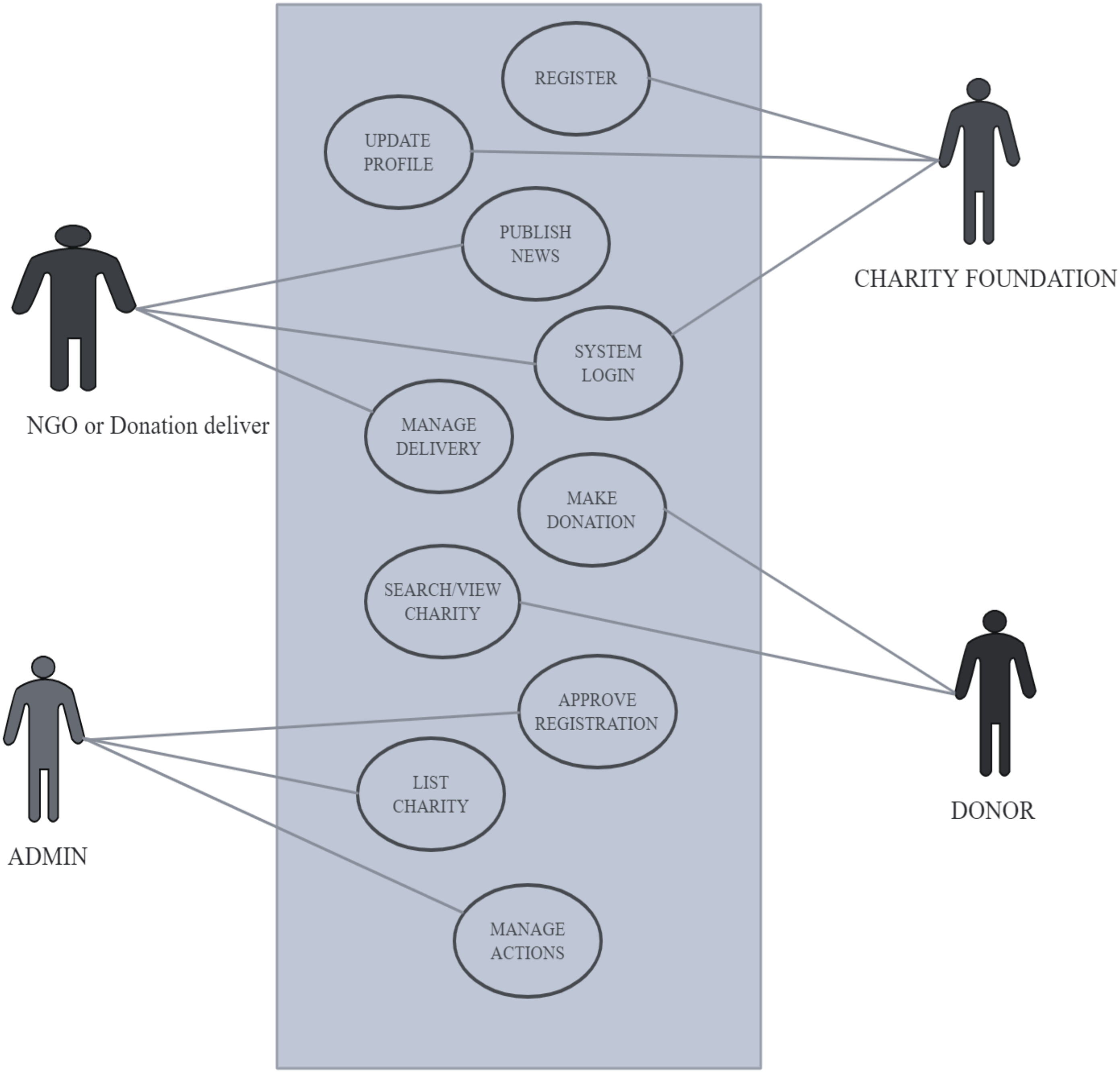
Activity Diagram

Activity diagrams represent workflows in a graphical way. They can be used to describe the business workflow or the operational workflow of any component in a system. Sometimes activity diagrams are used as an alternative to State machine diagrams.

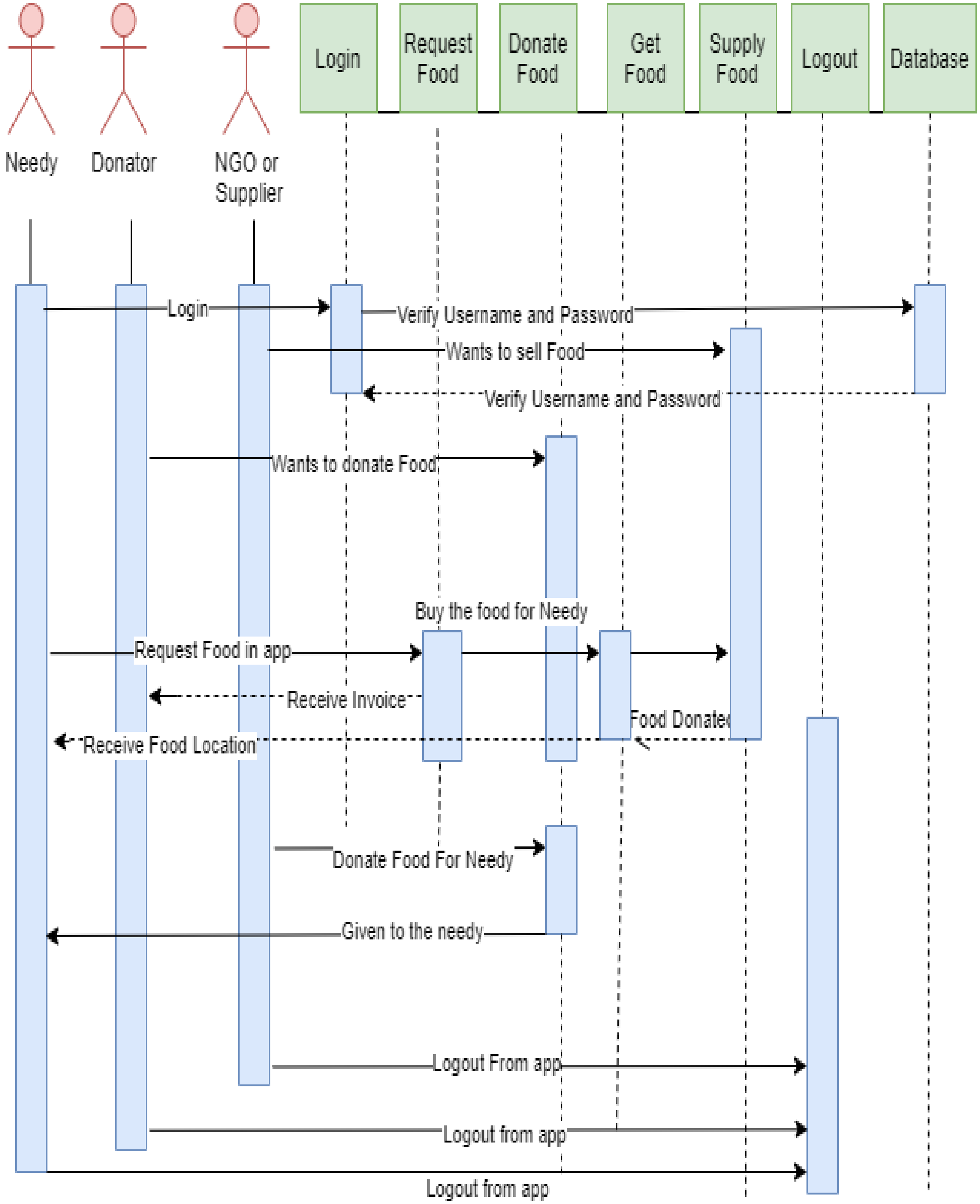
Sequence Diagram

Sequence diagrams in UML show how objects interact with each other and the order those interactions occur. It’s important to note that they show the interactions for a particular scenario. The processes are represented vertically and interactions are shown as arrows.

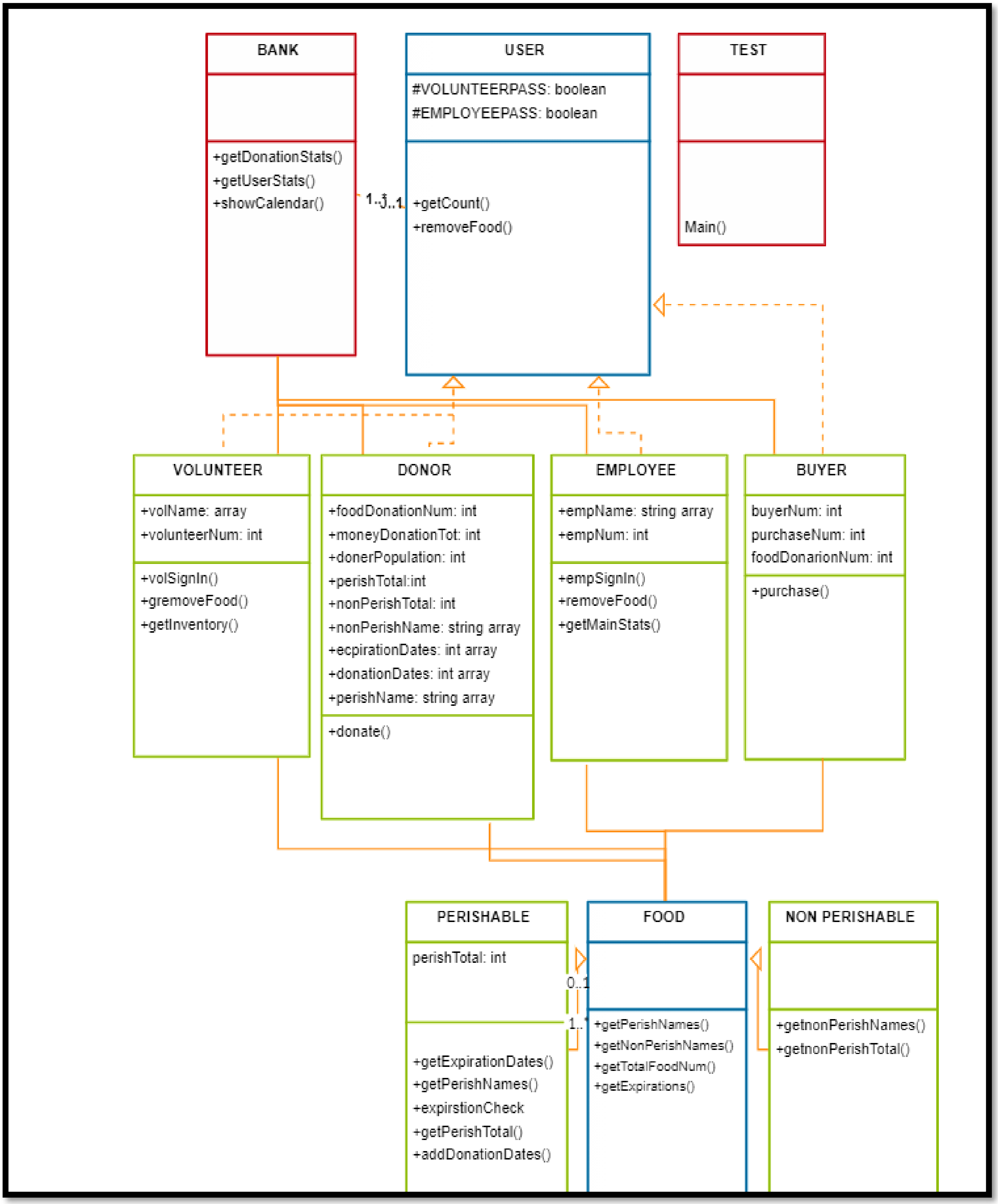
4.2.1 Use Case Diagram



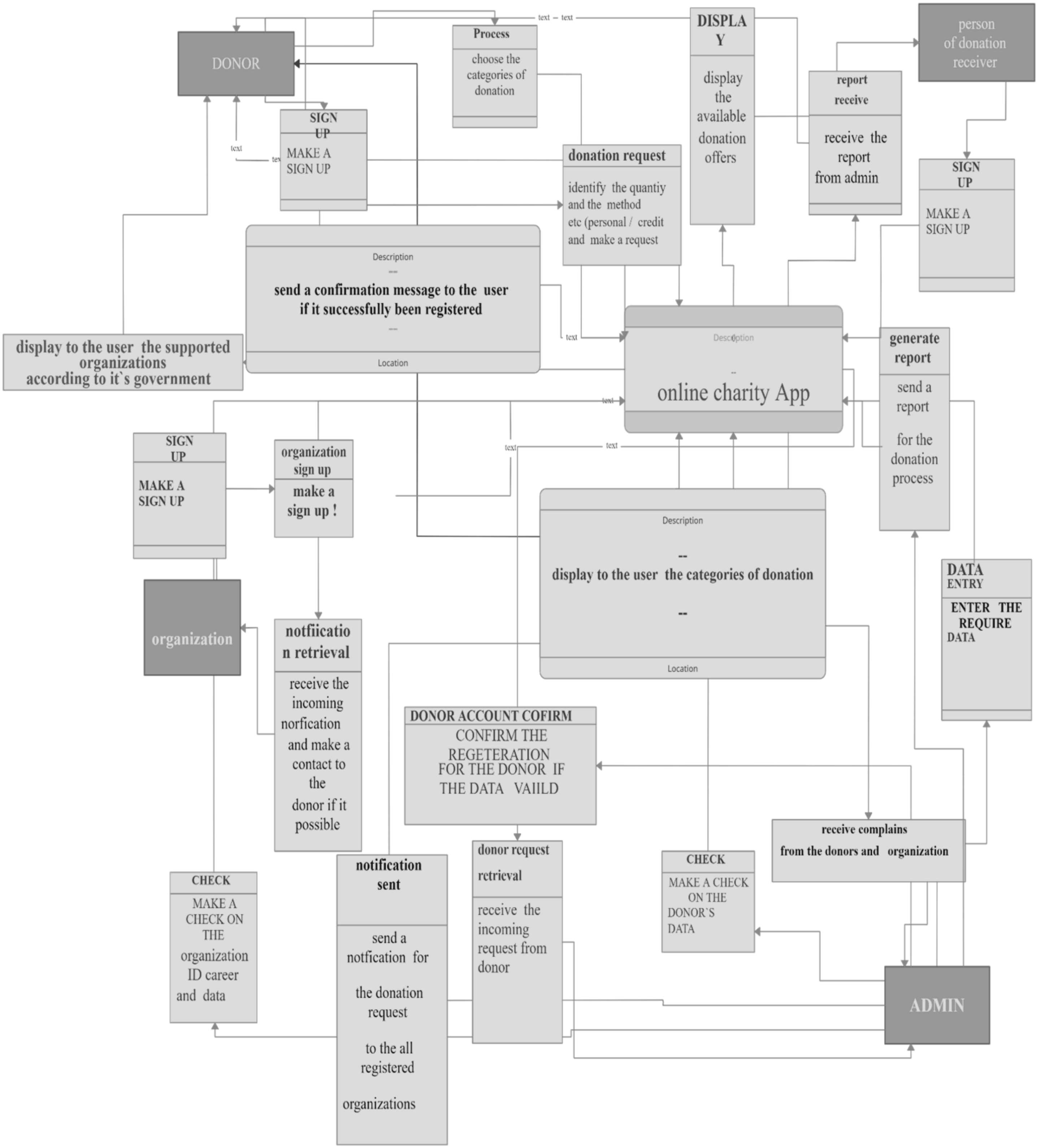
4.2.2 Sequence Diagram



4.2.3 Class Diagram



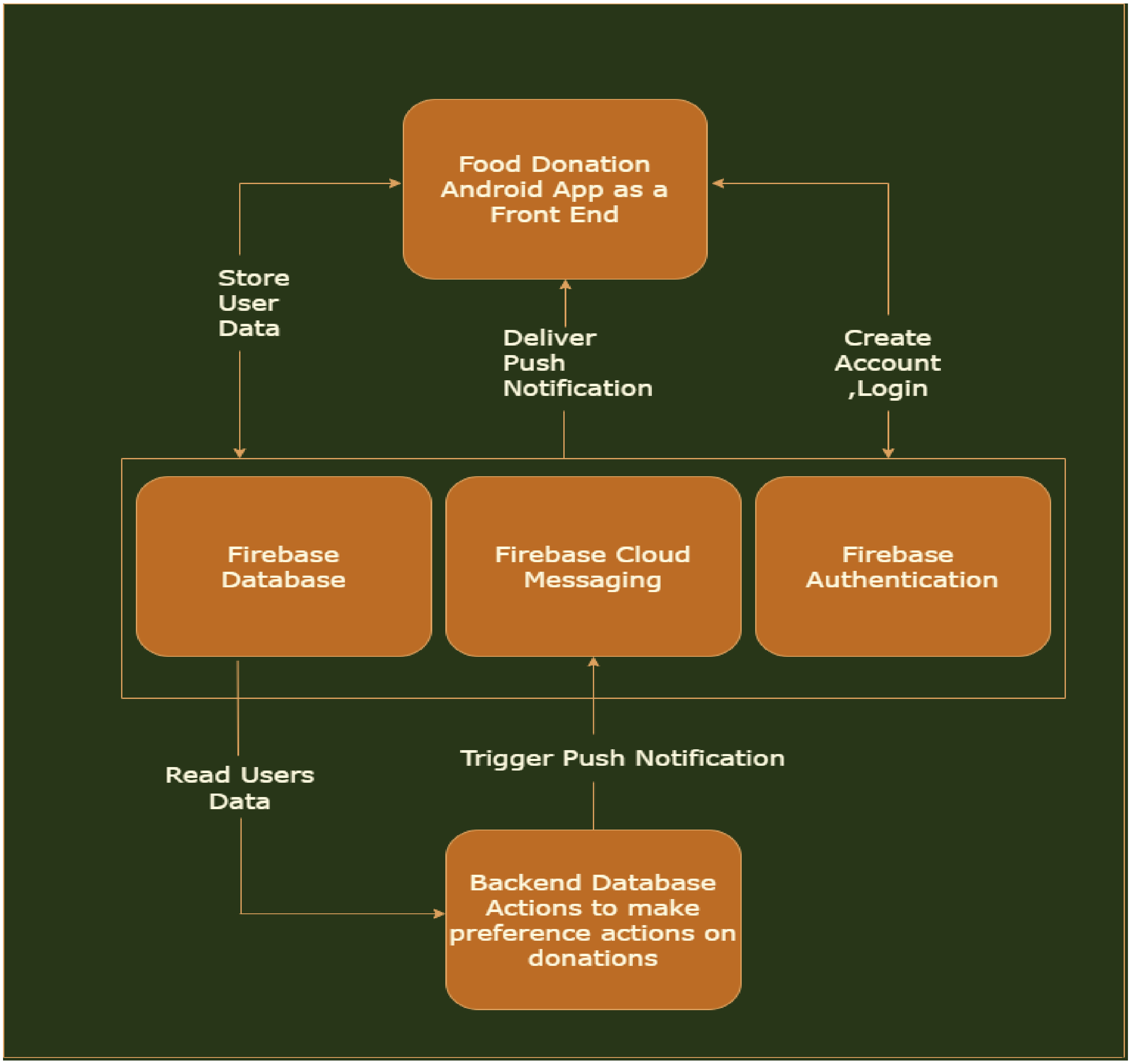
4.2.4 Data Flow Diagram


DONOR
text
text
Process
choose the categories of donation
text – text
DISPLAY
display the available donation offers
person of donation receiver
report receive
receive the report from admin
SIGN UP
MAKE A SIGN UP
donation request
identify the quantiy and the method etc (personal / credit and make a request
SIGN UP
MAKE A SIGN UP
Description
send a confirmation message to the user if it successfully been registered
Location
display to the user the supported organizations according to it`s government
text
Description
online charity App
generate report
send a report for the donation process
text
text
text
SIGN UP
MAKE A SIGN UP
organization sign up
make a sign up !
Description
display to the user the categories of donation
Location
DATA ENTRY
ENTER THE REQUIRE DATA
organization
notfiication retrieval
receive the incoming norfication and make a contact to the donor if it possible
DONOR ACCOUNT COFIRM
CONFIRM THE REGETERATION FOR THE DONOR IF THE DATA VAIILD
receive complains from the donors and organization
donor request retrieval
receive the incoming request from donor
CHECK
MAKE A CHECK ON THE DONOR`S DATA
CHECK
MAKE A CHECK ON THE organization ID career and data
notification sent
send a notfication for the donation request to the all registered organizations
ADMIN

CHAPTER 5

SYSTEM ARCHITECTURE

5.1 Architecture Overview


Food Donation Android App as a Front End
Store User Data
Deliver Push Notification
Create Account ,Login
Firebase Database
Firebase Cloud Messaging
Firebase Authentication
Read Users Data
Trigger Push Notification
Backend Database Actions to make preference actions on donations

Food Donation App as a Front End

Here, the Android Studio with some functionalities of Java is used as a front end with XML as a primary language. This is the module which happens to interact with that of the user to perform all the required GUI.

Firebase Data Base

The Firebase Real-time Database is a cloud-hosted database in which data is stored as JSON. The data is synchronized in real-time to every connected client. All of our clients share one Real-time Database instances and automatically receive updates with the newest data, when we build cross-platform applications with our iOS, and JavaScript SDKs.

The Firebase Real-time Database is a NoSQL database from which we can store and sync the data between our users in real-time. It is a big JSON object which the developers can manage in real-time. By using a single API, the Firebase database provides the application with the current value of the data and updates to that data. Real-time syncing makes it easy for our users to access their data from any device, be it web or mobile.

The Real-time database helps our users collaborate with one another. It ships with mobile and web SDKs, which allow us to build our app without the need for servers. When our users go offline, the Real-time Database SDKs use local cache on the device for serving and storing changes. The local data is automatically synchronized, when the device comes online.

Firebase Cloud Messaging

Firebase Cloud Messaging (FCM) is a cross-platform cloud messaging solution that lets you reliably send messages at no cost. In simple words, we can say that FCM is a highly sophisticated and advanced version of GCM.

FCM provides a reliable and battery-efficient connection between your server and devices that allows you to deliver and receive messages via notifications on iOS, Android, and the web at no cost. It can be used in our app for the live messaging purpose for apps good working.

Firebase Authentication

Food Donation App requires some level of authorization system. In some cases, validating a username/password set with our Users table is enough, but often, we need a more fine-grained permissions model to allow certain users to access certain resources and restrict them from others. Building a system to support the latter is not trivial and can be very time consuming. In this tutorial, we'll learn how to build a role-based auth API using Firebase, which will help us get quickly up and running. In this authorization model, access is granted to roles, instead of specific users, and a user can have one or more depending on how you design your permission model. Resources, on the other hand, require certain roles to allow a user to execute it.

Backend Database

The data which has got fetched from the front end and from the user actions is got stored in the backend for the further processing and analyzing which can be used in future cases.

5.2 Module Design Specification

Login & Registration

This phase involves login & registration for both the admin and Agent. The user's details are maintained confidential by maintaining separate account for each user. At the same time only, the admin can view the details of the registered agent.

Notification

This phase involves the notification to the agent by the admin. The user will send the notification which contains the location of food available via notification bar. This is achieved by using notification button.

Admin Module

In admin module, the administrator maintains the agent details as well as the donator details. The administrator collects the details on the whole. The administrator gives the orphanage details directly to the donator.

Donator Module

In donator module, the donator gives the wastage of food to the orphanage. The donator gives the request to the admin for the purpose of to collect the wastage food. The donator views the orphanage details and agent details.

Receiver Module

In Agent module, the Receiver maintain the orphanage details. It can also maintain the donator details. The Receiver give the request to the admin for collect the food from the donator. After collect the food the agent gives the alert message for the donator.

5.3 Program Design

Android Application

Android apps are written in the Java programming language and use Java core libraries. Developers may download the Android software development kit (SDK) from the Android website. The SDK includes tools, sample code and relevant documents for creating Android apps. Novice developers who simply want to play around with Android programming can make use of the App Inventor. Using this online application, a user can construct an Android app as if putting together pieces of a puzzle.

Java Programming Language

Java is a 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 meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to byte code 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. 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.

XML

Extensible Markup Language (XML) is a markup language that defines a set of rules for encoding documents in a format that is both human-readable and machine-readable. The design goals of XML emphasize simplicity, generality, and usability across the Internet.It is a textual data format with strong support via Unicode for different human languages. Although the design of XML focuses on documents, the language is widely used for the representation of arbitrary data structures such as those used in web services.Several schema systems exist to aid in the definition of XML-based languages, while programmers have developed many application programming interfaces (APIs) to aid the processing of XML data.

CHAPTER 6
SYSTEM IMPLEMENTATION

Java Code

DisplayActivity.java

```
package com.example.feed;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.net.Uri;
import android.os.Bundle;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;

public class DisplayActivity extends AppCompatActivity {

    private EditTextuname, fname, quantity, number;
    private Button btn;

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

```
uname = (EditText) findViewById(R.id.etname);
fname = (EditText) findViewById(R.id.etfoodname);
        quantity = (EditText) findViewById(R.id.etquantity);
        number = (EditText) findViewById(R.id.etnum);

        Intent i = getIntent();

        String name = i.getStringExtra("name");
        String food = i.getStringExtra("food");
        String quant = i.getStringExtra("quant");
        String num = i.getStringExtra("num");
        Double x =i.getDoubleExtra("lat",0);
        Double y =i.getDoubleExtra("lng",0);

uname.setText("Donor Name- "+name);
fname.setText("Food Name- "+food);
quantity.setText("Food Quantity- "+quant);
number.setText("Donor Number- "+num);

uname.setEnabled(false);
fname.setEnabled(false);
quantity.setEnabled(false);
number.setEnabled(false);

        final String address = "geo:"+String.valueOf(x)+","+String.valueOf(y);
btn = (Button) findViewById(R.id.btnloc);
```

```
btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {

                Uri gmmIntentUri = Uri.parse(address);
                Intent mapIntent = new Intent(Intent.ACTION_VIEW, gmmIntentUri);
mapIntent.setPackage("com.google.android.apps.maps");
                if (mapIntent.resolveActivity(getPackageManager()) != null) {
startActivity(mapIntent);
                }

//                Intent intent = new Intent();
//                intent.setAction(Intent.ACTION_VIEW);
//                intent.setData(Uri.parse(address));
//                startActivity(intent);
            }
        });

    }
}
```

DonateActivity.java

```
package com.example.feed;

import androidx.annotation.NonNull;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import android.Manifest;
import android.app.Activity;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.location.Location;
import android.os.Build;
import android.os.Bundle;
import android.text.TextUtils;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;

public class DonateActivity extends AppCompatActivity {

    private EditTextuname, fname, quantity, number;
    private DatabaseReferencereff;
    private Child member;
    private long maxid = 0;
    private Double lat,lng;
    @RequiresApi(api = Build.VERSION_CODES.M)
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_donate);

        if (!(ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED)) {
ActivityCompat.requestPermissions(this, new String[]
{Manifest.permission.ACCESS_FINE_LOCATION}, 123);
        }

uname = (EditText) findViewById(R.id.etname);
fname = (EditText) findViewById(R.id.etfoodname);
        quantity = (EditText) findViewById(R.id.etquantity);
        number = (EditText) findViewById(R.id.etnum);
        Button btn = (Button) findViewById(R.id.btnsubmit);
        member = new Child();

reff = FirebaseDatabase.getInstance().getReference().child("Child");
reff.addValueEventListener(new ValueEventListener() {
            @Override
            public void onDataChange(@NonNull DataSnapshotdataSnapshot) {
                if (dataSnapshot.exists()) {
maxid = dataSnapshot.getChildrenCount();
                }
            }
            @Override
            public void onCancelled(@NonNull DatabaseErrordatabaseError) {

            }
        });

btn.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
                String un, fn, qt, nm;

                un = uname.getText().toString().trim();
fn = fname.getText().toString().trim();
                qt = quantity.getText().toString().trim();
                nm = number.getText().toString().trim();

                if (TextUtils.isEmpty(un) || TextUtils.isEmpty(fn) || TextUtils.isEmpty(qt) ||
TextUtils.isEmpty(nm)) {
Toast.makeText(DonateActivity.this, "Fields must not be empty",
Toast.LENGTH_SHORT).show();
                }
else {
GeoLocationgeoLocation = new GeoLocation(getApplicationContext());

                    Location l = geoLocation.getLocation();

                    if (l != null) {
lat = l.getLatitude();
lng = l.getLongitude();

member.setUserName(un);
member.setFoodName(fn);
member.setQuantity(qt);
member.setNumber(nm);
member.setX(lat);
member.setY(lng);

reff.child(String.valueOf(maxid + 1)).setValue(member);

Toast.makeText(DonateActivity.this, "Donated", Toast.LENGTH_SHORT).show();

                        Intent intent = new Intent(DonateActivity.this, SecondActivity.class);
startActivity(intent);
finish();
                    }
                }
            }
        });
    }
}
```

FirstActivity.java

```
package com.example.feed;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import androidx.core.app.ActivityCompat;
import androidx.core.content.ContextCompat;
import android.Manifest;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.widget.Button;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.EditText;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;

public class FirstActivity extends AppCompatActivity {


    private static final String TAG = "FirstActivity";
    private FirebaseAuthmAuth;
    private EditTextETemail;
    private EditTextETpassword;
    long maxid=0;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_first);

        if (!(ContextCompat.checkSelfPermission(this,
Manifest.permission.ACCESS_FINE_LOCATION) ==
PackageManager.PERMISSION_GRANTED)) {
ActivityCompat.requestPermissions(this, new String[]
{Manifest.permission.ACCESS_FINE_LOCATION}, 123);
        }

mAuth = FirebaseAuth.getInstance();

ETemail = (EditText)findViewById(R.id.editText);
ETpassword = (EditText)findViewById(R.id.editText2);

        final Button signIn = (Button) findViewById(R.id.sign_in);
signIn.setOnClickListener(new OnClickListener() {
            @Override
            public void onClick(View view) {
signInIntoApp();
            }
        });

        Button signUp = (Button) findViewById(R.id.sign_up);
signUp.setOnClickListener(new OnClickListener() {
```

```
            @Override
            public void onClick(View view){
                    Intent intent = new Intent(FirstActivity.this, RegisterActivity.class);
startActivity(intent);
            }
        });
    }

 private void signInIntoApp(){
        String email = ETemail.getText().toString().trim();
        String password = ETpassword.getText().toString().trim();

if(TextUtils.isEmpty(email)){
Toast.makeText(this," Invalid Email ",Toast.LENGTH_SHORT).show();
            return;
        }

if(TextUtils.isEmpty(password)){
Toast.makeText(this," Invalid Password ",Toast.LENGTH_SHORT).show();
            return;
        }

mAuth.signInWithEmailAndPassword(email, password)
.addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if (task.isSuccessful()) {
```

```
                        // Sign in success, update UI with the signed-in user's information
Log.d(TAG, "signInWithEmail:success");
FirebaseUser user = mAuth.getCurrentUser();
                          Intent intent = new Intent(FirstActivity.this, SecondActivity.class);
startActivity(intent);
                    } else {
                        // If sign in fails, display a message to the user.
Log.w(TAG, "signInWithEmail:failure", task.getException());
Toast.makeText(FirstActivity.this," Authentication failed.",
Toast.LENGTH_SHORT).show();
                    }
                    // ...
                }
            });
    }

    @Override
    public void onStart() {
super.onStart();
        // Check if user is signed in (non-null) and update UI accordingly.
FirebaseUsercurrentUser = mAuth.getCurrentUser();
        //updateUI(currentUser);
    }
}
```

Geolocation.java

```
package com.example.feed;

import android.Manifest;
import android.content.Context;
import android.content.pm.PackageManager;
import android.location.Location;
import android.location.LocationListener;
import android.location.LocationManager;
import android.os.Bundle;
import android.widget.Toast;
import androidx.core.content.ContextCompat;

class GeoLocation implements LocationListener {

    Context context;

    public GeoLocation(Context c) {
this.context = c;
    }

    public Location getLocation() {

LocationManagerlm = (LocationManager)
context.getSystemService(Context.LOCATION_SERVICE);
        assert lm != null;
booleanonGPS = lm.isProviderEnabled(LocationManager.GPS_PROVIDER);
        if (onGPS) {

            if
(ContextCompat.checkSelfPermission(context,Manifest.permission.ACCESS_FINE_LOCATI
ON) != PackageManager.PERMISSION_GRANTED ){
                return null;
            }

lm.requestLocationUpdates(LocationManager.GPS_PROVIDER, 6000, 10, this);

            Location l = lm.getLastKnownLocation(LocationManager.GPS_PROVIDER);

            return l;
        }
else{
Toast.makeText(context,"Enable GPS",Toast.LENGTH_SHORT).show();
        }
        return null;
    }

    @Override
    public void onLocationChanged(Location location) {

    }
    @Override
    public void onStatusChanged(String s, int i, Bundle bundle) {

    }

    @Override
    public void onProviderEnabled(String s) {

    }

    @Override
    public void onProviderDisabled(String s) {

    }
}
```

MainActivity.java

```
package com.example.feed;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
```

```
public class MainActivity extends AppCompatActivity {

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

        new Handler().postDelayed(new Runnable() {
            @Override
            public void run() {
                Intent intent = new Intent(MainActivity.this, FirstActivity.class);
startActivity(intent);
finish();
            }
        },2000);
    }
}
```

ReceiveActivity.java

```
package com.example.feed;

import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;
import com.google.firebase.database.DataSnapshot;
import com.google.firebase.database.DatabaseError;
import com.google.firebase.database.DatabaseReference;
import com.google.firebase.database.FirebaseDatabase;
import com.google.firebase.database.ValueEventListener;
import java.util.ArrayList;

public class RecieveActivity extends AppCompatActivity {

    private DatabaseReferencereff;
    private ListViewmListView;
    private ArrayAdapter adapter;
    private ArrayList<String>arrayList;
    private long N=0;

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

Toast.makeText(RecieveActivity.this,"Please Wait.....",Toast.LENGTH_SHORT).show();

mListView = (ListView) findViewById(R.id.listview);
```

```
reff = FirebaseDatabase.getInstance().getReference().child("Child");
arrayList = new ArrayList<String>();

reff.addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(final DataSnapshotdataSnapshot) {
arrayList.clear();
           N=dataSnapshot.getChildrenCount();
for( inti = 1 ; i<= N ; i++ ){
arrayList.add(dataSnapshot.child(String.valueOf(i)).child("foodName").getValue(String.class));
           }

           adapter = new ArrayAdapter(RecieveActivity.this,
android.R.layout.simple_expandable_list_item_1,arrayList);
mListView.setAdapter(adapter);

mListView.setOnItemClickListener(new AdapterView.OnItemClickListener() {
              @Override
              public void onItemClick(AdapterView<?>adapterView, View view, int i, long l) {
                 Cache c = new Cache();
i++;

c.setUserName(dataSnapshot.child(String.valueOf(i)).child("userName").getValue(String.class)
);
c.setFoodName(dataSnapshot.child(String.valueOf(i)).child("foodName").getValue(String.class
));
c.setQuantity(dataSnapshot.child(String.valueOf(i)).child("quantity").getValue(String.class));

c.setNumber(dataSnapshot.child(String.valueOf(i)).child("number").getValue(String.class));
                        c.setX(dataSnapshot.child(String.valueOf(i)).child("x").getValue(Double.class));
                        c.setY(dataSnapshot.child(String.valueOf(i)).child("y").getValue(Double.class));

                        Intent intent = new Intent(RecieveActivity.this,DisplayActivity.class);

intent.putExtra("name",c.getUserName());
intent.putExtra("food",c.getFoodName());
intent.putExtra("quant",c.getQuantity());
intent.putExtra("num",c.getNumber());
intent.putExtra("lat",c.getX());
intent.putExtra("lng",c.getY());

startActivity(intent);
                    }
                });
            }

            @Override
            public void onCancelled(DatabaseErrordatabaseError) {

            }
        });
    }
}
```

RegisterActivity.java

```
package com.example.feed;

import androidx.annotation.NonNull;
import androidx.appcompat.app.AppCompatActivity;
import android.content.Intent;
import android.os.Bundle;
import android.text.TextUtils;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.EditText;
import android.widget.Toast;
import com.google.android.gms.tasks.OnCompleteListener;
import com.google.android.gms.tasks.Task;
import com.google.firebase.auth.AuthResult;
import com.google.firebase.auth.FirebaseAuth;
import com.google.firebase.auth.FirebaseUser;



public class RegisterActivity extends AppCompatActivity {

    private static final String TAG = ">";
    private EditTextETemail;
    private EditTextETpassword;
    private FirebaseAuthmAuth;
    protected void registerUser(){
        String email = ETemail.getText().toString().trim();
        String password = ETpassword.getText().toString().trim();

if(TextUtils.isEmpty(email)){
Toast.makeText(this," Invalid Email ",Toast.LENGTH_SHORT).show();
            return;
        }

if(TextUtils.isEmpty(password)){
Toast.makeText(this," Invalid Password ",Toast.LENGTH_SHORT).show();
            return;
        }

mAuth.signInWithEmailAndPassword(email, password)
.addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if (task.isSuccessful()) {
                            // Sign in success, update UI with the signed-in user's information
Log.d(TAG, "signInWithEmail:success");
FirebaseUser user = mAuth.getCurrentUser();

Toast.makeText(RegisterActivity.this," User already exists, Logging in... ",
Toast.LENGTH_SHORT).show();

                            Intent intent = new Intent(RegisterActivity.this, SecondActivity.class);
```

```
startActivity(intent);
finish();

                        } else {
createUser();
                        }

                        // ...
                    }
                })
    }

    protected void createUser(){

        String email = ETemail.getText().toString().trim();
        String password = ETpassword.getText().toString().trim();

mAuth.createUserWithEmailAndPassword(email, password)
.addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {
                    @Override
                    public void onComplete(@NonNull Task<AuthResult> task) {
                        if (task.isSuccessful()) {
                            // Sign in success, update UI with the signed-in user's information
Log.d(TAG, "createUserWithEmail:success");
FirebaseUser user = mAuth.getCurrentUser();
Toast.makeText(RegisterActivity.this,"Logging in... ", Toast.LENGTH_SHORT).show();
                            Intent intent = new Intent(RegisterActivity.this, SecondActivity.class);
```

```
startActivity(intent);
finish();
                        } else {
                            // If sign in fails, display a message to the user.
Log.w(TAG, "createUserWithEmail:failure", task.getException());
Toast.makeText(RegisterActivity.this, "Email already exists.",
Toast.LENGTH_SHORT).show();
                            // updateUI(null);
                        }

                        // ...
                    }
                });
    }

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

        // Initialize Firebase Auth
mAuth = FirebaseAuth.getInstance();

ETemail = (EditText)findViewById(R.id.email);
ETpassword = (EditText)findViewById(R.id.password);
```

```
        Button register = (Button) findViewById(R.id.register);
register.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) {
registerUser();
            }
        });
    }



  @Override
    public void onStart() {
super.onStart();
        // Check if user is signed in (non-null) and update UI accordingly.
FirebaseUsercurrentUser = mAuth.getCurrentUser();
        //updateUI(currentUser);
    }
}
```

XML Code

Activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.android.co
m/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=".MainActivity">


<ImageView
android:id="@+id/imageView"
android:layout_width="0dp"
android:layout_height="0dp"
android:background="#000010"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"

app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/logo"
        />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_display.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns: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=".DisplayActivity"
android:background="@drawable/new_login"
>

<EditText
android:id="@+id/etname"
android:layout_width="366dp"
android:layout_height="59dp"
android:ems="10"
android:inputType="textPersonName"
android:textSize="15sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.222"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.04"
    />

<EditText
android:id="@+id/etfoodname"
android:layout_width="366dp"
android:layout_height="59dp"
android:layout_marginStart="4dp"
android:layout_marginLeft="4dp"
```

```
android:ems="10"
android:inputType="textPersonName"
android:textSize="15sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.146"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/etname"
app:layout_constraintVertical_bias="0.034"
tools:ignore="HardcodedText" />

<EditText
android:id="@+id/etquantity"
android:layout_width="366dp"
android:layout_height="59dp"
android:ems="10"
android:inputType="textPersonName"
android:textSize="15sp"

app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.222"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/etfoodname"
app:layout_constraintVertical_bias="0.04000002" />
```

```
<EditText
android:id="@+id/etnum"
android:layout_width="366dp"
android:layout_height="59dp"
android:ems="10"
android:inputType="textPersonName"
android:textSize="15sp"

app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.222"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/etfoodname"
app:layout_constraintVertical_bias="0.207" />

<Button
android:id="@+id/btnloc"
android:layout_width="202dp"
android:layout_height="59dp"
android:text="View Location"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.454"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/etquantity"
app:layout_constraintVertical_bias="0.418" />
</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_donate.xml

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.constraintlayout.widget.ConstraintLayoutxmlns:android="http://schemas.android.co
m/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/new_login"
android:layout_height="match_parent"
tools:context=".DonateActivity">

<EditText
android:id="@+id/etname"
android:layout_width="366dp"
android:layout_height="59dp"
android:ems="10"
android:inputType="textPersonName"
android:hint="@string/your_name"
android:textSize="15sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.222"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:layout_constraintVertical_bias="0.04"
android:autofillHints="" />
```

```
<EditText
android:id="@+id/etfoodname"
android:layout_width="366dp"
android:layout_height="59dp"
android:layout_marginStart="4dp"
android:layout_marginLeft="4dp"
android:ems="10"
android:inputType="textPersonName"
android:hint="@string/food_name"
android:textSize="15sp"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.146"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/etname"
app:layout_constraintVertical_bias="0.034"
tools:ignore="HardcodedText" />

<EditText
android:id="@+id/etquantity"
android:layout_width="366dp"
android:layout_height="59dp"
android:ems="10"
android:inputType="textPersonName"
android:hint="@string/contact_number"
android:textSize="15sp"
```

```
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.222"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/etfoodname"
app:layout_constraintVertical_bias="0.04000002" />

<EditText
android:id="@+id/etnum"
android:layout_width="366dp"
android:layout_height="59dp"
android:ems="10"
android:hint="Food quantity"
android:inputType="textPersonName"
android:textSize="15sp"

app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.222"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/etfoodname"
app:layout_constraintVertical_bias="0.207" />

<Button
android:id="@+id/btnsubmit"
android:layout_width="202dp"
android:layout_height="59dp"
```

```
android:text="@string/donate"
app:layout_constraintBottom_toBottomOf="parent"
app:layout_constraintEnd_toEndOf="parent"
app:layout_constraintHorizontal_bias="0.454"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toBottomOf="@+id/etquantity"
app:layout_constraintVertical_bias="0.418" />

</androidx.constraintlayout.widget.ConstraintLayout>
```

Activity_receive.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayoutxmlns: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=".RecieveActivity"
android:orientation="vertical"
android:background="@drawable/new_login"
>

<ListView
android:id="@+id/listview"
android:layout_width="match_parent"
android:layout_height="match_parent"
```

```
android:theme="@style/TextAppearance.AppCompat.Large"
android:textSize="30sp"
/>
</RelativeLayout>
```

Register_Activity.xml

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayoutxmlns: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:orientation="vertical"
android:paddingLeft="18dp"
android:paddingRight="18dp"
android:background="@drawable/new_login"
android:gravity="center_horizontal"
tools:context="com.example.feed.RegisterActivity">
<ImageView
android:id="@+id/imageView2"
android:layout_width="match_parent"
android:layout_height="150dp"
android:layout_marginTop="25dp"
android:layout_marginBottom="0dp"
app:layout_constraintBottom_toTopOf="@+id/editText"
app:layout_constraintEnd_toEndOf="parent"
```

```
app:layout_constraintHorizontal_bias="0.038"
app:layout_constraintStart_toStartOf="parent"
app:layout_constraintTop_toTopOf="parent"
app:srcCompat="@drawable/logo" />

<EditText
android:id="@+id/email"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="100dp"
android:background="@android:color/transparent"
android:drawablePadding="12dp"
android:ems="10"
android:hint="Email"
android:inputType="textEmailAddress"
android:maxLines="1"
android:padding="8dp"
android:textColorHint="#fff" />

<View
android:layout_width="match_parent"
android:layout_height="1dp"
android:background="#f9d7db"/>
<View
android:layout_width="match_parent"
android:layout_height="1dp"
android:background="#f9d7db"/>
```

```
<View
android:layout_width="match_parent"
android:layout_height="1dp"
android:background="#f9d7db"/>

<EditText
android:id="@+id/password"
android:layout_width="match_parent"
android:layout_height="wrap_content"
android:layout_marginTop="20dp"
android:maxLines="1"
android:padding="8dp"
android:inputType="textPassword"
android:ems="10"
android:textColorHint="#fff"
android:drawablePadding="12dp"
android:background="@android:color/transparent"
android:hint="Password"/>


<View
android:layout_width="match_parent"
android:layout_height="1dp"
android:background="#f9d7db"/>
```

```
<Button
android:id="@+id/register"
android:layout_width="206dp"
android:layout_height="40dp"
android:layout_marginTop="80dp"
android:background="#8172"
android:textColor="#010203"
android:text="Register"/>

</LinearLayout>
```

CHAPTER 7

SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.
Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box. you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

Unit Testing:

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

Test strategy and approach

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

Test cases and Reports

TEST REPORT 1

New User Registration

Test Case ID	Test Case /Action To Perform	Expected Result	Actual Result	Pass/ Fail
1.	New user Registration	Displays “Registration Successful”	Registered Successfully	Pass
2.	Login to user Id	Displays “User Login successful”	User Login Correct	Pass

TEST REPORT 2

Login Process

Test Case ID	Test Case /Action To Perform	Expected Result	Actual Result	Pass/ Fail
1.	Enter the correct username and Password	Displays “Username and Password is correct”	Username and Password is Correct	Pass
2.	Click the “Submit” button	Displays “Login is successful”	Login is successful.	Pass

TEST REPORT 3

Donator page

Test Case ID	Test Case /Action To Perform	Expected Result	Actual Result	Pass/ Fail
1.	To check the donation details are uploaded well	Donation details are uploaded well.	Valid details	Pass

TEST REPORT 4

Receiver page

Test Case ID	Test Case /Action To Perform	Expected Result	Actual Result	Pass/ Fail
1.	To check the available food items are listed properly.	Available food items are listed well.	Valid list of available food items.	Pass

CHAPTER 8
CONCLUSION

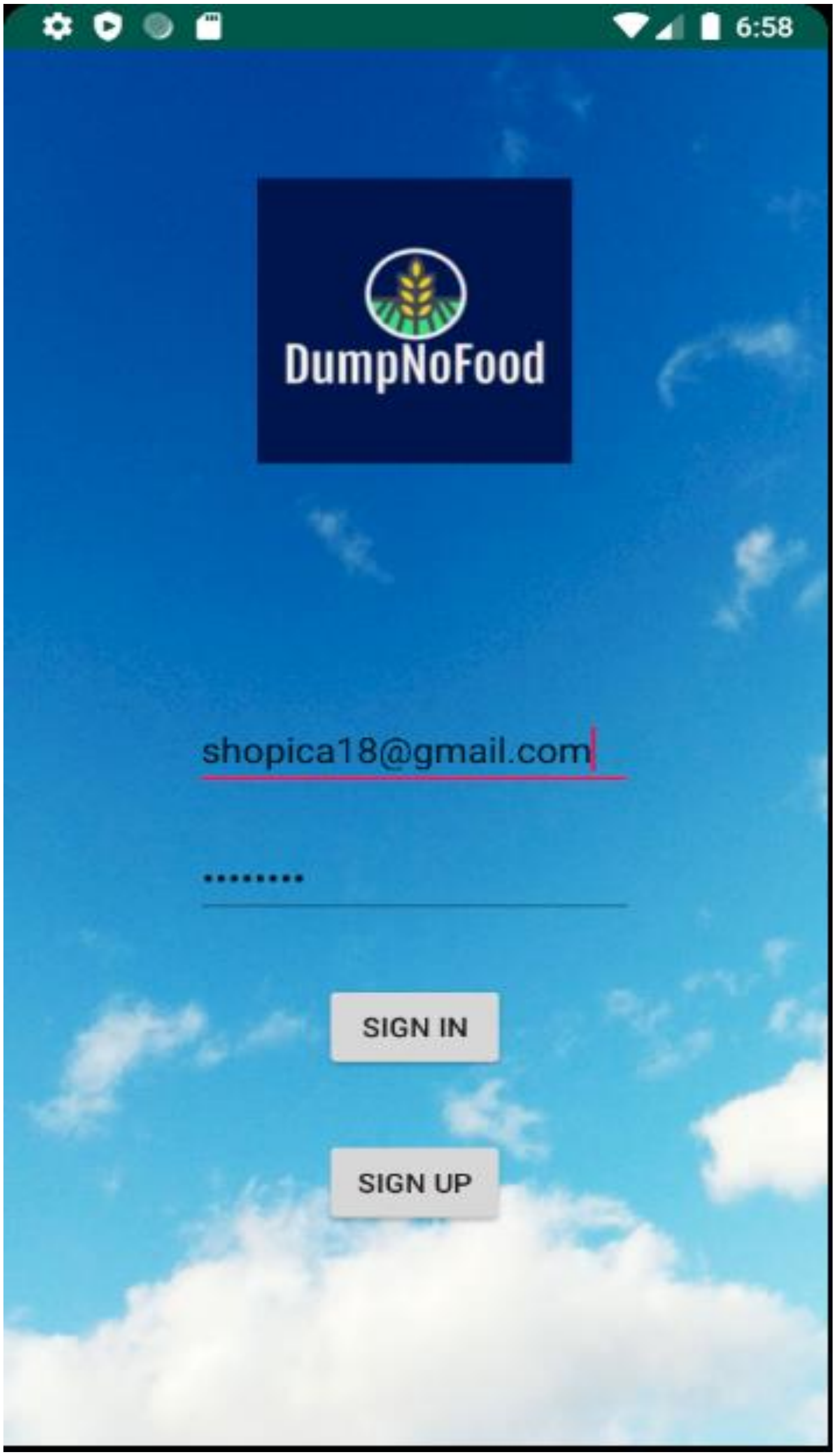
8.1 Conclusion and Future Enhancements

Thus, it's better understood how the proposed system is better than the prevailing system. The surplus food produced in functions, gatherings are often easily donated to the needy. Visualization of the impact of donation features a positive impact on the users. This is an attempt focused on feeding the hungry people and minimizing the food wastage problem at the same time. It will be compatible with everyone. In the existing system, there are many websites available that have an intermediary person between donor and recipient. Using this application the donor and recipient can communicate with each other directly. The Donor needs to send a request to the recipient. If they accept, food will be sent to them. This application is developed to avoid the food scarcity problem.

APPENDICES

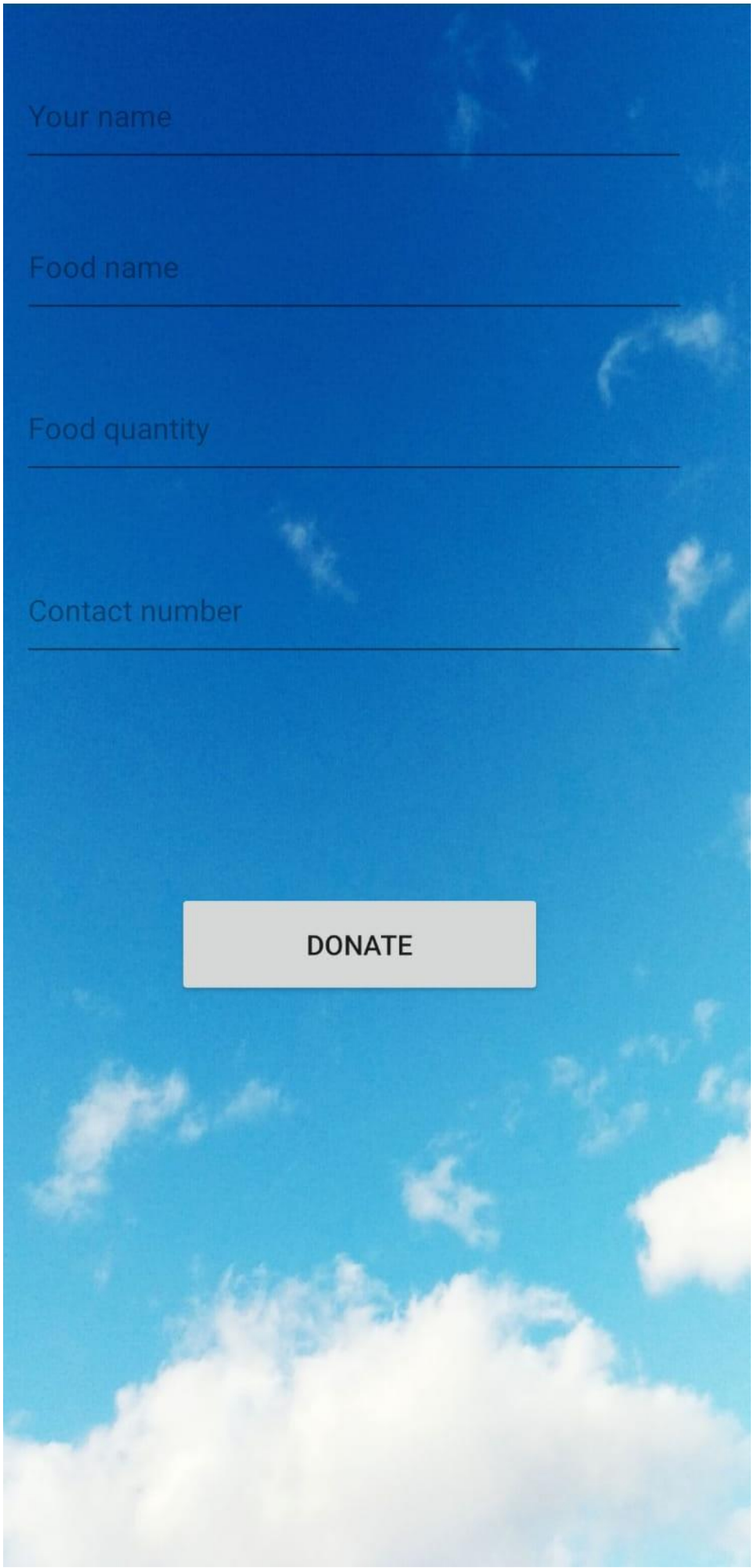
A.1 SAMPLE SCREENS

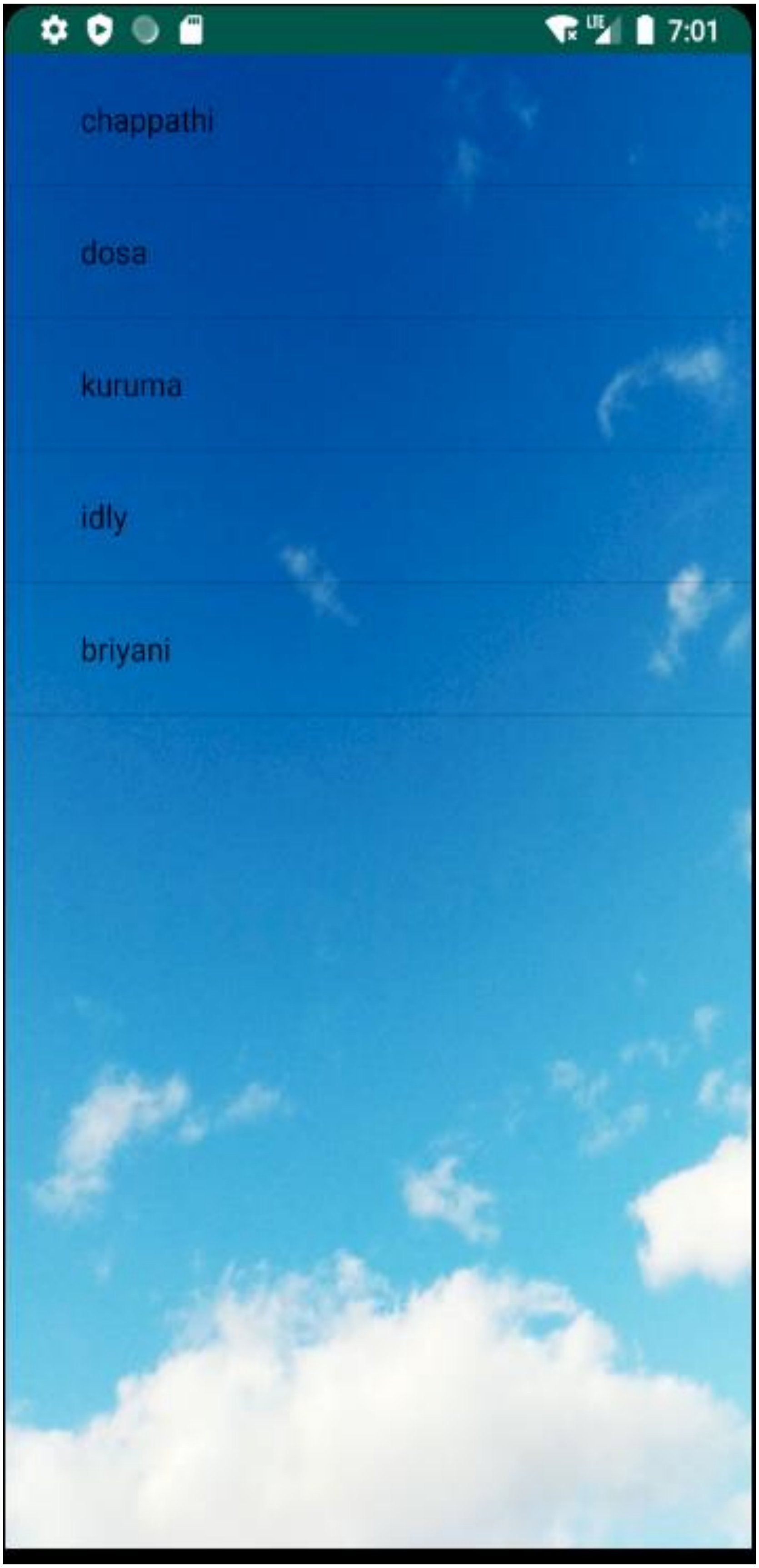
SCREEN SHOTS



shopica18@gmail.com

•••••••••


Your name
Food name
Food quantity
Contact number
DONATE



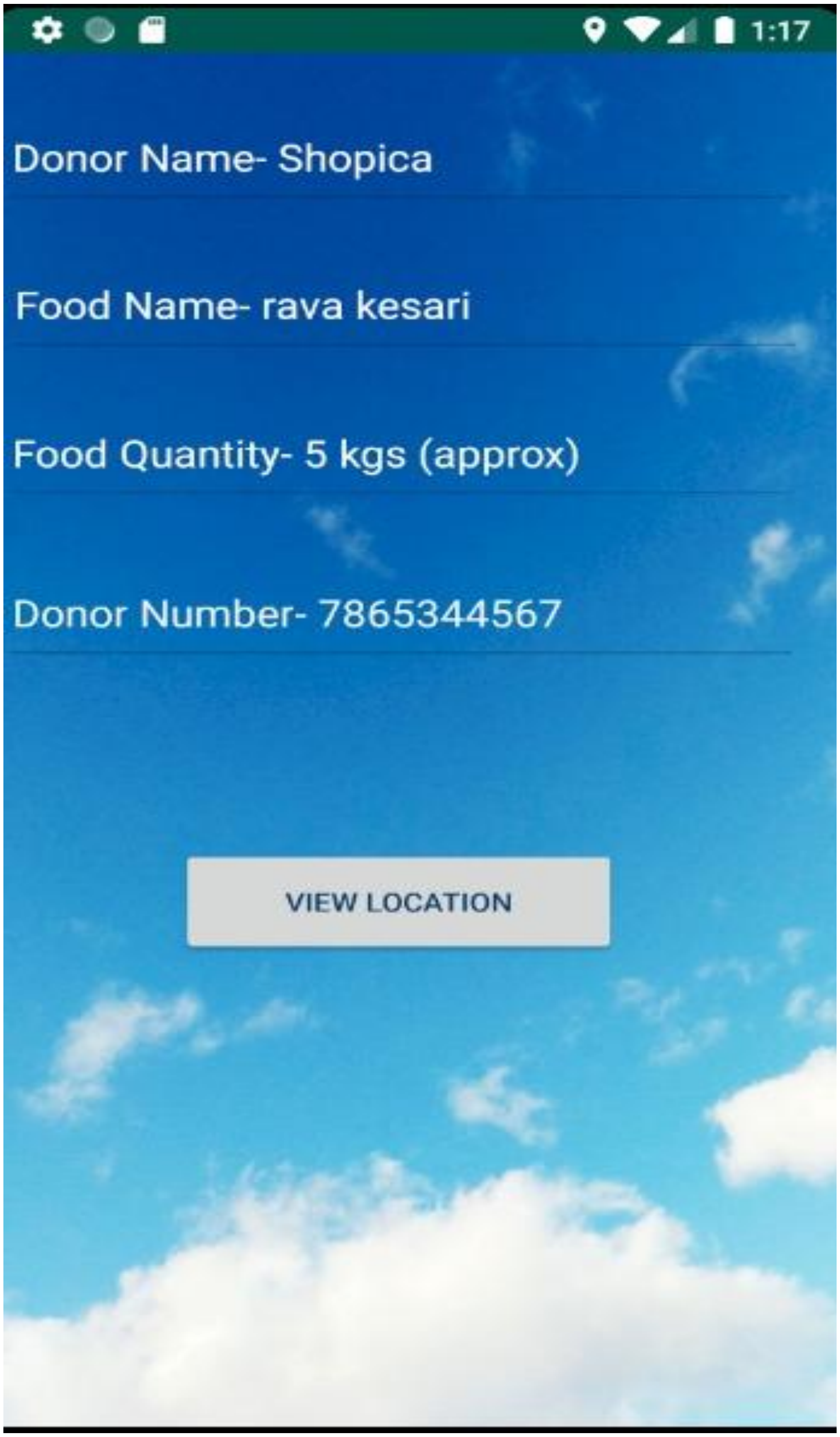
chappathi

dosa

kuruma

idly

briyani

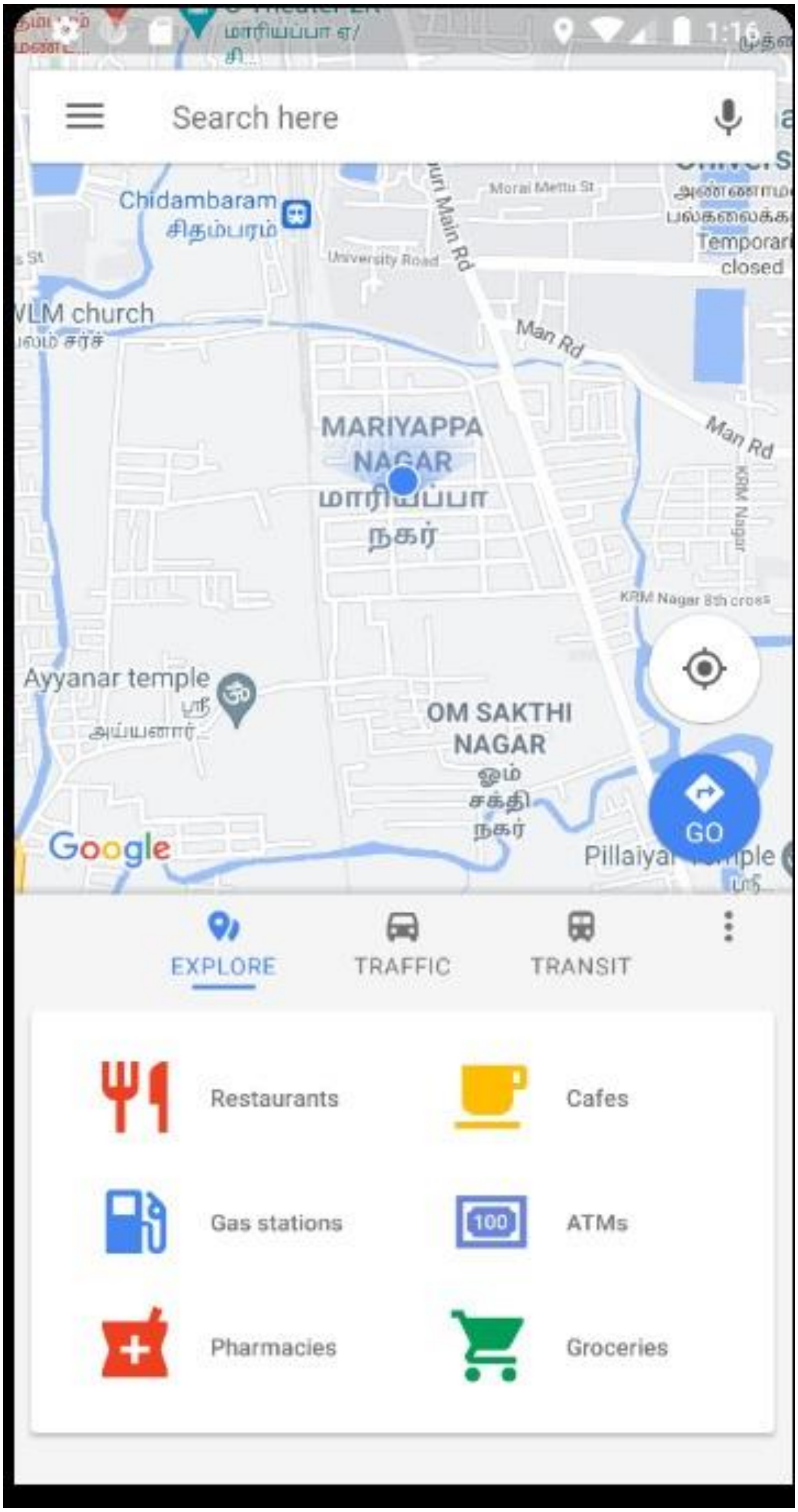


Donor Name- Shopica

Food Name- rava kesari

Food Quantity- 5 kgs (approx)

Donor Number- 7865344567


1:16
Search here
Chidambaram
சிதம்பரம்
University Road
Moral Mettu St
Temporarily
closed
VLM church
Man Rd
Man Rd
MARIYAPPA
NAGAR
மாரியப்பா
நகர்
KRM Nagar
KRM Nagar 8th cross
Ayyanar temple
OM SAKTHI
NAGAR
ஓம்
சக்தி
நகர்
Google
GO
Pillaiyar Temple
EXPLORE
TRAFFIC
TRANSIT
Restaurants
Cafes
Gas stations
100
ATMs
Pharmacies
Groceries

A.2 PUBLICATIONS

Paper Title : Food Wastage Reduction by Donation to Charity Institutions

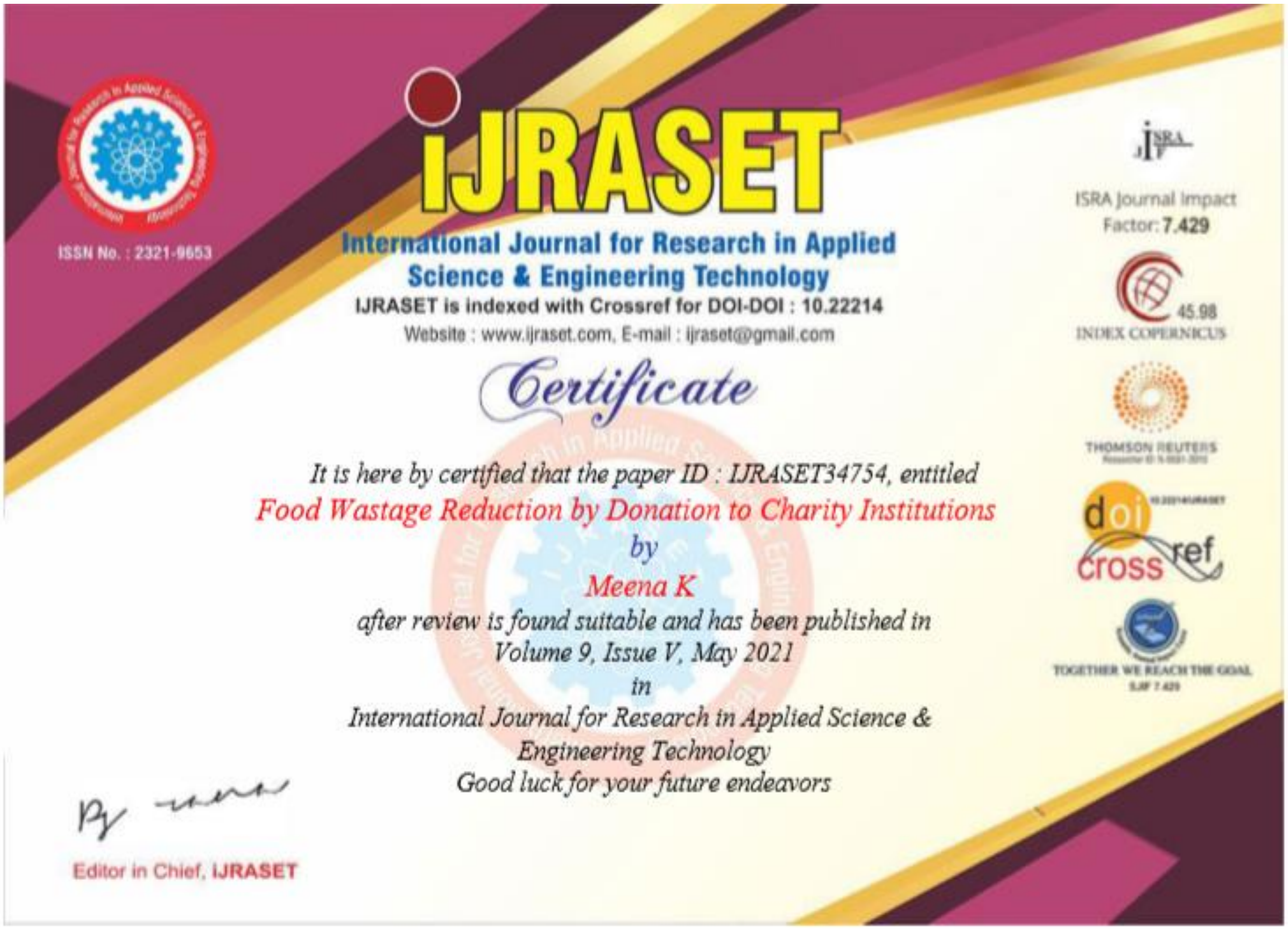
Published in :

International Journal for Research in Applied Science & Engineering Technology (IJRASET)

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429

Volume 9 Issue V May 2021

Available at https://www.ijraset.com/fileserve.php?FID=34754



ISSN No. : 2321-9653

iJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

Certificate

It is here by certified that the paper ID : IJRASET34754, entitled

Food Wastage Reduction by Donation to Charity Institutions

by

Meena K

after review is found suitable and has been published in

Volume 9, Issue V, May 2021

in

International Journal for Research in Applied Science & Engineering Technology

Good luck for your future endeavors

Editor in Chief, IJRASET

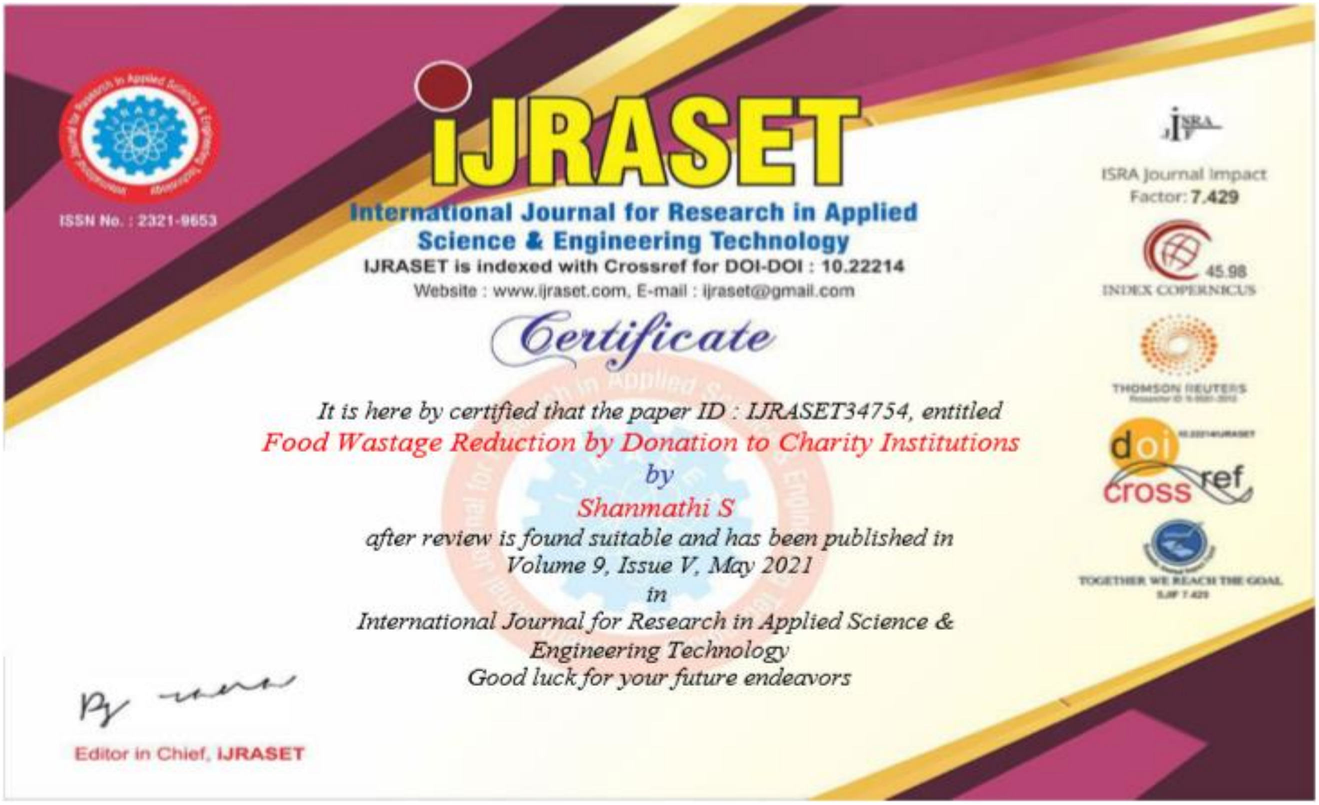
ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS

TOGETHER WE REACH THE GOAL

SJIF 7.429



ISSN No. : 2321-9653

IJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET34754, entitled

Food Wastage Reduction by Donation to Charity Institutions

by

Shanmathi S

after review is found suitable and has been published in

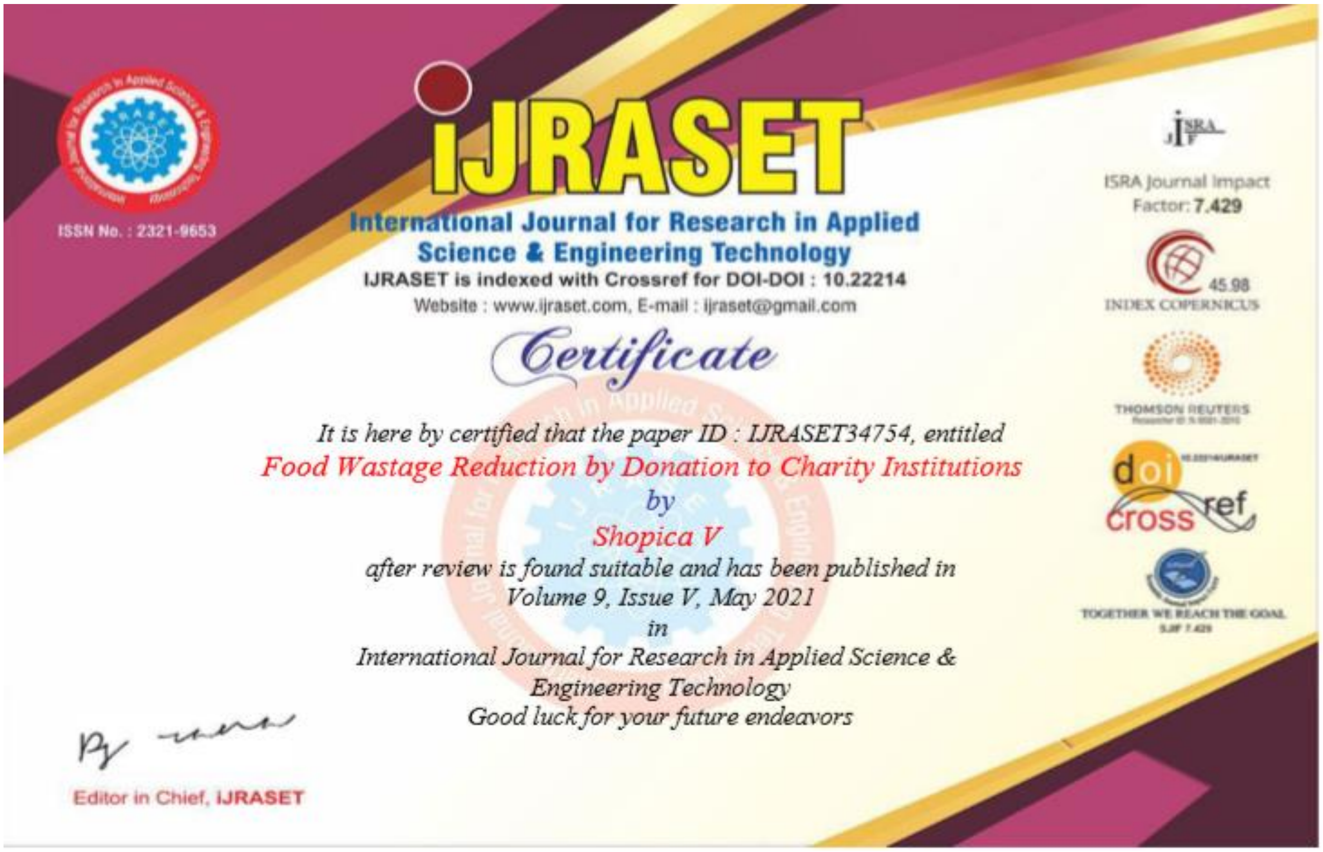
Volume 9, Issue V, May 2021

in

International Journal for Research in Applied Science & Engineering Technology

Good luck for your future endeavors

Editor in Chief, IJRASET



ISSN No. : 2321-9653

IJRASET

International Journal for Research in Applied Science & Engineering Technology

IJRASET is indexed with Crossref for DOI-DOI : 10.22214

Website : www.ijraset.com, E-mail : ijraset@gmail.com

ISRA Journal Impact Factor: 7.429

45.98 INDEX COPERNICUS

THOMSON REUTERS

TOGETHER WE REACH THE GOAL SJIF 7.429

Certificate

It is here by certified that the paper ID : IJRASET34754, entitled

Food Wastage Reduction by Donation to Charity Institutions

by

Shopica V

after review is found suitable and has been published in

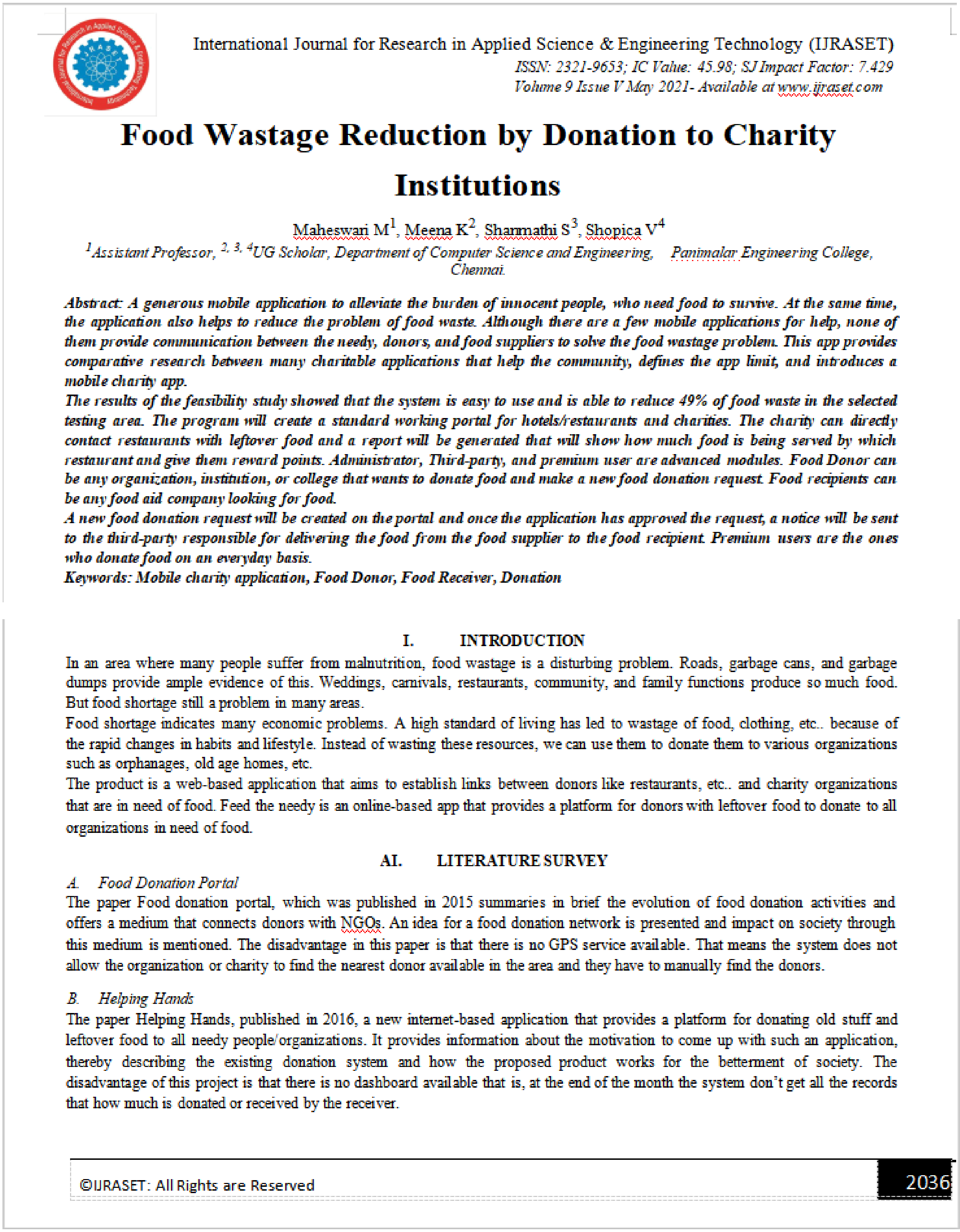
Volume 9, Issue V, May 2021

in

International Journal for Research in Applied Science & Engineering Technology

Good luck for your future endeavors

Editor in Chief, IJRASET



International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue V May 2021- Available at www.ijraset.com

Food Wastage Reduction by Donation to Charity Institutions

Maheswari M[1], Meena K[2], Sharmathi S[3], Shopica V[4]
[1]Assistant Professor, [2, 3, 4]UG Scholar, Department of Computer Science and Engineering, Panimalar Engineering College, Chennai.

Abstract: A generous mobile application to alleviate the burden of innocent people, who need food to survive. At the same time, the application also helps to reduce the problem of food waste. Although there are a few mobile applications for help, none of them provide communication between the needy, donors, and food suppliers to solve the food wastage problem. This app provides comparative research between many charitable applications that help the community, defines the app limit, and introduces a mobile charity app.

The results of the feasibility study showed that the system is easy to use and is able to reduce 49% of food waste in the selected testing area. The program will create a standard working portal for hotels/restaurants and charities. The charity can directly contact restaurants with leftover food and a report will be generated that will show how much food is being served by which restaurant and give them reward points. Administrator, Third-party, and premium user are advanced modules. Food Donor can be any organization, institution, or college that wants to donate food and make a new food donation request. Food recipients can be any food aid company looking for food.

A new food donation request will be created on the portal and once the application has approved the request, a notice will be sent to the third-party responsible for delivering the food from the food supplier to the food recipient. Premium users are the ones who donate food on an everyday basis.

Keywords: Mobile charity application, Food Donor, Food Receiver, Donation

I. INTRODUCTION

In an area where many people suffer from malnutrition, food wastage is a disturbing problem. Roads, garbage cans, and garbage dumps provide ample evidence of this. Weddings, carnivals, restaurants, community, and family functions produce so much food. But food shortage still a problem in many areas.

Food shortage indicates many economic problems. A high standard of living has led to wastage of food, clothing, etc.. because of the rapid changes in habits and lifestyle. Instead of wasting these resources, we can use them to donate them to various organizations such as orphanages, old age homes, etc.

The product is a web-based application that aims to establish links between donors like restaurants, etc.. and charity organizations that are in need of food. Feed the needy is an online-based app that provides a platform for donors with leftover food to donate to all organizations in need of food.

AI. LITERATURE SURVEY

A. Food Donation Portal

The paper Food donation portal, which was published in 2015 summaries in brief the evolution of food donation activities and offers a medium that connects donors with NGOs. An idea for a food donation network is presented and impact on society through this medium is mentioned. The disadvantage in this paper is that there is no GPS service available. That means the system does not allow the organization or charity to find the nearest donor available in the area and they have to manually find the donors.

B. Helping Hands

The paper Helping Hands, published in 2016, a new internet-based application that provides a platform for donating old stuff and leftover food to all needy people/organizations. It provides information about the motivation to come up with such an application, thereby describing the existing donation system and how the proposed product works for the betterment of society. The disadvantage of this project is that there is no dashboard available that is, at the end of the month the system don't get all the records that how much is donated or received by the receiver.

C. Beyond Food sharing: Supporting Food waste Reduction With ICT's

The paper Beyond food sharing: Supporting food waste reduction With ICTs', published in 2016, guaranteeing food security is key in improving the quality of life of citizens at all levels of society. The recent economic crisis has increased the number of people living in conditions of food poverty, especially in developed regions. Despite a growing awareness of the importance of reducing waste and managing food surplus, the role of ICTs in this domain is still unclear and rarely documented. This paper describes the use of ICT tools to recover food surplus at different stages of the supply chain and also describes the way forward for an integrated set of ICT tools to reduce waste from producers to households.

D. Food Donations Using a Forecasting-simulation Model

The paper Food donations using a forecasting-simulation model, published in 2016, presents a methodology to estimate donations for non-profit hunger relief organizations.

These organizations are committed to alleviating hunger around the world and depend mainly on the benevolence of donors to achieve their goals. However, the quantity and frequency of donations they receive varies considerably over time which presents a challenge in their fight to end hunger. A simulation model is developed to determine the expected quantity of food donations received per month in a multiware house distribution network.

The simulation model is based on a state-space model for exponential smoothing. A numerical study is performed using data from a non-profit hunger relief organization. The results show that good estimation accuracies can be achieved with this approach. Furthermore, non-profit hunger relief organizations can use the approach discussed in this paper to predict donations for proactive planning.

E. Smartphone Based Waste Food Supply Chain for Aurangabad City Using GIS Location Based and Google Web Services

The Paper Smartphone Based Waste Food Supply Chain for Aurangabad City Using GIS Location Based And Google Web Services, published in 2014, describes the client-server GIS and Smartphone application for the hunger free city. At the client side App provide facility to food to the charity for the help of hungry people. Donors enter basic information like latitude and long quantity of waste food and type of waste along with value and contact number. Charities can pick up that waste food and deliver food to hungers.

Completion of registration will placed onto server database where charities can store the entries of donor in table format and shows the optimal path between donor locations to nearest charity along with direction. So, wastage food can easily deliver to hungry people within a time.

F. Development of a Sustainable Food Supply Chain by Post Harvest Program

The paper Development of a Sustainable Food Supply Chain by Post Harvest Program introduces The Post Harvest Project (PHP), which is a group of committed technology and capital partners dedicated to delivering technological solutions to food waste in the supply chain. Of the many technological solutions being pursued by PHP, an innovative food preservation technology called nanoICE is described in detail.

An effort is underway in Ghana to build a series of small community food process plants based on improved cold storage on fishing boats and on-shore facilities for preserving fish, which in an important source of protein.

The aim of PHP's effort in Ghana is to engage local communities to feed people and create opportunities, and improve health through good nutrition.

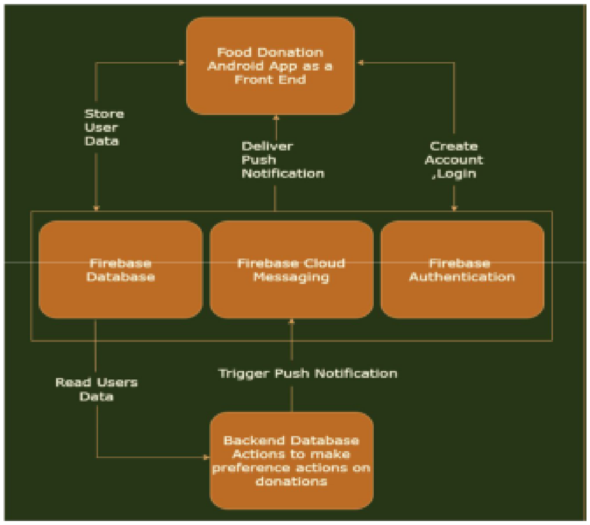
BI. EXISTING SYSTEM

At present, the need of the system is completed using websites which aren't accessible quicker and supply no awareness about the service to the planet. there's no real interaction between the donor and recipient since everything disclosed by intermediates. one more reason includes there's no active mobile application available in locality.

IV. PROPOSED SYSTEM

The system will make sure that the surplus food from functions, restaurants, and hotels goes to the hungry people instead of being thrown away resulting in food wastage. The system will act as a communication platform between the Donor and the Recipient. In our system, the Donor and the recipient can locate each other easily.

V. SYSTEM ARCHITECTURE


Food Donation Android App as a Front End
Store User Data
Deliver Push Notification
Create Account ,Login
Firebase Database
Firebase Cloud Messaging
Firebase Authentication
Read Users Data
Trigger Push Notification
Backend Database Actions to make preference actions on donations

A system architecture is a conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system .Here, the Android Studio with some functionalities of Java, CSS, and HTML is used as a front end with XML as a primary language. This is the module that happens to interact with that of the user to perform all the required GUI. The Firebase Realtime Database is a cloud-hosted database in which data is stored as JSON. The data is synchronized in real-time to every connected client.

VI. FUNCTIONALITIES

A. Login & Registration

This phase involves signing in and registering for both the donor and the recipient. The user's details are maintained confidentially by maintaining a separate account for each user. At the same time, the admin can view the details of the registered agent.

B. Donor Module

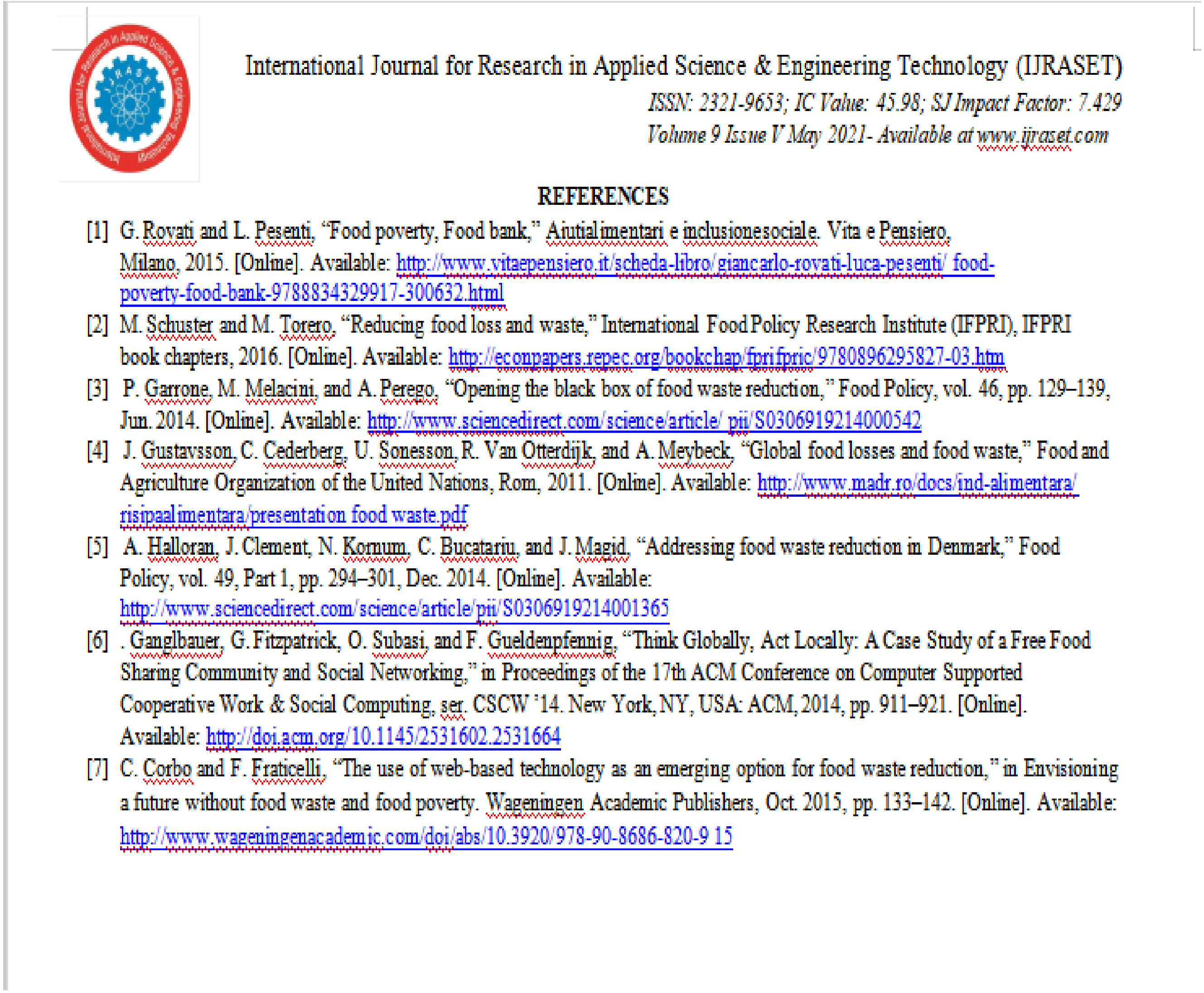
In the donor module, the donor gives the leftover food to the orphanage. The donor gives the request to the admin for the purpose of collecting the leftover food. The donor can view the orphanage details.

C. Receiver Module

In the Receiver module, the Receiver maintains the orphanage details. It can also view the donor details. The Receiver gives the request to the admin for collecting the food from the donor. After collecting the food, the recipient gives the alert message to the donor.

VII. CONCLUSION

Thus, it's better understood how the proposed system is better than the prevailing system. The surplus food produced in functions, gatherings are often easily donated to the needy. Visualization of the impact of donation features a positive impact on the users. This is an attempt focused on feeding the hungry people and minimizing the food wastage problem at the same time. It will be compatible with everyone. In the existing system, there are many websites available that have an intermediary person between donor and recipient. Using this application the donor and recipient can communicate with each other directly. The Donor needs to send a request to the recipient. If they accept, food will be sent to them. This application is developed to avoid the food scarcity problem.



REFERENCES

[1] G. Rovati and L. Pesenti, "Food poverty, Food bank," Aiutialimentari e inclusionesociale. Vita e Pensiero, Milano, 2015. [Online]. Available: http://www.vitaepensiero.it/scheda-libro/giancarlo-rovati-luca-pesenti/ food-poverty-food-bank-9788834329917-300632.html

[2] M. Schuster and M. Torero, "Reducing food loss and waste," International Food Policy Research Institute (IFPRI), IFPRI book chapters, 2016. [Online]. Available: http://econpapers.repec.org/bookchap/fprifpric/9780896295827-03.htm

[3] P. Garrone, M. Melacini, and A. Perego, "Opening the black box of food waste reduction," Food Policy, vol. 46, pp. 129–139, Jun. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/ pii/S0306919214000542

[4] J. Gustavsson, C. Cederberg, U. Sonesson, R. Van Otterdijk, and A. Meybeck, "Global food losses and food waste," Food and Agriculture Organization of the United Nations, Rom, 2011. [Online]. Available: http://www.madr.ro/docs/ind-alimentara/ risipaalimentara/presentation food waste.pdf

[5] A. Halloran, J. Clement, N. Kornum, C. Bucatariu, and J. Magid, "Addressing food waste reduction in Denmark," Food Policy, vol. 49, Part 1, pp. 294–301, Dec. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306919214001365

[6] . Ganglbauer, G. Fitzpatrick, O. Subasi, and F. Gueldenpfennig, "Think Globally, Act Locally: A Case Study of a Free Food Sharing Community and Social Networking," in Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing, ser. CSCW '14. New York, NY, USA: ACM, 2014, pp. 911–921. [Online]. Available: http://doi.acm.org/10.1145/2531602.2531664

[7] C. Corbo and F. Fraticelli, "The use of web-based technology as an emerging option for food waste reduction," in Envisioning a future without food waste and food poverty. Wageningen Academic Publishers, Oct. 2015, pp. 133–142. [Online]. Available: http://www.wageningenacademic.com/doi/abs/10.3920/978-90-8686-820-9 15

REFERENCES

[1] G. Rovati and L. Pesenti, "Food poverty, Food bank," Aiutialimentari e inclusionesociale. Vita e Pensiero, Milano, 2015. [Online]. Available: http://www.vitaepensiero.it/scheda-libro/giancarlo-rovati-luca-pesenti/ food-poverty-food-bank-9788834329917-300632.html

[2] M. Schuster and M. Torero, "Reducing food loss and waste," International Food Policy Research Institute (IFPRI), IFPRI book chapters, 2016. [Online]. Available: http://econpapers.repec.org/bookchap/fprifpric/9780896295827-03.htm

[3] P. Garrone, M. Melacini, and A. Perego, "Opening the black box of food waste reduction," Food Policy, vol. 46, pp. 129–139, Jun. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/ pii/S0306919214000542

[4] J. Gustavsson, C. Cederberg, U. Sonesson, R. Van Otterdijk, and A. Meybeck, "Global food losses and food waste," Food and Agriculture Organization of the United Nations, Rom, 2011. [Online]. Available: http://www.madr.ro/docs/ind-alimentara/ risipaalimentara/presentation food waste.pdf

[5] A. Halloran, J. Clement, N. Kornum, C. Bucatariu, and J. Magid, "Addressing food waste reduction in Denmark," Food Policy, vol. 49, Part 1, pp. 294–301, Dec. 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306919214001365

[6] T. Quested, E. Marsh, D. Stunell, and A. Parry, “Spaghetti soup: the complex world of food waste behaviours,” Resources, Conservation and Recycling, vol. 79, pp. 43–51, 2013.

[7] E. Ganglbauer, G. Fitzpatrick, O. Subasi, and F. Gueldenpfennig, “Think Globally, Act Locally: A Case Study of a Free Food Sharing Community and Social Networking,” in Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing, ser. CSCW ’14. New York, NY, USA: ACM, 2014, pp. 911–921. [Online]. Available: http://doi.acm.org/10.1145/2531602.2531664

[8] C. Corbo and F. Fraticelli, “The use of web-based technology as an emerging option for food waste reduction,” in Envisioning a future without food waste and food poverty. Wageningen Academic Publishers, Oct. 2015, pp. 133–142. [Online]. Available: http://www.wageningenacademic.com/doi/abs/10.3920/978-90-8686-820-915

[9] Q. Wu, W. Fang, and X. J. Wu, “Smart environment protection in wuxi,” School of IoT Engineering, Jiangnan University, Wuxi, China and China Working group of IEEE smart city of Wuxi, Tech. Rep., 2016.

[10] M. Hojer and J. Wangel, ¨ Smart Sustainable Cities: Definition and Challenges. Cham: Springer International Publishing, 2015, pp. 333– 349. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-09228-7 20

[11] L. Falasconi, M. Vittuari, A. Politano, and A. Segre, “Food ` Waste in School Catering: An Italian Case Study,” Sustainability, vol. 7, no. 11, pp. 14

745–14 760, Nov. 2015. [Online]. Available: http://www.mdpi.com/2071-1050/7/11/14745

[12] R. Sonnino and S. McWilliam, "Food waste, catering practices and public procurement: A case study of hospital food systems in Wales," Food Policy, vol. 36, no. 6, pp. 823–829, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306919211001163

[13] G. Farr-Wharton, J. H.-J. Choi, and M. Foth, "Food Talks Back: Exploring the Role of Mobile Applications in Reducing Domestic Food Wastage," in Proceedings of the 26th Australian Computer-Human Interaction Conference on Designing Futures: The Future of Design, ser. OzCHI '14. New York, NY, USA: ACM, 2014, pp. 352–361. [Online]. Available: http://doi.acm.org/10.1145/2686612.2686665