**An UDP Report**

**on**

**“Love Food Hate Waste”**

**By**

**Priyanka Maity (200339616157)**

**Chetna Padhiyar (200339616167)**

**Prof. Shakil Saiyad**

**PIET-DS, Limda**

**Submitted To**



**Computer Engineering Department**

**Parul Institute of Engineering and Technology-Diploma Studies**

**(Parul University)**

**Limda,Vaghodia,Vadodara**

# CERTIFICATE

This is to certify that the Love Food Hate Waste report submitted by Priyanka Maity (200339616157) and Chetna Padhiyar (200339616167) of Computer Engineering of PIET-DS is the record of work carried out by them under our supervision and guidance. The work has reached a level required for being accepted for examination.

**GUIDE** **HOD** **PRINCIPLE**

**Prof.** **Shakil Saiyad** **Prof. Hetal Bhaidasana**  **Dr.** **Ruchi Srivastava**

PIET-DS PIET-DS PIET-DS

Parul University Parul University Parul University

# ACKNOWLEDGEMENT

I would like to express my profound gratitude to **Prof. Hetal Bhaidasana (HOD)**, of computer department, and **Dr. Ruchi Srivastava** of PIET-DS for their contributions to the completion of my project titled Love Food Hate Waste. I would like to express my special thanks to our mentor for his time and efforts he/she provided throughout the year. Your useful advice and suggestions were really helpful to me during the project’s completion. In this aspect, I am eternally grateful to you.

# ABSTRACT

We are going to develop a project named Love Food Hate Waste. The project we are go for the people who doesn’t get food. That’s why we are developing our project name Love Food Hate Waste. By which people can donate extra food to the people who needed it by the help of NGOs. Like hotels extra food, weddings or functions extra food. One most advantage of our project is if any extra food is there but we can’t eat because it was spoil then we can give it for making biogas.

# TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| NO. | CHAPTER NAME | PAGE NO. |
| 1 | INTRODUCTION | 1 |

* 1. Project Summary
  2. Existing System
  3. Proposed System
  4. Scope

1. SYSTEM REQUIREMENTS 3
   1. Feasibility Study
   2. Requirements of new system
   3. Information of tools
2. DIAGRAMS 8
   1. E-R Diagram
   2. Use case Diagram
   3. Activity Diagram
   4. System Flow Diagram
   5. Data Dictionary Diagram
3. IMPLEMENTATION 24
4. CONCLUSION 28
5. FUCTURE SCOPE 29
6. BIBLIOGRAPHY 30

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| FIGURE NO. | TITLE | PAGE NO. |
| 1 | E-R Diagram | 10 |
| 2 | Use case Diagram | 12 |
| 3 | Activity Diagram | 14 |
| 4 | System Flow Diagram | 16 |
| 5 | Class Diagram | 19 |
| 6 | Data Dictionary Diagram | 20 |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| TABLE  NO. | TITLE | PAGE NO. |
| 1 | Login | 20 |
| 2 | User Table | 20 |
| 3 | Registration | 21 |
| 4 | Doner Table | 22 |
| 5 | Accepter Table | 22 |

# LIST OF IMPLEMENTATION STEPS

|  |  |  |
| --- | --- | --- |
| STEP NO. | TITLE | PAGE NO. |
| 1 | Login Page | 24 |
| 2 | Registration Page | 24 |
| 3 | Home Page | 25 |
| 4  5  6  7 | Search Food  Doner Page  Usable Food  Unusable Food | 25  26  27  27 |

**CHAPTER 1: INTRODUCTION**

## PROJECT SUMMARY

We will be creating a Love Food Hate Waste by which anyone can donate food by the help of NGOs to the people who doesn’t get food. The project we are go for the people who doesn’t get food. That’s why we are developing our project name Love Food Hate Waste. By which people can donate extra food to the people who needed it by the help of NGOs. Like hotels extra food, weddings or functions extra food.

One most advantage of our project is if any extra food is there but we can’t eat because it was spoil then we can give it for making biogas. Food is hailed as nectar, and wastage of food is considered a sin. Food wastage is a huge problem arising in today's world. It has become a serious issue in our society in the last years that affects “poor and rich countries” equally.

Do you know that one-third of all food globally goes to waste? That’s enough to feed 3 billion people! If food waste were a country it would be the third-largest greenhouse gas emitter on the planet! As per one data, the average person in India wastes 137 grams of food every single day. That’s 0.96 kg per week or 50 kg per year. In India, 40% of the food is wasted which is equivalent to Rs 92,000 crores a year.

## EXISYING SYSTEM

At present there is no such existing system present in our country. So this project is a new and unique idea which have a potiental to change how we can handle food waste system. This unique idea can help solve many problems that are faced in todays system.

## PROPOSED SYSTEM

We can provide the food that is left in hotels/restaurants or like wedding food to those who need it, through the application.

The food which was wasted we can used it like biogas and also used it like a fertilizer in farm. We can connect Waste management companies, Agricultural sector, Sewage treatment plants, Food industry companies with biogas plant.

## ADVANTAGES

* We can provide extra food to the people who need it.
* Keeps the environment clean and fresh.
* Creates employment.
* Waste management will help you to earn money.
* We can use waste food to produce biogas.
* We can make organic fertilizer from waste food.
* We can register for giving their extra food through NGOs to the people who needed it.

## SCOPE

## Registration

## User can register for giving their extra food through NGOs to the people who needed it.

## Doner

## Users can take a note and give the extra food to the NGOs like hotels, weddings, functions, home (extra food).

## Accepter

## The people who doesn’t have money to buy food then we can help them by giving our extra food (like hotels, weddings, functions, home (extra food)).

# CHAPTER 2: SYSTEM REQUIREMENT SYTUDY

## FEASIBILITY STUDY

* It is one of the important stages of system development stages.
* When a new system is proposed the main pointy taken into consideration is weather the system is feasible for not.
* If the proposed system does not pass the feasibility test then the further development of the project must be immediately stopped.
* If the system fails the feasibility test then the work must be stopped otherwise all further work done is wasted.
* This study consist of 3 steps which are listed below

Operational feasibility

Technical feasibility

Economic feasibility

**OPERATIONAL FEASIBILITY**

* Operational feasibility stated that the system should work under conditions for it to be operationally feasibile
* The operations that checks weather the system is feasible or not are as follows:
* An unauthorized person cannot make any changes in stored information.
* The system will operate only when servers as well as networka are up.

**TECHNICAL FEASIBILITY**

* For the system it should be technical feasible.
* There should be some computer attached to the server, these nodes.
* While a network should have been established between the server and the node.
* The minimum memory requirement is 2 GB.

**ECONOMIC FEASIBILITY**

* + - * It checks weather the system is under budget or not.
      * If the amounts spend behind developing is more than the profit made by then the system is not considered to be economic feasible hence the organization must see the profit made by system is compatible by the amount spend to develop it.
      * To check if the proposed system is feasible or not the
      * Organization must consider the following steps.
      * There should be a network established in organization and licensed version of Android Studio. If their requirements are available then the system is economically feasibile.

We have performed this feasibility test and are happy to state that our project passed all the 3 feasibility test that is operational feasibility, technical feasibility and the economic feasibility.

## TOOLS AND TECHNOLOGY

**SOFTWARE REQUIREMENTS**

1. **Frontend**
2. **Backend**

# Frontend

**Cascading Style Sheets** (**CSS**)

**CSS** is a [stylesheet](https://developer.mozilla.org/en-US/docs/Web/API/StyleSheet) language used to describe the presentation of a document written in [HTML](https://developer.mozilla.org/en-US/docs/Web/HTML) or [XML](https://developer.mozilla.org/en-US/docs/Web/XML/XML_introduction) (including XML dialects such as [SVG](https://developer.mozilla.org/en-US/docs/Web/SVG), [MathML](https://developer.mozilla.org/en-US/docs/Web/MathML) or [XHTML](https://developer.mozilla.org/en-US/docs/Glossary/XHTML)). CSS describes how elements should be rendered on screen, on paper, in speech, or on other media.

CSS is among the core languages of the **open web** and is standardized across Web browsers according to [W3C specifications](https://www.w3.org/Style/CSS/#specs). Previously, the development of various parts of CSS specification was done synchronously, which allowed the versioning of the latest recommendations. You might have heard about CSS1, CSS2.1, or even CSS3. There will never be a CSS3 or a CSS4; rather, everything is now CSS without a version number.

After CSS 2.1, the scope of the specification increased significantly and the progress on different CSS modules started to differ so much, that it became more effective to [develop and release recommendations](https://www.w3.org/Style/CSS/current-work) [separately per module](https://www.w3.org/Style/CSS/current-work). Instead of versioning the CSS specification, W3C now periodically takes a snapshot of [the latest stable state of the CSS](https://www.w3.org/TR/css/) [specification](https://www.w3.org/TR/css/) and individual modules progress. CSS modules now have version numbers, or levels, such as [CSS Color Module Level 5](https://drafts.csswg.org/css-color-5/).

**JavaScript**

[JavaScript](https://www.lighthouselabs.ca/en/intro-front-end-developer-course-javascript) is used by programmers across the world to create dynamic andinteractive web content like applications and browsers. JavaScript is so popular that it's the most used programming language in the world, used as a client-side programming language by [97.0% of all websites](https://w3techs.com/technologies/details/cp-javascript). Client-side languages are those whose action takes place on the user's computer, rather than on the server.

JavaScript is versatile enough to be used for a variety of different applications, like software, hardware controls, and servers. JavaScript is most known for being a web-based language, because it's native to the web browser. The web browser can naturally understand the language, like how a native English speaker can naturally understand English.

**HTML**

* HTML stands for Hyper Text Markup Language.
* HTML is the standard markup language for creating Web pages.
* HTML describes the structure of a Web page.
* HTML consists of a series of elements.
* HTML elements tell the browser how to display the content.
* HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.
* The <html> element is the root element of an HTML page.
* The <head> element contains meta information about the HTML page.
* The <title> element specifies a title for the HTML page (which is shown in the browser's title bar or in the page's tab).
* The <body> element defines the document's body, and is a container for all the visible contents, such as headings, paragraphs, images, hyperlinks, tables, lists, etc.
* The <h1> element defines a large heading.
* The <p> element defines a paragraph.

**PHP**

PHP is a general-purpose scripting language geared toward web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994.

* PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
* PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
* You add, delete, modify elements within your database through PHP.
* Access cookies variables and set cookies.
* Using PHP, you can restrict users to access some pages of your website.
* It can encrypt data.

# Backend

**Python**

* Python is a high-level, general-purpose programming language.
* Its design philosophy emphasizes code readability with the use of significant indentation.
* Python is dynamically-typed and garbage-collected.
* It supports multiple programming paradigms, including structured, object-oriented and functional programming.

**C++**

* C++ is a cross-platform language that can be used to create high-performance applications.
* C++ was developed by Bjarne Stroustrup, as an extension to the C language .
* C++ gives programmers a high level of control over system resources and memory.
* The language was updated 4 major times in 2011, 2014, 2017, and 2020 to C++11, C++14, C++17, C++20.

## Java

* Java is a widely used object-oriented programming language andsoftwar**e** platform that runs on billions of devices, including notebook computers, mobile devices, gaming consoles, medical devices and many others.
* The rules and syntax of Java are based on the C and C++ languages.

**Kotlin**

* Kotlin is an open-source, statically-typed programming language that supports both object-oriented and functional programming.
* Kotlin provides similar syntax and concepts from other languages, including C#, Java, and Scala, among many others.
* Kotlin does not aim to be unique—instead, it draws inspiration from decades of language development

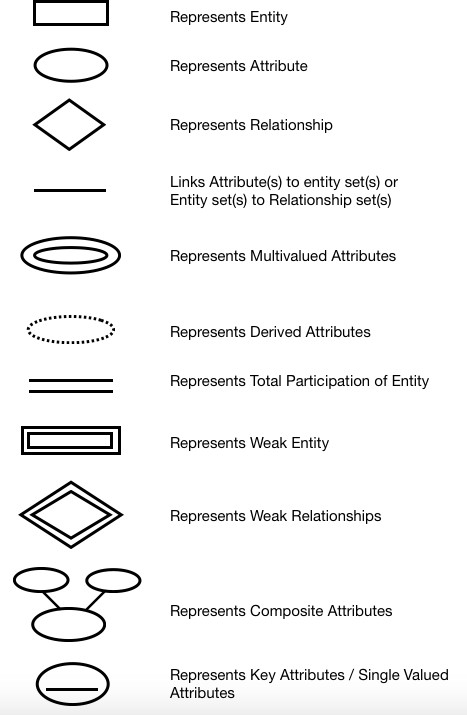
# CHAPTER 3: DIAGRAMS

## E-R DIAGRAM

An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research.

Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes.

**SYMBOLS OF E-R DIAGRAM**



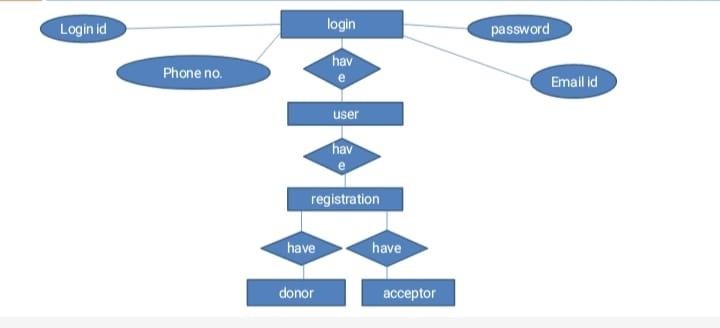
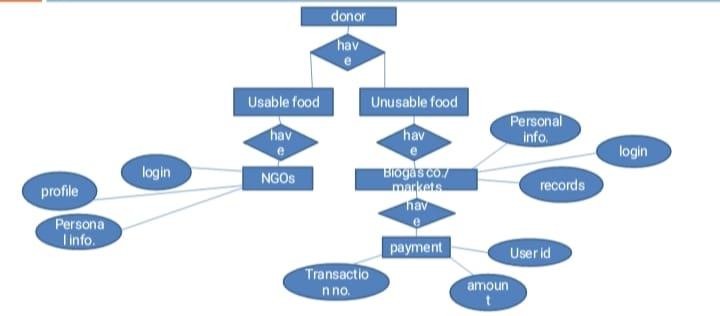


Fig 3.1.1: Login



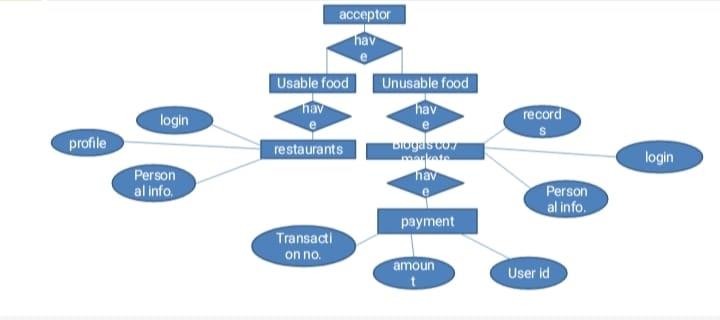
Fig 3.1.2: Donor

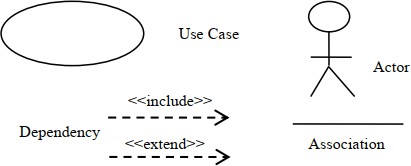
Fig 3.1.3: Acceptor

# USE CASE DIAGRAM

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well.

The use cases are represented by either circles or ellipses.

**SYMBOLS OF USE CASE DIAGRAM**



Registration/login

Read item

donor

View donation request

Place food to donate

Request for items

receiver

Track receiver location

Accept food request

Track donor location and contact number

admin

Notify user

Update database

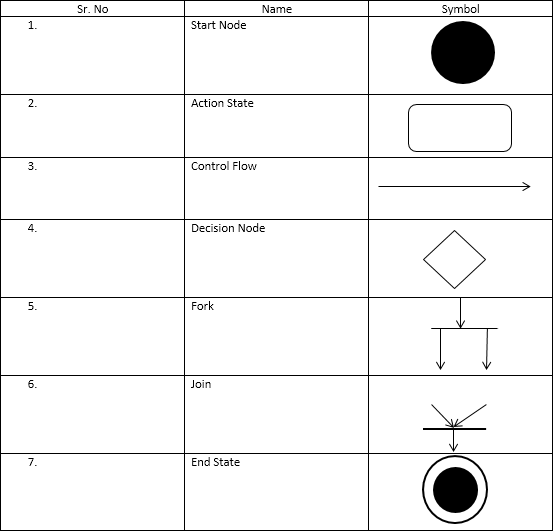
Fig 3.2: Use Case Diagram

# ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency.

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram.

**SYMBOLS OF USE ACTIVITY DIAGRAM**



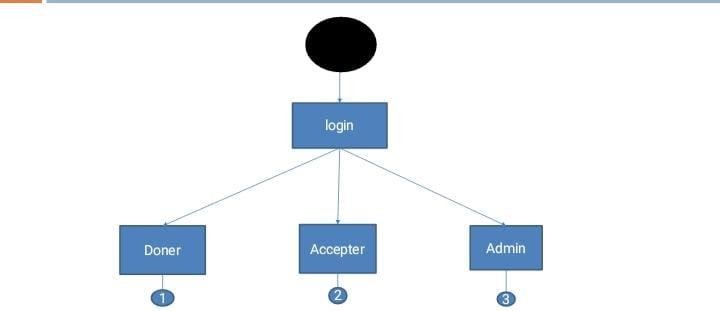


Fig 3.3.1: Login

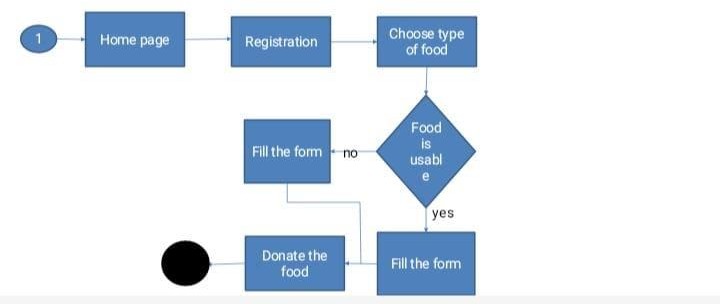


Fig 3.3.2: Donate Food

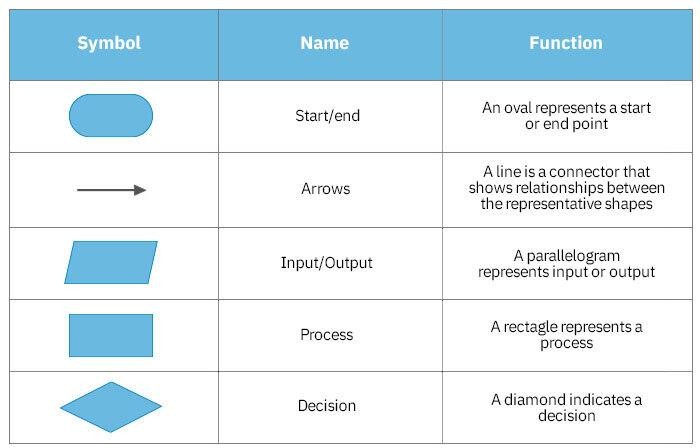


Fig 3.3.3: Accept Food

# SYSTEM FLOW DIAGRAM

System flowcharts are a way of displaying how data flows in a system and how decisions are made to control events. To illustrate this, symbols are used. They are connected together to show what happens to data and where it goes.

**SYMBOLS OF SYSTEM FLOW DIAGRAM**

**

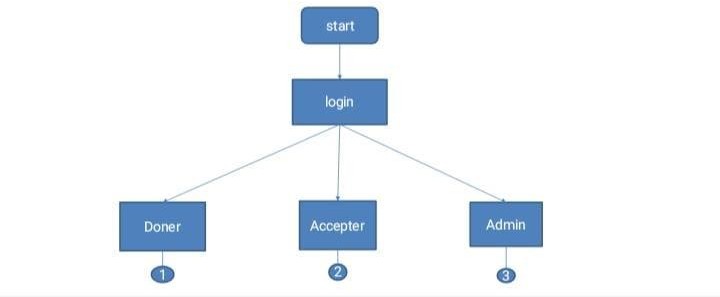


Fig 3.4.1: Login

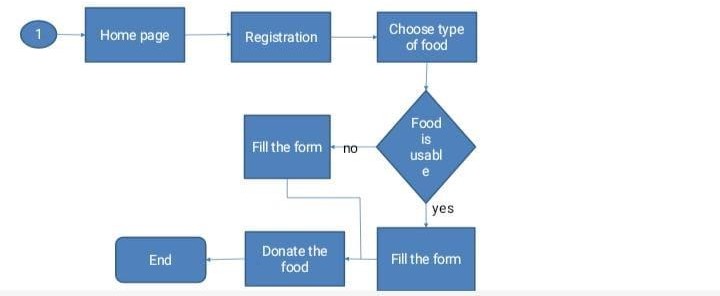


Fig 3.4.2: Donate Food



Fig 3.4.3: Accept Food

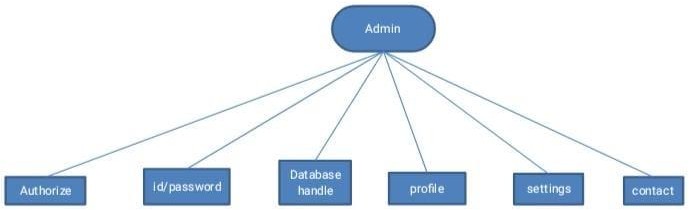
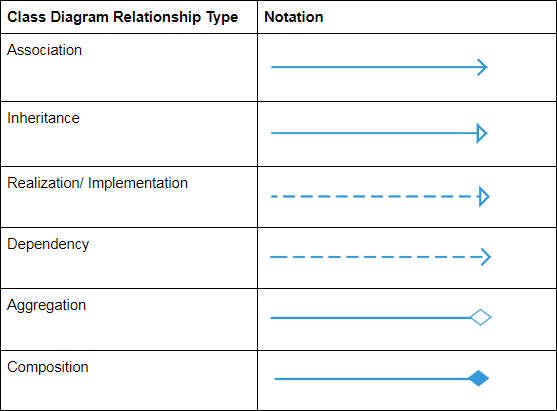


Fig 3.4.4: Admin

# CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

**SYMBOLS OF CLASS DIAGRAM**



|  |
| --- |
| Login |
| Phone no:int(10) |
| Email: varchar(30) |
| Password:varchar(8) |

|  |  |  |  |
| --- | --- | --- | --- |
| Registration | | |  |
| User id: varchar | | |
| User name: varchar | | |
| Phone no: int(10) | | |
| Email id: varchar(30) | | |
| Password:varchar(8) | | |
| Address: varchar | | |
|  | |  | |
|  | User | | |
| User id: varchar | | |
| User name: | | |
| Phone no:int(10) | | |
| Email: varchar(30) | | |
| Password:varchar(8) | | |

|  |
| --- |
| Accepter |
| User id: varchar |
| Company name |
| Address: varchar |
| Phone no: int(10) |
| Email id:  varchar(30) |
| Taddermark: :  varchar |

|  |
| --- |
| Donor |
| User id: varchar |
| Hotel name: : varchar |
| Address: varchar |
| Phone no: int(10) |
| Email id: varchar(30) |
| Taddermark: : varchar |
| Type of food: : varchar |

Fig 3.5: Class Diagram

# DATA DICTIONARY

Data dictionary is an information which contains meta data. It is usually a part of the system catalog. Data dictionary contains description of schema i.e, overall logical structure of the database.

This can in involve information such as table names, owners, column, column name, data types, size and constraints.

**Table : Login**

|  |  |  |  |
| --- | --- | --- | --- |
| Login | | | |
| **Field name** | **Data type** | **Description** | **Constraints** |
| Phone no. | Int(10) | User phone no | Primary key |
| Email id | Varchar(30) | User email id | Not null |
| Password | Varchar(8) | User password | Not null |

**Table : User Table**

|  |  |  |  |
| --- | --- | --- | --- |
| User table | | | |
| Field name | Data type | Description | Constraints |
| User id | Int(8) | User id | Primary key |

|  |  |  |  |
| --- | --- | --- | --- |
| User name | Varchar | User full name | Not null |
| Phone no. | Int(10) | User phone no | Not null |
| Email id | Varchar(30) | User email id | Not null |
| Password | Varchar(8) | User password | Not null |

**Table : Registration**

|  |  |  |  |
| --- | --- | --- | --- |
| Registration | | | |
| Field name | Data type | Description | Constraints |
| User id | Int(8) | User id | Primary key |
| User name | Varchar | User full name | Not null |
| Phone no. | Int(10) | User phone no. | Not null |
| Email id | Varchar(30) | User email id | Not null |
| Password | Varchar(8) | User password | Not null |
| Address | Varchar | Address of user | Not null |

**Table : Doner Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Doner Table** | | | |
| **Field name** | **Data type** | **Description** | **Constraints** |
| User id | varchar | User id no | Primary key |
| Hotel name | Varchar | Hotel name | Not null |
| Address | Varchar | Address of hotel | Not null |
| Phone no. | Int | Phone no. of hotel | Not null |
| Email id | Varchar(30) | Email id of hotel | Not null |
| Trademark | Varchar | Mark of location | Not null |
| Type of food | Varchar | Usable or Unusable | Not null |

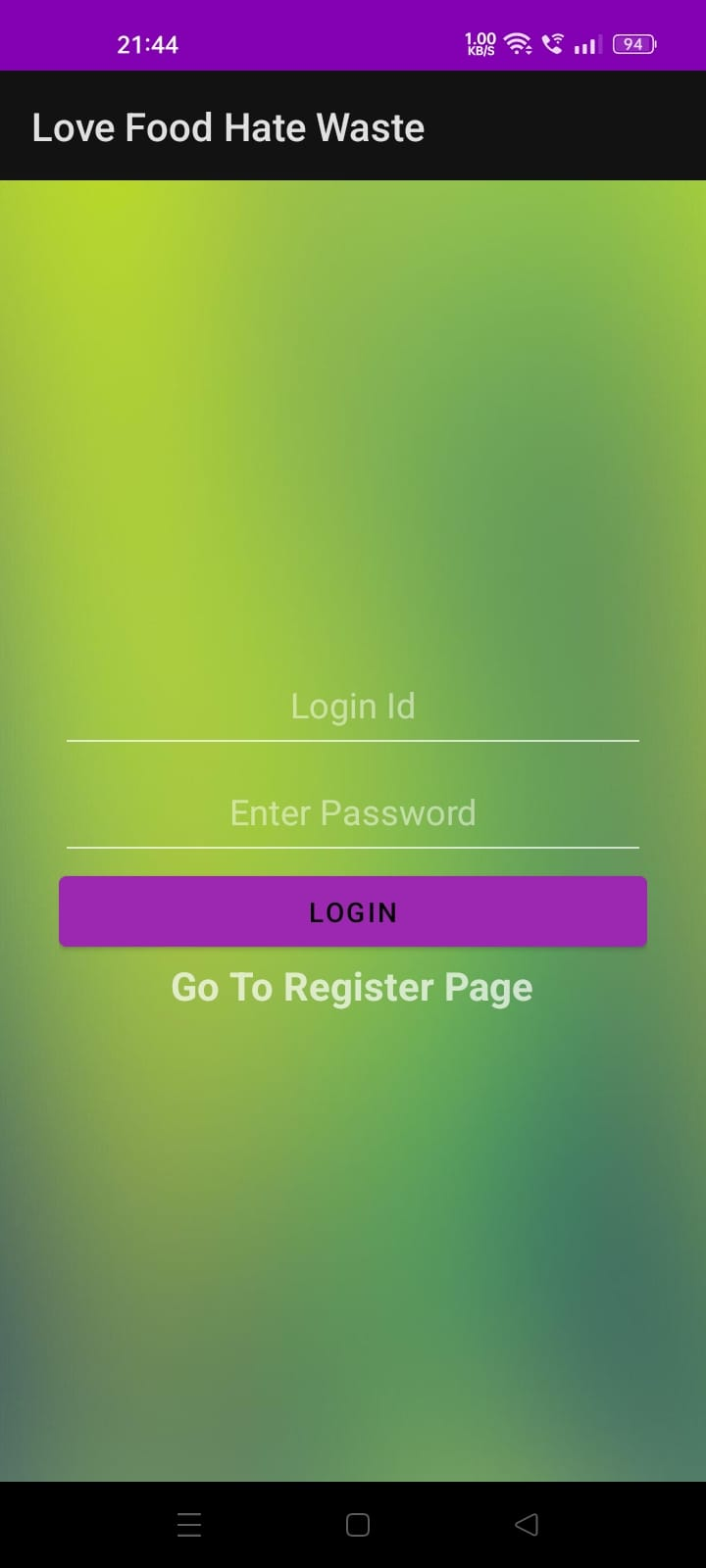
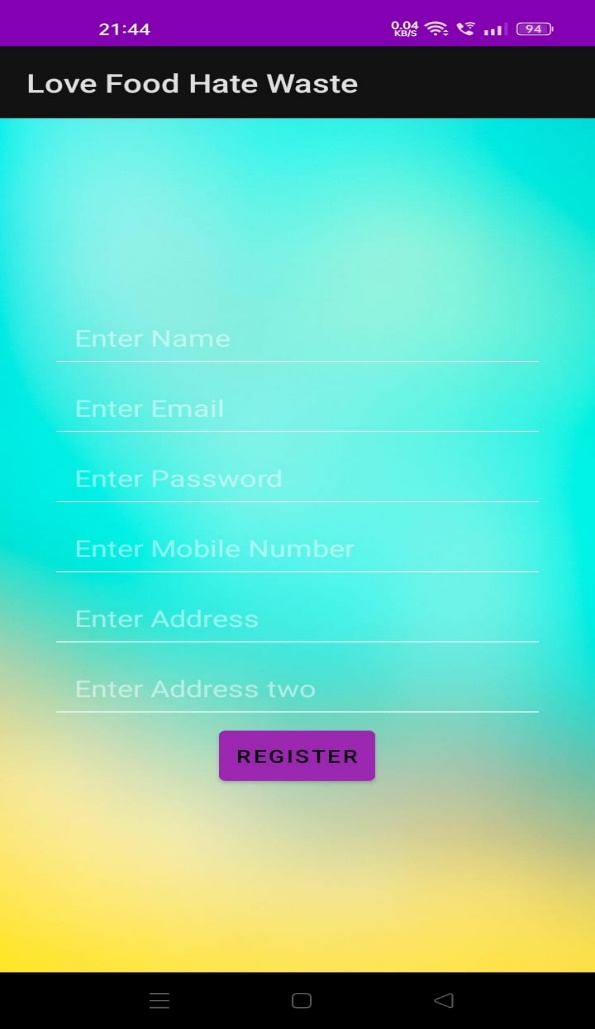
**Table : Accepter Table**

|  |  |  |  |
| --- | --- | --- | --- |
| **Accepter Table** | | | |
| **Field name** | **Data type** | **Description** | **Constraints** |
| User id | varchar | User id | Primary key |
| Company name | Varchar | Hotel name | Not null |
| Address | Varchar | Address of hotel | Not null |

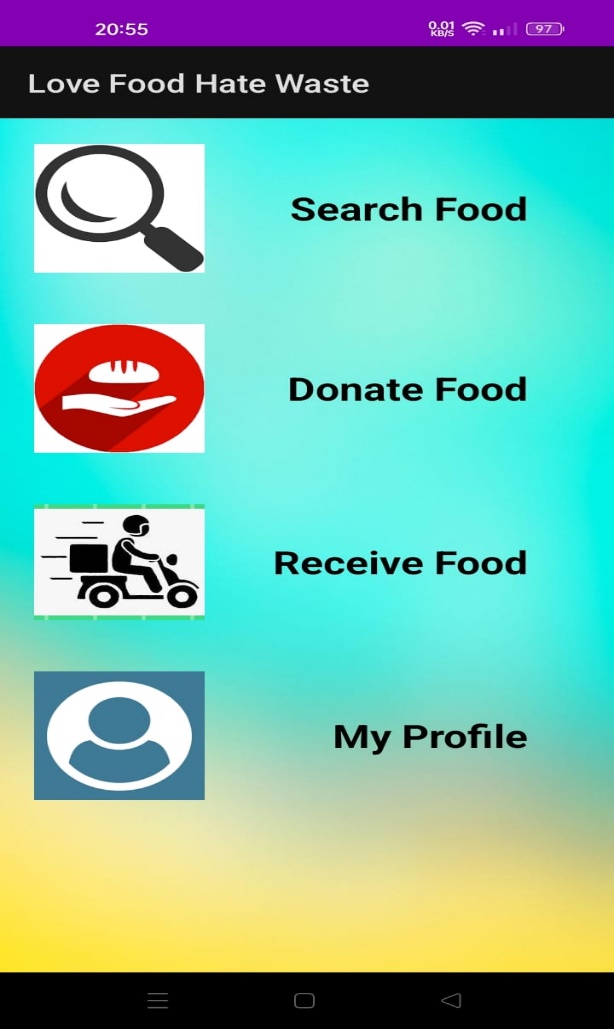
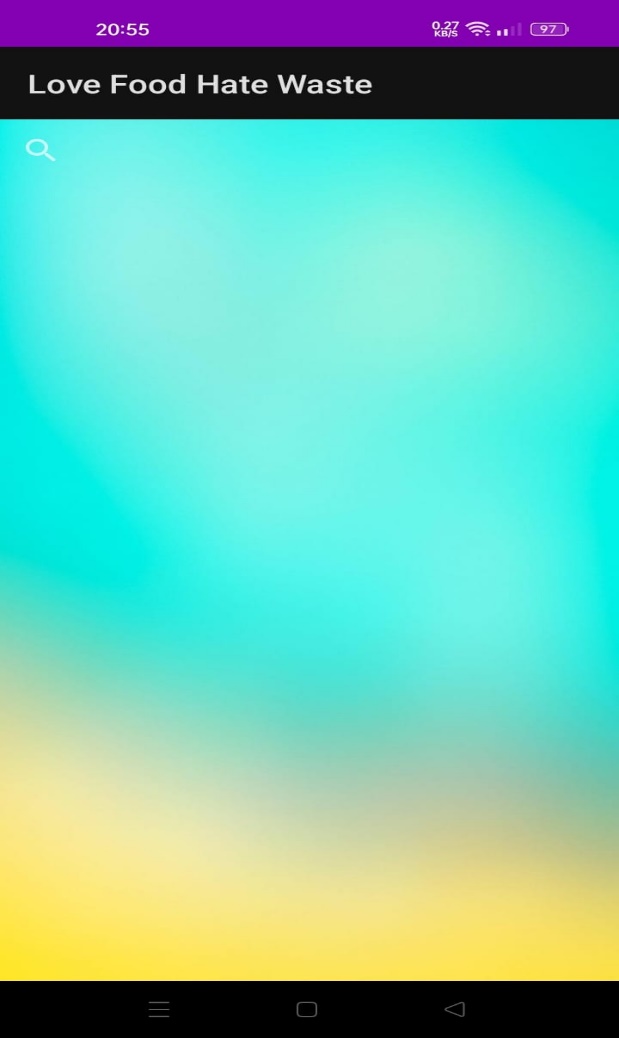
|  |  |  |  |
| --- | --- | --- | --- |
| Phone no. | Int | Phone no. of hotel | Not null |
| Email id | Varchar(30) | Email id of hotel | Not null |
| Trademark | Varchar | Mark of location | Not null |

# CHAPTER 4: IMPLEMENTATION

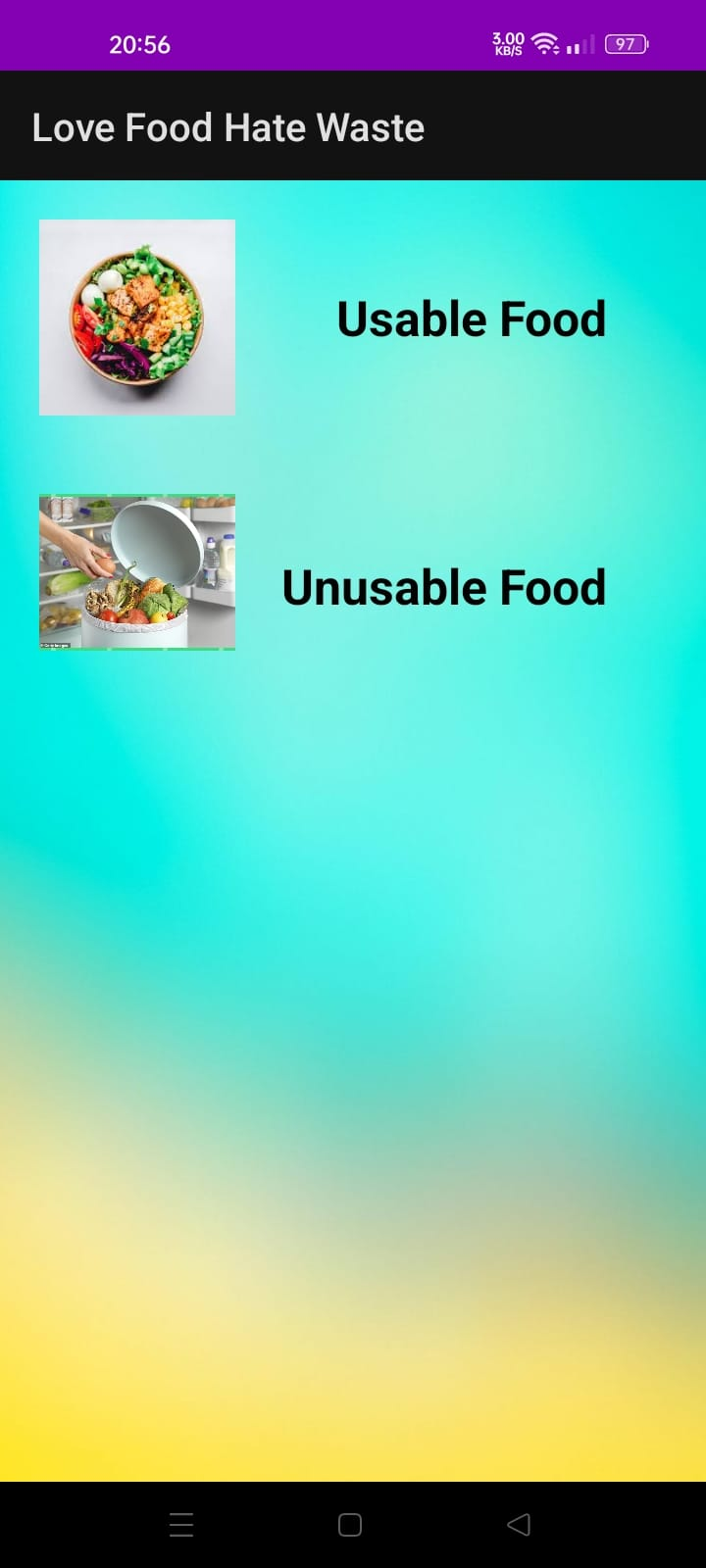
Login:- Register:-



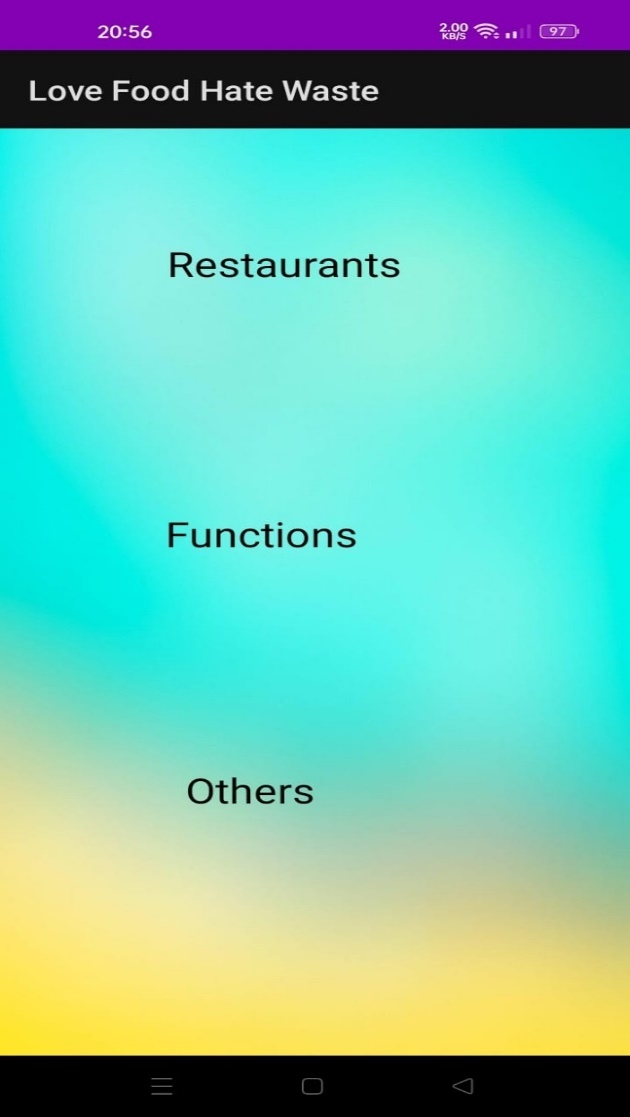
Home Page:- Search Food:-



Donate Food:-



Usable Food:- Unusable Food:-



# CHAPTER 5: CONCLUSION

We are going to develop a service called Love Food Hate Waste which will help to the people who doesn’t get food. By the help of this application we can easily connect the doner and accepter in one application. We can give the extra food of hotels, restaurants(extra food) to the people who doesn’t get food by the help of NGOs.

Also we can give the markets spoil food to the industries for making biogas plants.

# CHAPTER 6: FUTURE SCOPE

* + In 2022, the food waste management industry is expected to be worth $62.6 billion.
  + According to Future Market Insights (FMI), the total market value is estimated to reach US$ 116.4 bill by 2032, with a CAGR of 6.4 per cent for the years 2021-32.
  + Waste Management in India is basically all those activities, which are required to manage waste from its beginning to the final disposal.
  + Waste Management majorly includes things like the collection, transport, treatment, and the ultimate disposal of waste with a high level of monitoring and regulation.

# CHAPTER 7: BIBLIOGRAPGY

* + - <https://www.w3schools.com>
    - <https://www.lovelycoding.org/waste-food-management-system>
    - [https://www.avristech.com/food-waste-management-in-india](https://www.avristech.com/food-waste-management-in-india/)