

MINI PROJECT REPORT ON
Food Wastage Preventing System

Submitted in partial fulfilment of the requirements for the degree of

BACHELOR OF COMPUTER APPLICATIONS Submitted by
M.Sowmiya 220027098

under the Guidance and Supervision of
{Dr. Priya Govindarajan
AP-I/CSE/SRC} }



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SHANMUGHA ARTS, SCIENCE, TECHNOLOGY & RESEARCH ACADEMY
SASTRA DEEMED UNIVERSITY
(A University under section 3 of the UGC Act, 1956) Srinivasa
Ramanujan Centre
Kumbakonam – 612001 Tamil Nadu, India

JULY 2020
SHANMUGHA ARTS, SCIENCE, TECHNOLOGY & RESEARCH ACADEMY
SASTRA DEEMED UNIVERSITY
(A University under section 3 of the UGC Act, 1956)
Srinivas Ramanujan Centre
Kumbakonam – 612001 Tamil Nadu, India



BONAFIDE CERTIFICATE

Certified that this project work titled “**Food Wastage Preventing System**” submitted to Srinivasa Ramanujan Centre, Shanmugha Arts, Science, Technology & Research Academy (SASTRA) Deemed to be University, Tirumalaisamudram-613401 by **M.Sowmiya-220027098, G.Abinaya-220027005, R.Sujitha-220027108** in partial fulfilment of the requirement for the award of the degree of **BACHELOR OF COMPUTER APPLICATIONS** carried out under our supervision, during the period December 2019 - July 2020.

Coordinators:

1. **Mr. H. Manikandan M.E.,**
Dept. of CSE,
Srinivasa Ramanujan Centre
2. **Dr. Priya Govindarajan**
Dept. of CSE,
Srinivasa Ramanujan Centre

HOD

Dr. S. Sivagurunathan
Dept. of CSE,
Srinivasa Ramanujan Centre

Submitted for University Examination held on 29.07.2020

EXAMINER - I **EXAMINER – II**
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SHANMUGHA ARTS, SCIENCE, TECHNOLOGY & RESEARCH ACADEMY
SASTRA DEEMED UNIVERSITY
(A University under section 3 of the UGC Act, 1956)

Srinivasa Ramanujan Centre
Kumbakonam – 612001
Tamil Nadu, India



DECLARATION

We submit this project work titled “**Food Wastage Preventing System**” to SASTRA Deemed to be University, Tirumalaisamudram–613 401, in partial fulfillment of the requirement for the award of the degree of **BACHELOR OF COMPUTERAPPLICATIONS**.

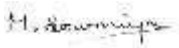
We declare that it is our work carried out under the Supervision and guidance of **Dr.Priya Govindarajan** Department of Computer Science and Engineering, Srinivasa Ramanujan Centre, Kumbakonam.

Date:29.07.2020

Place:Kumbakonam

Name : M. Sowmiya

Reg. No :220027098

Signature: 

ACKNOWLEDGEMENT

We pay our sincere obedience to GOD ALMIGHTY for their grace and infinite mercy and for showering on us their blessings.

We would like to express our sincere thanks to our Chairman **Prof. R.Sethuraman**, Vice Chancellor **Dr.S.Vaidhyasubramaniam** and Registrar **Dr.R.Chandramouli** for giving us an opportunity to be a student of this esteemed institution.

We express our deepest thanks to our revered Dean **Dr.V.Ramaswamy**, and **Dr.A.Alli Rani**, Associate Dean, Srinivasa Ramanujan Centre, SASTRA Deemed to be University for all their moral support and suggestions when required without any reservations.

We exhibit our pleasure in expressing our thanks to **Dr.S.Sivagurunathan**, Head of the Department, Computer Science and Engineering, Srinivasa Ramanujan Centre for his encouragement during our project work.

We would like to express our deep sense of gratitude to the project coordinators **Prof.H.Manikandan** and **Dr.Priya Govindarajan**, Assistant Professor, Department of Computer Science and Engineering, for their cordial support, and meticulous guidance which enabled us to complete this project successfully.

Without the support of our Parents and friends this project would never have become a reality. We owe our sincere thanks to them. We dedicate this work to all our well-wishers, with love and affection. Our sincere thanks to all faculty in the department who have directly or indirectly helped us in completing this project.

Abstract

ABSTRACT

Preventing the pre-consumed food from getting wasted is the primary goal of our **Food Wastage Preventing System** project. Our project intends to facilitate the Donors like Restaurant Owners, Wedding Ceremony Holders and other Individuals who are all willing to donate the excess amount of food, to those who need it through the Volunteers of NGOs like “**Feeding India**.” It will prevent the food from getting wasted because of not knowing Recipient’s location

Through this project it is possible to identify the location of the people who have excess food, and also possible to identify the location of people who need it by letting them to enrol themselves in this Application. The excess food will be carried out by the volunteer delivery person and will be delivered to the Recipient. Our project ensures that every person in hunger get feed by coordinating the Donor, Recipient and the volunteer delivery person into a single platform.

TABLE OF CONTENTS

Chapter. No.	CONTENTS	Page No.
1.	Introduction	10
2.	Software Project Plan 2.1 Software Project Plan 2.2 Objective of the Study 2.2.1 Existing System 2.2.2 Proposed System	11
3.	Software Requirements Specification 3.1 Introduction 3.2 Overall Description 3.3 System Features 3.4 Other Non-Functional Requirements	15
4.	System Analysis 4.1 Architecture Diagram 4.2 Use Case Diagram 4.3 Activity Diagram 4.4 Sequence Diagram 4.5 Class Diagram	20
5.	Design 5.1 Input Design 5.2 Output Design 5.3 Data Flow Diagram	26
6.	Testing 6.1. Unit Testing 6.2. Black Box testing 6.3. White Box testing 6.4. Integration testing 6.5. System Testing 6.6. End to End Testing 6.7. Usability Testing 6.8. Validation Testing	31

7.	Implementation 7.1 Modules and Description 7.2 Problems Faced 7.2 Lesson Learnt	34
8.	Conclusion & Future Enhancement 8.1 Future Enhancement 8.2 Conclusion	37
9.	References	41
10.	Appendix 10.1 Glossary 10.2 Coding	43

INTRODUCTION

1.INTRODUCTION

Pre consumed food wastage has been a serious problem faced by almost the entire world. By some estimation nearly 40% of food produced get wasted even before it reaches the plate, which could feed 2.6 crores of people. There is no doubt weddings and banquets are a huge source of food wastage, but restaurant and hotels also contribute to food wastage. But only some of them donate it to their staff and other personnel or to orphanages.

“Feeding India”, “Robinhood Army”, “GiftAMealInIndia” are some of the NGOs which were founded with the objective of eliminating hunger and connect hunger and food waste as solution for each other. They collect the excess food from individuals, weddings, canteens, hotels, restaurants and other events and redistribute it to the needy, free of cost.

SOFTWARE PROJECT PLAN

2.1. SOFTWARE PROJECT PLAN

S. No.	Phase / Milestone	Description	Time Bound	Deliverable	Remarks
1	Process Initialization	Various modules constituting the product are identified.	22-12-2019	Report containing the key activities	Implementation logic for various modules familiarized and the way to integrate them is also identified
2	Software Requirements / Estimation	Various software and hardware for implementing various modules are recognized and their various interface requirements are gathered.	17-01-2020	SRS (functional & Non-functional) Report, Person Cost & Resources Estimation	Various kinds of requirements Are identified
3	Software Architect Description & SQAP	The analysis model is converted into design model and different diagrams like, use-case, class, sequence, and collaboration diagrams are drawn.	05-02-2020	Design model constituting of various diagrams	Various kinds of diagrams required for the modules are drawn.
4	System Development	Various diagrams constructed are used for functional modules. Coding are written for various modules	10-03-2020	Functional modules and Integration	Coding is performed for various modules

S. No.	Phase / Milestone	Description	Time Bound	Deliverable	Remarks
5	Software Testing	Test plan, Various testing Methods like unit Testing, integration Testing, usability testing	14-04-2020	Test Report consisting of various testing.	Various kinds of tests are performed.
6	Software Deployment	The software is implemented in JAVA language using Android studio IDE.	20-05-2020	Product Deployment	The software has run successfully
7	Software Maintenance	The part of the product which requires maintenance is identified and their guidelines are listed	19-05-2020	Problem reporting, Solution, distribution, Defect prevention	The information required for proper maintenance of the Product is listed
8	Software Configuration and Risk Management	Creation of baseline various workplace for the product and the associated risks are identified	21-06-2020	Baseline, change control, record, risk, (Identification, Qualification & Migration)	Various workplace and possible risks are identified

2.2. Objective of the Study

- *Recipients can easily identify the donors. And similarly, volunteer delivery persons can identify the donors and recipients.*
- *Donors can easily post the food availability through this application.*
- *It is working on 24*7 facility*
- *Can be used to identify the food served and unserved areas easily*

2.2.1. Existing System

- *Finding the Donor with excess food is done by contacting each donor through mobile phone.*
- *For distributing the excess food among the recipients, delivery volunteers of the neighbourhood allocated permanently and if the person is not available for the service it needs to be informed manually.*
- *Above type of allocation will consume lot of time, effort and also decrease the efficiency of finding nearest active volunteers.*
- *A registered recipient can't request for the food from the donor and should depend on the volunteer delivery person's decisions.*
- *Individual donor with small quantity of food could not contribute the food through the existing system.*

2.2.2. Proposed Work

- *Registered Recipient can request for food from the donor who fulfil the quantity of food and type of food they need.*
- *Delivery volunteers for deliver task will be allocated based on their current location. The nearest one with respect to the donor's location will receive the delivery task.*
- *Donors can post the availability of excess food from their location at any time and remove them as well.*
- *Individual donors with small quantity of food can either hand it over to the nearest delivery volunteer or to the nearest Food Bank ATMs.*

SOFTWARE REQUIREMENT SPECIFICATION

3.1 Introduction

3.1.1 Data Requirements:

The set of data that is involved in any project is defined using data requirements. For this project, the main data required is the login information to register the application, location information and the post's information. Without this information the application cannot process the transaction

3.1.2 Functional Requirements:

Functional requirements are properties that must exist in the final system. For any mobile application, user need to download the application. To use the application, the user needs to register and login to the application after installing by providing login information. Once, he or she logs into the application, they can use all the features.

3.1.3 Performance Requirements:

Response time, scalability, platform dependencies, tolerance are the performance requirements that should be considered when developing any system. The application or system should be able to respond quickly when the user interacts with the application. The application should be developed in such a way that it should be scalable enough to accept new features when we want to expand the application complexity. The application 5 should run in all the specified software and hardware requirements from the design phase of the project. Also, the tolerance rate (fault tolerance) of the application should be at a higher level in case of network issues, connectivity issues, and when the application crashes or stops. It should be able to deliver the information about any of those issues to the user when the system is no longer able to provide results when the user wants.

3.1.4 System Requirements:

The application should be installed into a device, system or any machine in such a way that it should have basic requirements like supporting software and hardware of the device, accessing in-built software, say camera for mobile device, internet permissions, and potential security issues such as virus or any malware detection.

3.2 Overall Description Developer

Side:

Software Requirements for Development: - ○

Operating System: Windows 10 (64 bit) ○

IDE: Android Studio 3.5.3 version ○

Database: Firebase ○ Developing

language: Java JDK 8.0 version ○ User

Interface Language: XML ○ Maps API:

Mapbox Maps API Hardware Requirements

for Development: - ○ Processor:

AMD ○ RAM:

4GB RAM ○ Hard Disk:

4GB ○ Speed:

2.00 GHZ ○ Keyboard:

standard Windows Keyboard ○ Mouse:

2 or 3 Button Mouse

○ Monitor Resolution: 1366

****768***

User Side:

Software Requirements:

○ Android API level: 10 and above

3.3 System Features

3.2.1 IDE's, Tools and Technologies:

3.3.1.1. Android Studio

Android Studio [6] is exclusively designed for developing Android applications. It consists of all Android SDK tools to design, develop, maintain, test, debug and publish our app. The IDE is designed very efficiently which makes the developer's job easy. It also supports the IntelliJ IDE, the main idea behind this IDE is that it automatically senses the variables, methods, classes, built-in functions or it could be anything else when we press the first letter of it.

3.3.1.2. Android Software Development Kit (SDK):

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

3.3.1.3. Android Debug Bridge (ADB):

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

3.3.1.4. Gradle Build:

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

3.3.1.5. Android Device Monitor:

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

3.3.1.6. SDK Manager:

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

3.3.1.7. AVD Manager:

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

3.3.1.8. Firebase Database:

Android also supports Firebase Database which is a cloud-based database to develop application and perform any CRUD (Create, Update, and Delete) operations. It is much flexible and Data security; backup is more efficient.

3.4 Other Non-Functional Requirements

- 1. Application should be capable enough to handle many users without affecting its performance*
- 2. The application should be portable. So, moving from one device to other device does not create any problem.*
- 3. Privacy of information, the export of restricted technologies, intellectual property rights, etc. should be audited.*

3.4.1. Performance

- Easy tracking of records and updating can be done.*

3.4.2. Availability

- Application will be available to all the registered users.*

3.4.3. Security

- Since it is available for only registered users as well as only the administrator has the rights to access the user records all data will be secured other users cannot access it.*

3.4.4. Maintainability

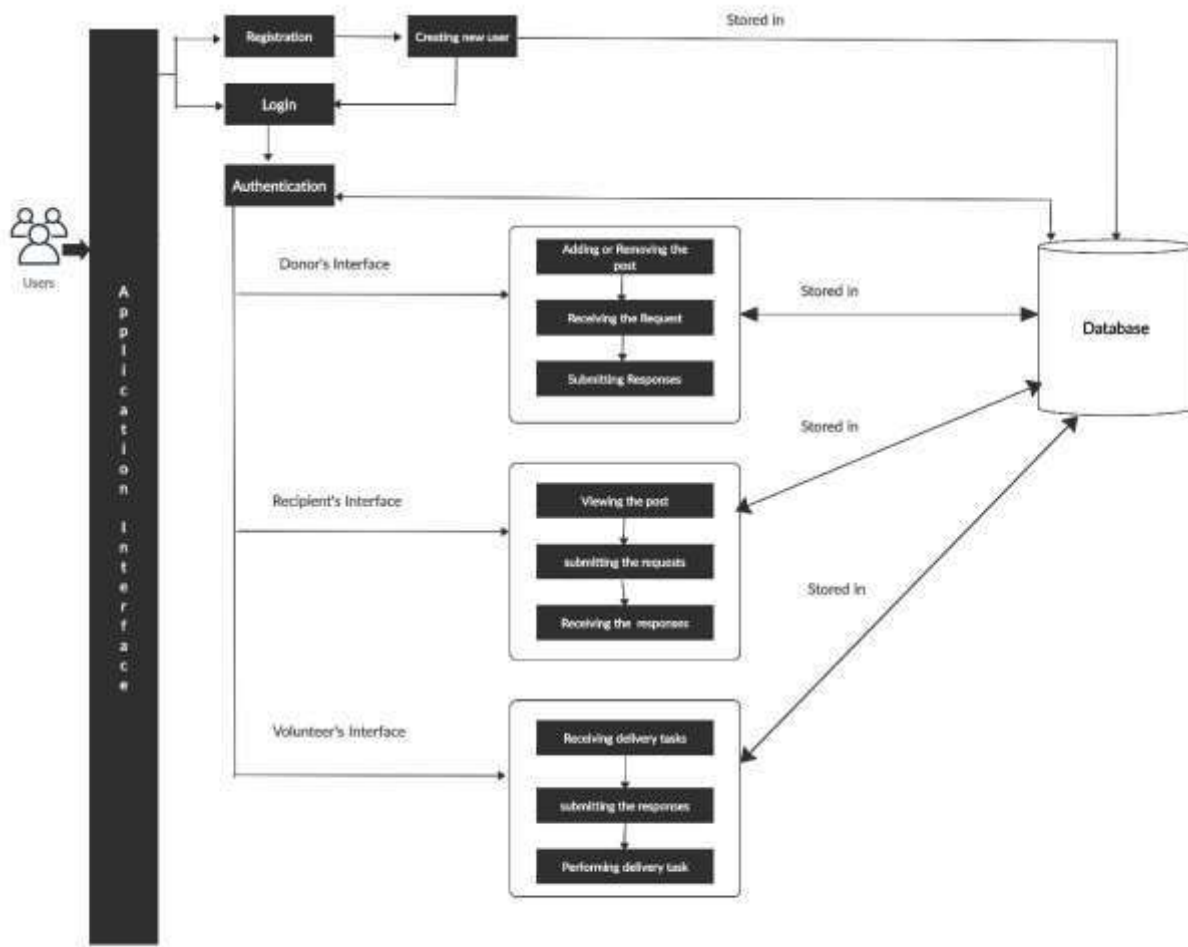
- Backup and records are stored in firebase cloud, so it is easy to maintain*

3.4.5. Portability

- This application is designed in android studio IDE so all the android operating system users can use it.*

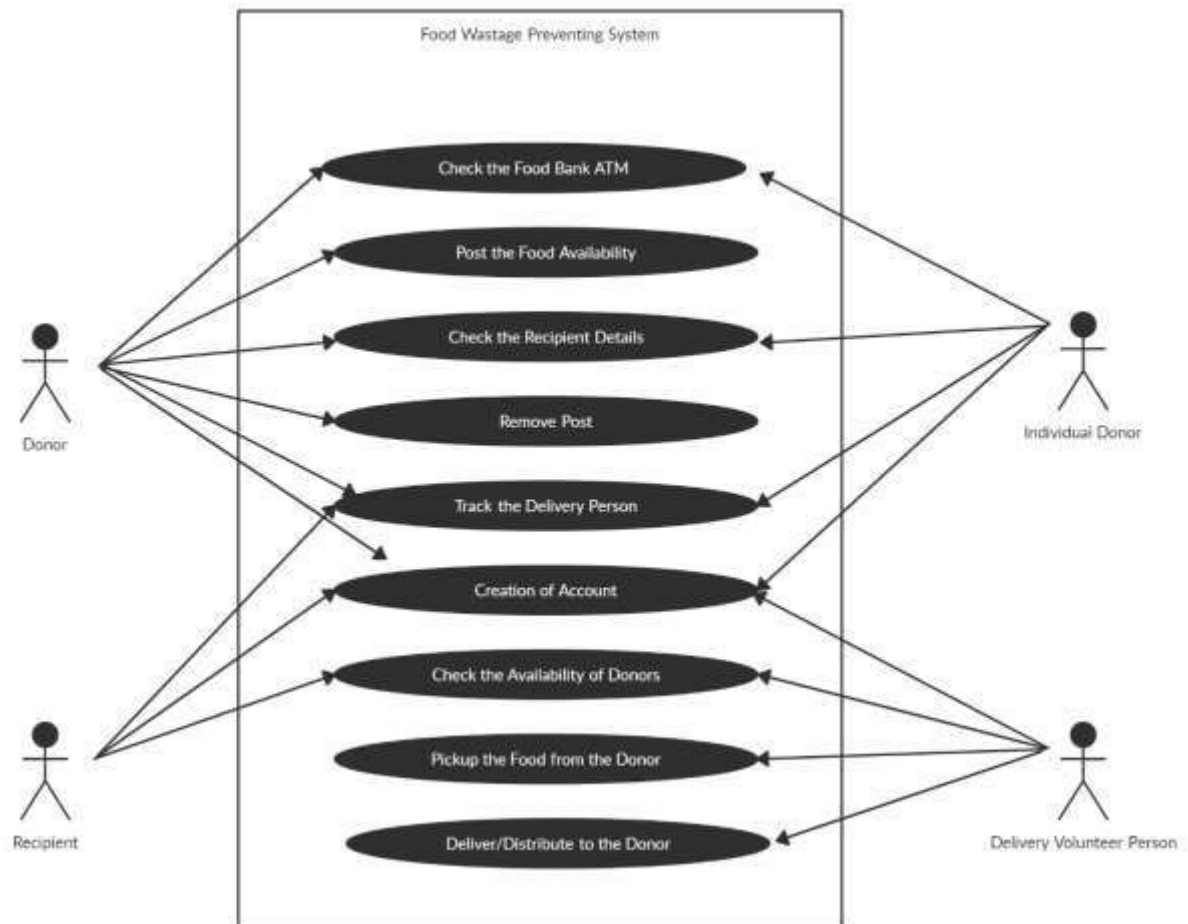
SYSTEM ANALYSIS

4.1. Architecture Diagram

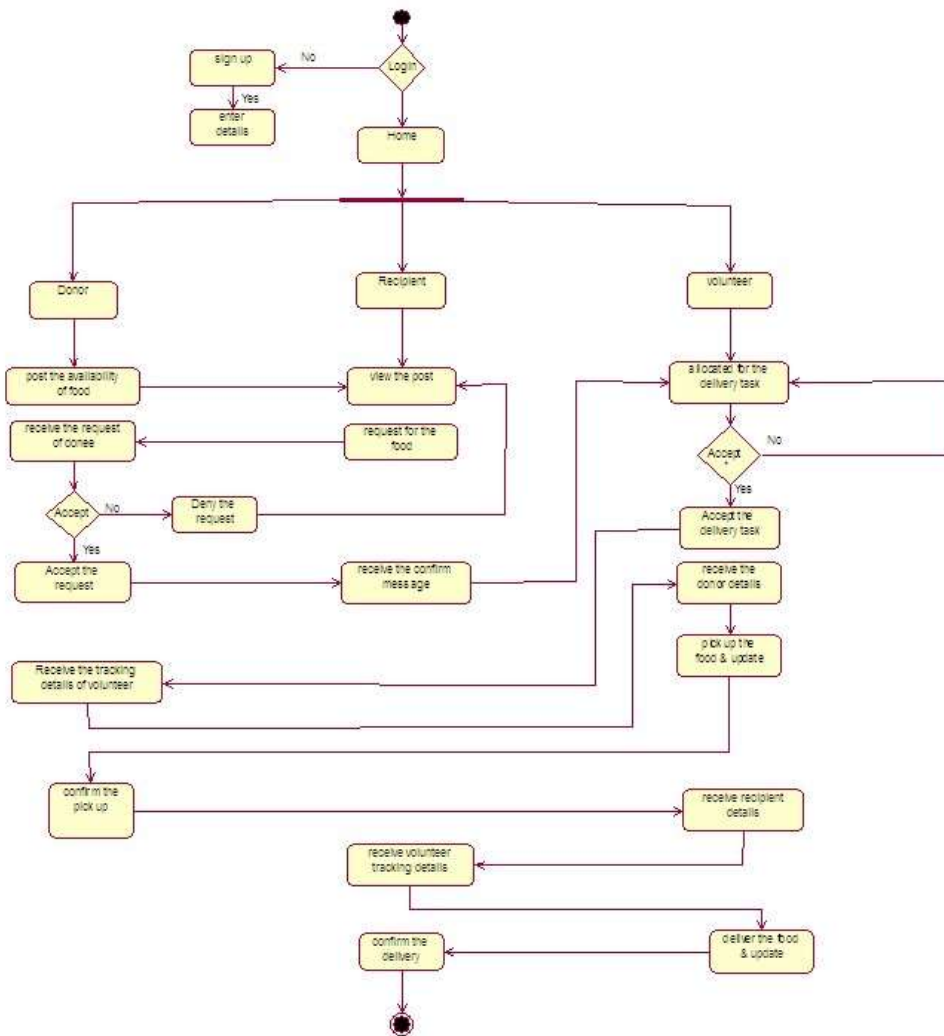


Architecture Diagram

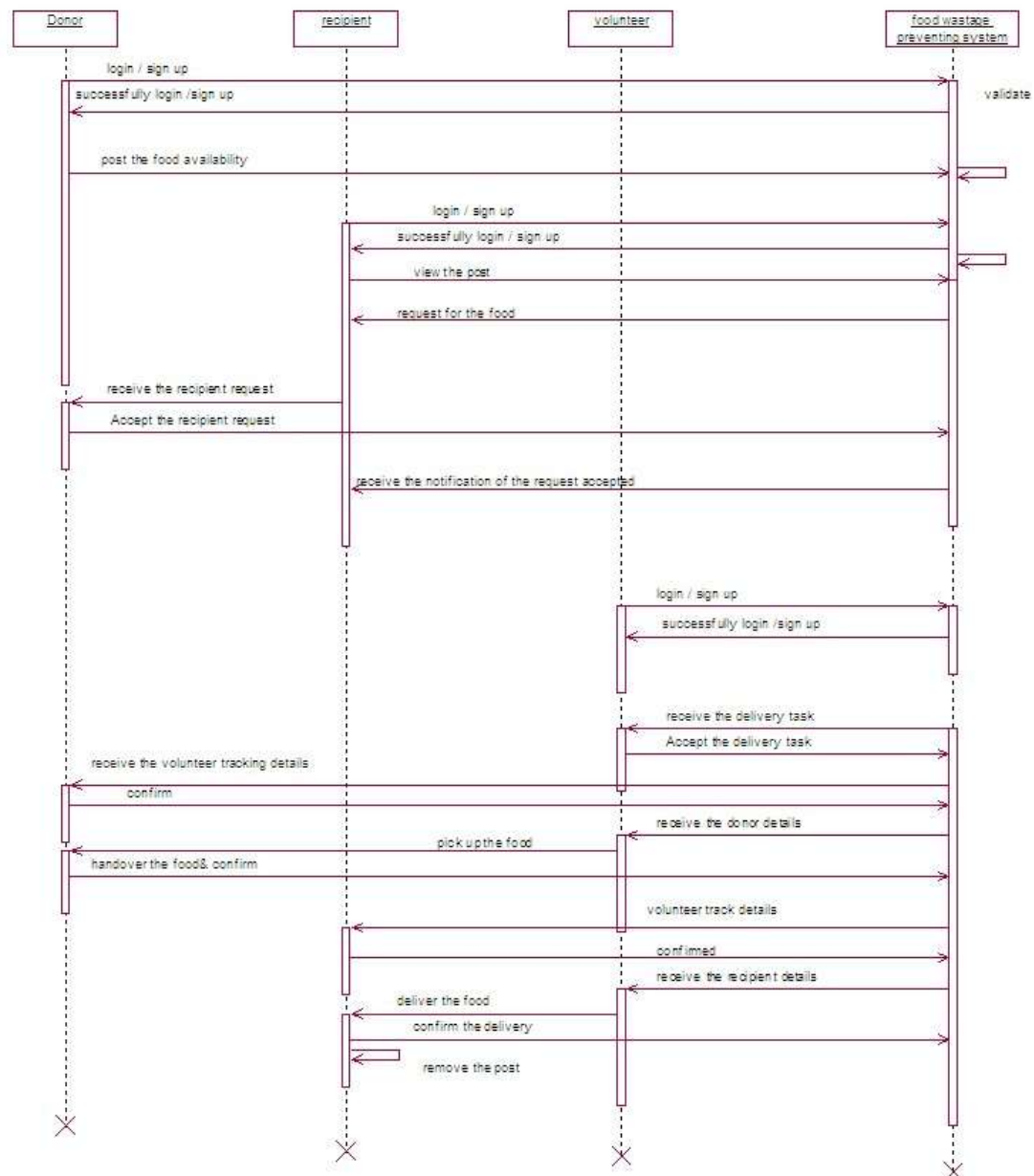
4.2. Use Case Diagram



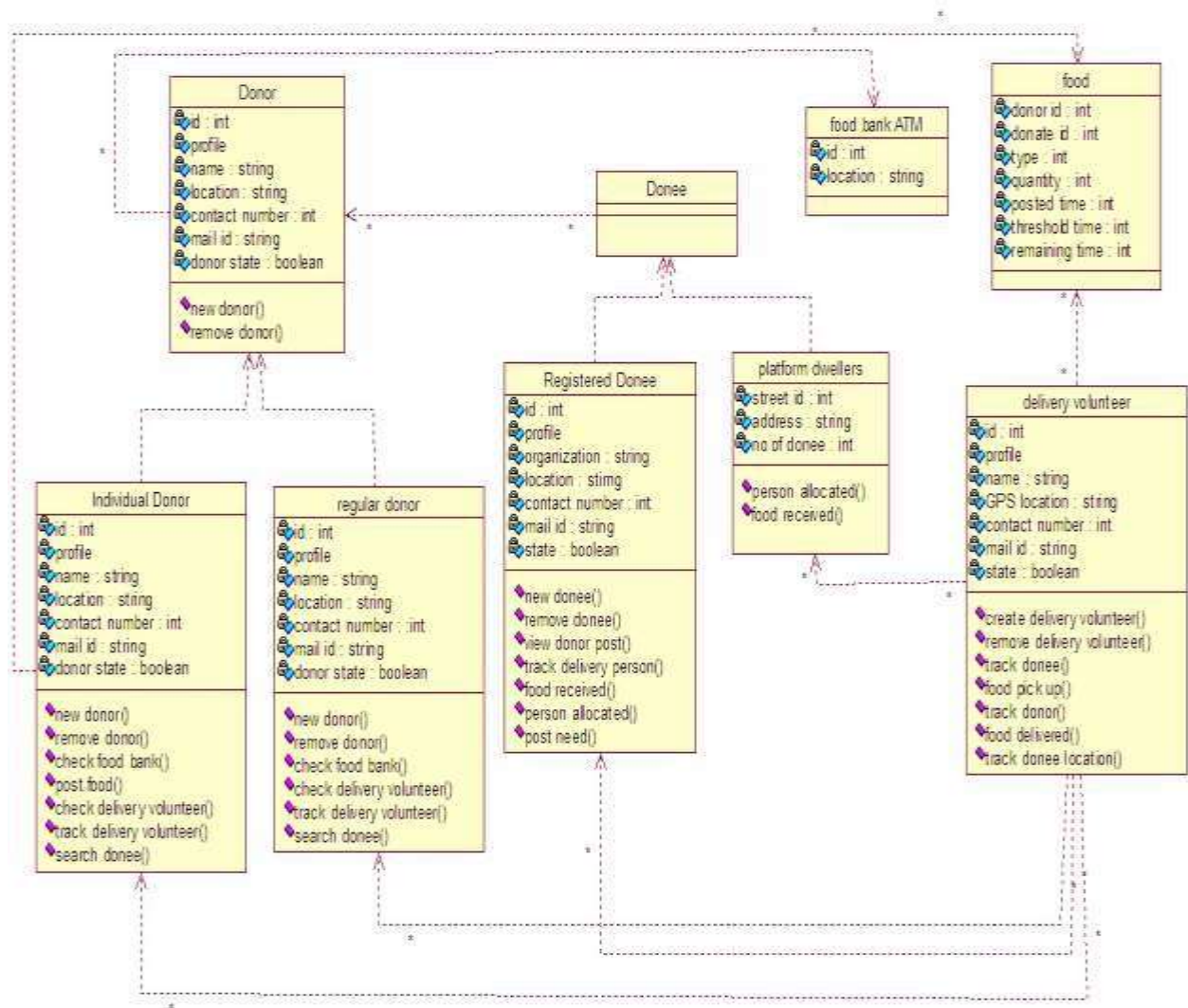
4.3. Activity Diagram



4.4. Sequence Diagram



4.5. Class Diagram



DESIGN

Design

5.1 Input Design

Input design is part of overall system designs that requires special attention to make the data entered free from errors. The forms are designed using the control available in Android studio IDE.

It is the process of converting the user originated input to a computer-based format. This project first will have several forms like donating form, updating form, signup form, login form, food requesting form, etc., in which all inputs entered gets collected.

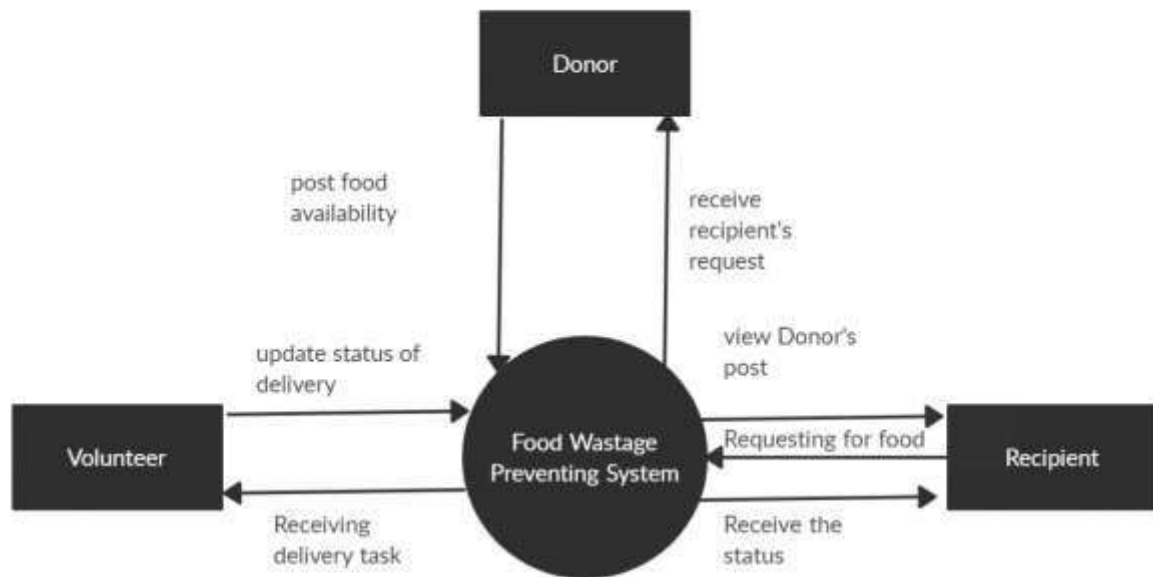
5.2 Output Design

As for this application “Food Wastage Preventing System” the main scope is to show the posts to the recipients. Output is the main reason for developing this application and the basis of which the usefulness of the application evaluated.

The output is designed in such a way that it is attractive, convenient, and informative. The forms are designed, which make the output more pleasing.

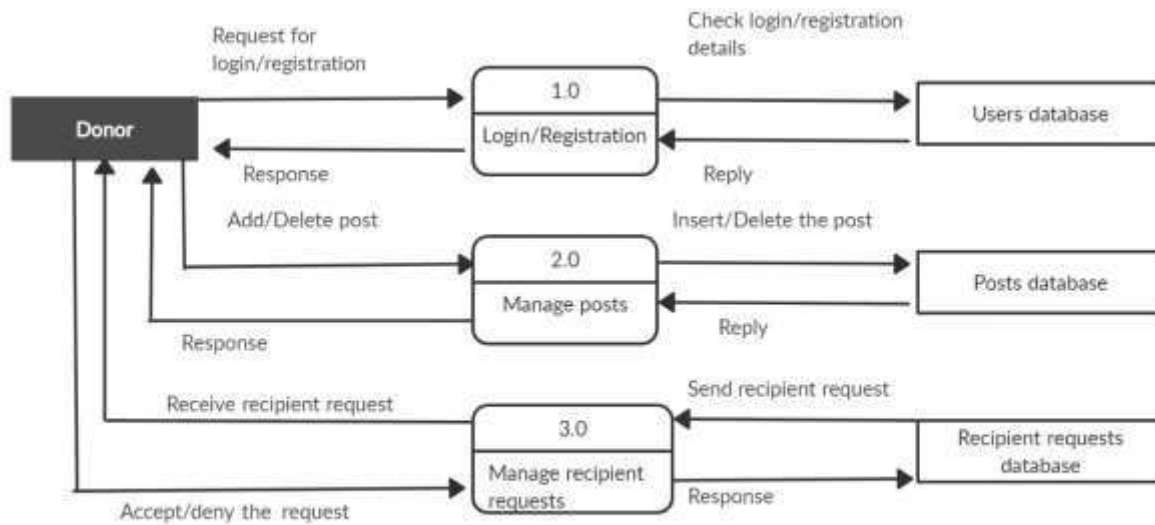
5.3. Data Flow Diagram

5.3.1. Data Flow Diagram Level-0



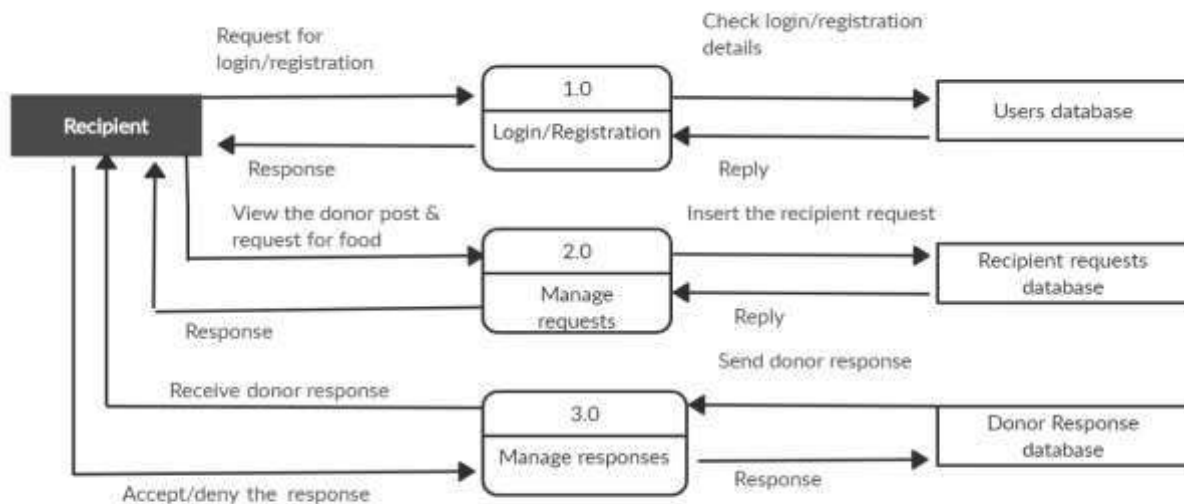
DFD Diagram Level-0

5.3.2. Donor Side Data Flow Diagram Level-1



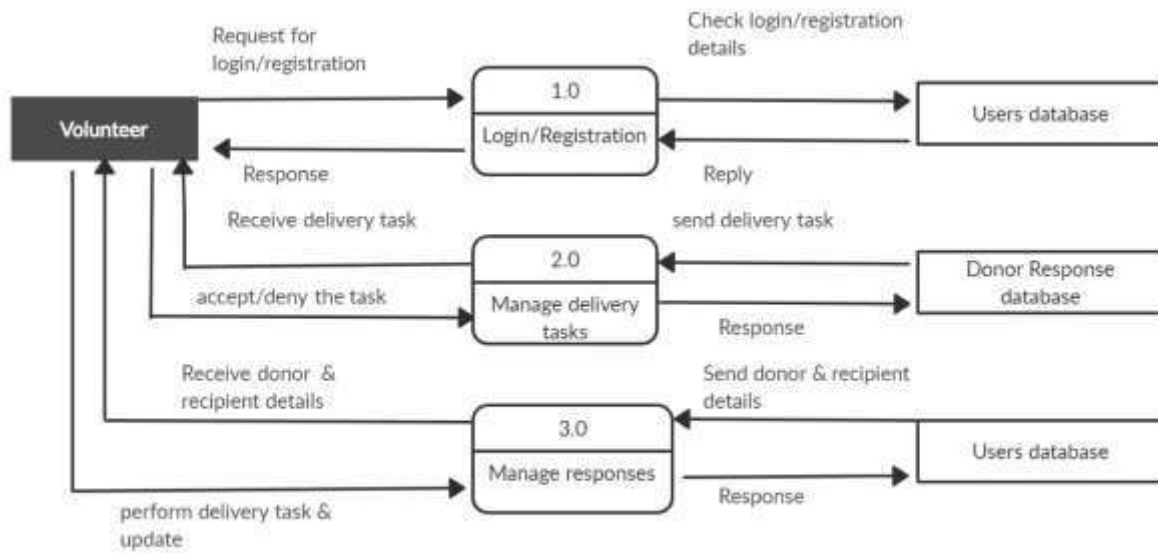
Donor side DFD-Level 1

5.3.3. Recipient Side Data Flow Diagram Level-1



Recipient side DFD-Level 1

5.3.4. Volunteer Side Data Flow Diagram Level-1



Volunteer side DFD-Level 1

TESTING

6. Testing

Test plan is necessary for any project to plan the testing phase and decide the scope of the project. Test plan involves collecting design specifications about the project, wiring test cases, executing them manually or automatically using automated testing tools. Testing any application is highly important. Test plan is a method of documenting the test cases, specification plans and other basic level details about how the application works.

Test Activities for this project includes various testing like:

6.1. Unit Testing:

- *I have tested all the modules of the application individually by running as a test program*

6.2. Black Box testing:

- *In this project, sample test cases are written and manual testing is done to check the functionality of the application.*

6.3. White Box testing:

- *Once the application meets the user requirements and functionalities according to the test cases, its internal logic is completely tested to ensure that the application does not have any logical errors or issues.*

6.4. Integration testing:

- *After testing the modules individually, tested them by integrating all the sub modules, modules into one application.*

6.5. System Testing:

- *It refers to checking whether the system in which the application is built meets the necessary requirements like software support. For example: In this project, I have checked whether the device in which the application developed is compatible with the software (Android Studio)*

6.6. End to End Testing:

- *Tested the complete environment of application by connecting the device with different machines, installing as an APK file, with the database and in local network.*

6.7. Usability Testing:

- *Finally, usability testing is performed by testing the application's flow, UI design and how flexible and easy the application is easy to use.*

6.8. Validation Testing

- *It is to check whether the application actually meets the user needs.*

IMPLEMENTATION

7.Implementation

7.1. Modules and its Description

Login & Registration Module

This Phase includes login & registration of Donors, Recipients & Volunteer Delivery People. Everyone has separate account and can see the updates, notification and all information regarding the process through it.

Donors Module

In this module each Donor have a separate account and through that whenever they have excess food, they can login and enter the details of the excess food and post it and they have the flexibility to remove the post any time.

Recipients Module

In this phase Registered Recipient could view the availability of food, and can request for the food from the donor. They can track the volunteer allocated for him/her to deliver the food. And they can also post their need in the Application.

Delivery Volunteers Module In this Module after the request of the recipient accepted by the donor, the details of the donor will be sent to Deliver person. After collecting the food from the Donor, the details of the recipient will be sent to the Volunteer, so that he/she could deliver it to the recipient

7.2 Problems Faced

- ***Finding the suitable maps API for this application and integrating the maps API with the application is one of the major problems.***
- ***Determining the suitable database for this application was also difficult.***
- ***Finding the current location of each users and store it as a location data in firebase database was one of the hardest milestones we faced.***

7.3. Lesson Learnt

- ***Android app development using android studio.***
- ***Cloud based database firebase Management and integration with respect to Android Studio.***
- ***JAVA programming languages basic.***
- ***Usage of XML for layout creation and definition.***
- ***Android app execution and deployment.***
- ***After gone through so many background studies we figured out Mapbox API would be better choice and we successfully integrated it with our application.***
- ***For finding the current location of all the users we use the Geocoding API, Places API, and Directions API provided by the Mapbox.***

- *Comparing to all the databases we prefer to use the Firebase database because of its simplicity, storage capability, efficiency in data handling, data retrieving and storing methodology.*
- *Since our application is android based and requires internet connection for executing it, we conclude that it is a wise choice to use the firebase database. It also reduces risk of losing the data as well eliminate the need for storage space required for storing all the data.*

FUTURE ENHANCEMENTS

8.1. Future Enhancements

- *Since it is an android based application a non-android and non-IOS user cannot access this application. For such type of users, a portal to receive the phone calls from them and convert their speech into text for further process can be included for improvising the scope of this application.*
- *Moreover, implementing this application in iOS platform as an iPhone App is one of the important considerations as we have many iPhone users.*

CONCLUSION

8.2. Conclusion

The food wastage preventing system is an android based application for primarily providing portal to the donor to post the food availability and allowing the Recipients to request the food to the donor who fulfil their requirements and making the transporting and delivery process much easier by connecting the volunteer delivery person through an another interface.

The goals that are achieved by this application are:

- *Simplification of the operation*
- *User friendly*
- *Instant Access*
- *Less processing time for getting information*
- *Improved productivity*
- *Optimum utilization of resources*
- *Efficient management of records*

REFERENCES

9. References

[1] Mark L. Murphy, “Android Programming Tutorials, Jan 2011, Version 3.1.

[2] <https://codinginflow.com/>.

[3] <https://www.youtube.com/smallacademy>

[4] <https://w3points.com/android-studio/>

[5] https://www.tutorialspoint.com/android/android_studio.htm

APPENDIX

10.1 Glossary

10.1.1. Login Page

1:47 PM0.02N/s 47 °C40 Volts +18%

Login

Email

password

LOGIN

Forget Password?

Don't have an account? SignUp

10.1.2. Signup page

1:48 PM0.02N/s 47 °C40 Volts +18%

☒ I am a recipient

☐ I am a donor

☐ I am a volunteer

UserName

password

confirm password

Organization Name

email id

Contact Number

Location

☐ I agree to the terms and conditions

SIGNUP

10.1.3 . Donation Form

Donation Form

Donor Name

donor4

Location

d4 street

Quantity of Food

1

Type of the Food

COO...

Description of the food

Posting Date and Time

Threshold Date and Time

Contact Number

8778341050

Images of the Food

Donation Form

Contact Number

8778341050

Images of the Food

CHOOSE

CAMERA

UPLOAD

Packing Details

☒ will deliver

☐ need pickup

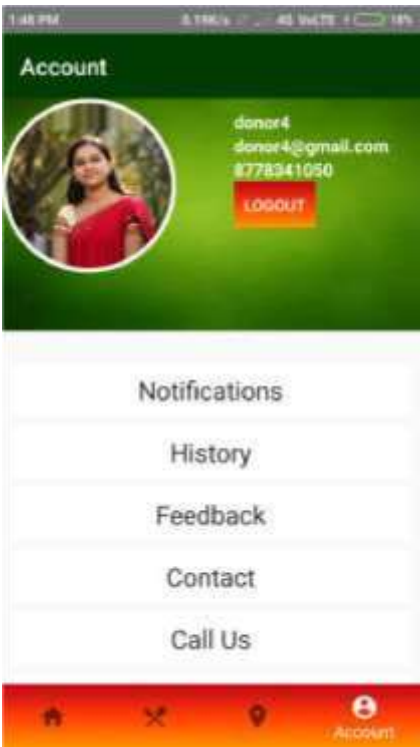
Pickup Details

☒ packed food

☐ unpacked food

POST

10.1.4 . User Account Page



10.1.5. User Account Update Option



10.1.6. History Page



10.1.7. Recipient requests



10.1.8 . Donor post Page



10.1.9. Donor post details



10.1.7.

10.1.10. Recipient Request Page



10.1.11. volunteer locating



10.2. Coding

```
<com.mapbox.mapboxsdk.maps.MapView
    android:id="@+id/mapView"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    mapbox:mapbox_cameraTargetLat="38.9098"
    mapbox:mapbox_cameraTargetLng="-77.0295"
    mapbox:mapbox_cameraZoom="12" />
<Button
    android:id="@+id/startButton"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="16dp"
    android:layout_marginEnd="16dp"
    android:background="@color/graylight"
    android:enabled="false"
    android:text="Start navigation"
    android:textColor="#FFF"
    mapbox:layout_constraintStart_toStartOf="parent"
    mapbox:layout_constraintTop_toTopOf="parent" />
<Button
    android:id="@+id/findButton"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:layout_marginStart="16dp"
    android:layout_marginLeft="16dp"
    android:layout_marginTop="69dp"
    android:layout_marginEnd="16dp"
    android:background="@color/graylight"
    android:enabled="false"
    android:text="find driver"
    android:textColor="#FFF"
    mapbox:layout_constraintStart_toStartOf="parent"
    mapbox:layout_constraintTop_toTopOf="parent" />

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