# Food Waste Management System

**Mustajab Amin**

64340

**Muhammad Arqum**

64397

**Under the Supervision of**

**Mr. Talha Vidhani**

**2022-2023**

**College of Computing & Information Sciences**
**PAF-KIET Campus, Karachi Pakistan**

# College of Computing & Information Sciences

## NN Campus, Karachi, Pakistan

## CERTIFICATE

This project "**Food Waste Management System Web App**" presented by **Mustajab Amin (64340) & Muhammad Arqum (64397)** under the direction of their project advisor and approved by the project examination committee, has been presented to and accepted by the CoCIS, in partial fulfilment of the requirements for the bachelor's degree of Computer Science.

Mr. Talha Vidhani

(Project Advisor)

**Dr. Khalid Khan**

(Director CoCIS)

**Usman Khan**

(FYP Committee Head)

# ABSTRACT

The commonly known fact about hunger in the world is not shortage of food but rather access to the food. The amount of world's food thrown in garbage is one-third of the total food. Right now, world produces enough food to feed every person on the planet. In order to help, we have developed an web based application that will provide a platform to the restaurants to donate their leftovers to those in need. The main objective of this project based on web and mobile application is to make sure to deliver leftover food by restaurant to NGOs which will then provide it to those in need. This system consists of four modules that are admin, NGO's, restaurants and drivers. The majority of the population today uses smart phones with active internet connectivity, which is the basic requirement for this product to function properly. NGOs will send request to restaurants for food and if they will have leftover food, they will accept request and select available rider to pick up from restaurants. On the other hand, restaurants can also donate food by sending request to NGO and after acceptance of request they will donate food.

# ACKNOWLEDGEMENT

***In the name of Allah, the most Gracious and the Most Merciful.***

***Peace and blessing of Allah be upon Prophet Muhammad ﷺ***

First, praise of Allah, for giving us this opportunity, the strength and the patience to complete our FYP finally, after the challenges and difficulties. We would like to thank our supervisor ***Sir Talha Vidhani*** for his guidance, motivation and most his significant contribution in this project, expert ***Sir Saad Akbar, Mam Saboohi*** and ***Miss Yumna*** for giving us the opportunity to work on this project. We would also like to thanks our parents for financial and moral support and our friends who have helped and motivated us throughout. May Allah reward them all abundantly. Ameen

# DEDICATION

This report is dedicated to PAF-KIET University, our Teacher, our Supervisor, our Parents, our fellow colleagues and the hard-working students of PAF-KIET, with a hope that they will succeed in every aspect of their Academic Career and this project may help them in any aspect of their life.

# TABLE OF CONTENTS

# LIST OF FIGUERS

# CHAPTER 1

# 1. Introduction

## 1.1. Motivations

We usually see a lot of leftover food being dumped in trash by restaurants and hotels.so we had a idea how we can save this food from being dumped in trash. There are a lot of NGOs around us which ask our support to help the people in need.so we decided to build a system between restaurants and NGOs where NGOs can directly ask for food from restaurant's if they have some leftover. On the other hand, if restaurants have leftover food and they want to donate it they can as well without getting into trouble of finding ngos to donate food. In this manner food will not be wasted and NGOs can provide food to those who are in need of it.

## 1.2. Problem Statement

A drastic increase can be seen in food waste as per data given by food and agriculture organization ,1/3 food produce by human consumption is wasted globally, which account for almost 1.3 billion tons per year. On the other hand, as per WHO 20% of population face extreme food shortage. Hence there is need to come up with a solution that can reduce food waste and help people in need.

- Restaurants doesn't have any idea what to do with leftover food so they dump it in trash.
- NGOs have no idea who to ask for food or who has excess of it

## 1.3. Objectives and Contributions

**• Sends a notification to the restaurant.**

The user can create the order of food tap the button to send a notification to the restaurant and request for the food.

**• Reduces the amount of time asking for food**

The NGOs no longer needs to call and check with every restaurant for food. They don't need to waste their energy by calling different restaurants every day and every time. They just need to make a list of food and send a restaurants and wait for their response. In a same manner restaurant also don't need to ask every ngos to take their remaining food just send request and wait for their response.

**• restaurants can also donate**

On some occasion where restaurant cook abundant of food or there is a strike on that day or some reservations have been cancelled due to bad circumstances. They can donate food without waiting for NGOs request. And save food from dumping into trash and help those in need.

**• Safe and dependable.**

We have super admin so only he can register Ngos, Restaurants and Riders. So no fake restaurants and ngos can register themselves.in this way only authentic business will be involved in it and we will be assure that our food is reaching to right people

## 1.4. Project Scope

The scope of this project is to develop and implement a food waste management System. This project will overcome the problems associated with the excess/leftover food from restaurants. and distribute to the needy people through NGOs. NGOs will collect the leftover or excess food from above mentioned venues for the distribution to the needy people. And if restaurants have abundant of food they can also donate it to ngos and if ngos need that food they will accept their request and send their rider to pick up food from restaurant and donate it to people in need.

## 1.5. Organization of the Report

1. Introduction
2. Literature Review
3. Project Planning
4. Project Diagram
5. GUI of Web Application
6. Conclusion

# CHAPTER 2

## 2. Literature Review/Process Review

### 2.1. Introduction

Food waste is a critical issue with significant environmental, social, and economic implications. Web applications have emerged as promising tools for addressing food waste by enabling efficient management and waste reduction throughout the food supply chain. This literature review aims to explore the existing research on food waste management web applications, highlighting key features, benefits, challenges, and future directions.

### 2.2. Literature Review

A comprehensive search of academic databases, industry reports, and relevant websites was conducted using keywords such as "food waste," "web application," "technology," and "sustainability." The review focused on recent articles published within the last decade to ensure up-to-date information.

**Features of Food Waste Management Web Applications:**

Food waste management web applications encompass various features, including food inventory tracking, expiration date monitoring, donation coordination, meal planning, recipe suggestions for utilizing leftovers, user feedback mechanisms, and data analytics for waste monitoring and optimization (Gómez-Verdejo et al., 2021; Ganti et al., 2020).

**Benefits of Food Waste Management Web Applications:**

Web applications for food waste management offer several advantages, such as improved inventory management, reduced food waste, cost savings for businesses, enhanced food safety, increased donation opportunities, and heightened consumer engagement and awareness (Ganti et al., 2020; Pareigis et al., 2020).

**User Acceptance and Adoption:**

Factors influencing user acceptance of food waste management web applications include ease of use, perceived usefulness, trust, and social norms. User-centric design and intuitive interfaces are critical in promoting adoption and engagement (Yoon & Kwon, 2019; Pareigis et al., 2020).

**Integration with Supply Chain Partners:**

Successful food waste management web applications require collaboration and integration with supply chain partners, including suppliers, retailers, food service providers, and waste management organizations. Streamlining processes and ensuring seamless data sharing are essential for maximizing impact (Gómez-Verdejo et al., 2021; Heikkilä et al., 2021).

**Data Security and Privacy:**

Due to the collection and storage of sensitive data, ensuring data security and privacy is paramount. Researchers have addressed concerns related to data protection, privacy regulations compliance, and secure infrastructure development for data management (Heikkilä et al., 2021; Zuin & Rodrigues, 2020).

**Impact Assessment and Evaluation:**

Food waste management web applications' effectiveness can be evaluated through frameworks and indicators that assess their environmental, social, and economic impacts. These include waste reduction, resource conservation, greenhouse gas emissions reduction, cost savings, and social benefits (Gómez-Verdejo et al., 2021; Zuin & Rodrigues, 2020).

**Conclusion:**

Food waste management web applications have demonstrated potential in reducing food waste and promoting sustainable practices. While the features and benefits are promising, challenges such as user adoption, data security, and integration with supply chain partners remain. Future research should focus on usability studies, user-centered design approaches, comprehensive impact assessments, and cross-sector collaboration to enhance the effectiveness and scalability of these web applications.

In conclusion, food waste management web applications offer valuable solutions for addressing food waste. By leveraging technology and engaging users, these applications can transform food waste management practices and contribute to a more sustainable and efficient food system.

## 2.3. Functional and Non-Functional Requirements

### *2.3.1. Functional Requirements*

- Ngos and restaurant must register to use this application.
- Only driver's approved by admin will be able to sign in the application.

### *2.3.2. Non-Functional Requirements*

**1. Performance Requirements**

1.1. Application will perform with minimal lag.

1.2. Realtime database.

**2. Reliability**

2.1. Ensure Data consistency

**3. Security**

3.1. Password protection

## 2.4. Project Significance

The significances of this project are:

- Provide leftover food from restaurant to NGO's which can be provided to those in need.
- Reduce the wastage of food.
- Register NGO's and restaurant so we don't have to search for them explicitly.

## 2.5. Software Platform

The IDE which we are using to build mobile application is Android Studio and language used is javascipt on the other hand for web app we have used visual studio code and used codeigniter 3 framework using php language.

## 2.6. Services

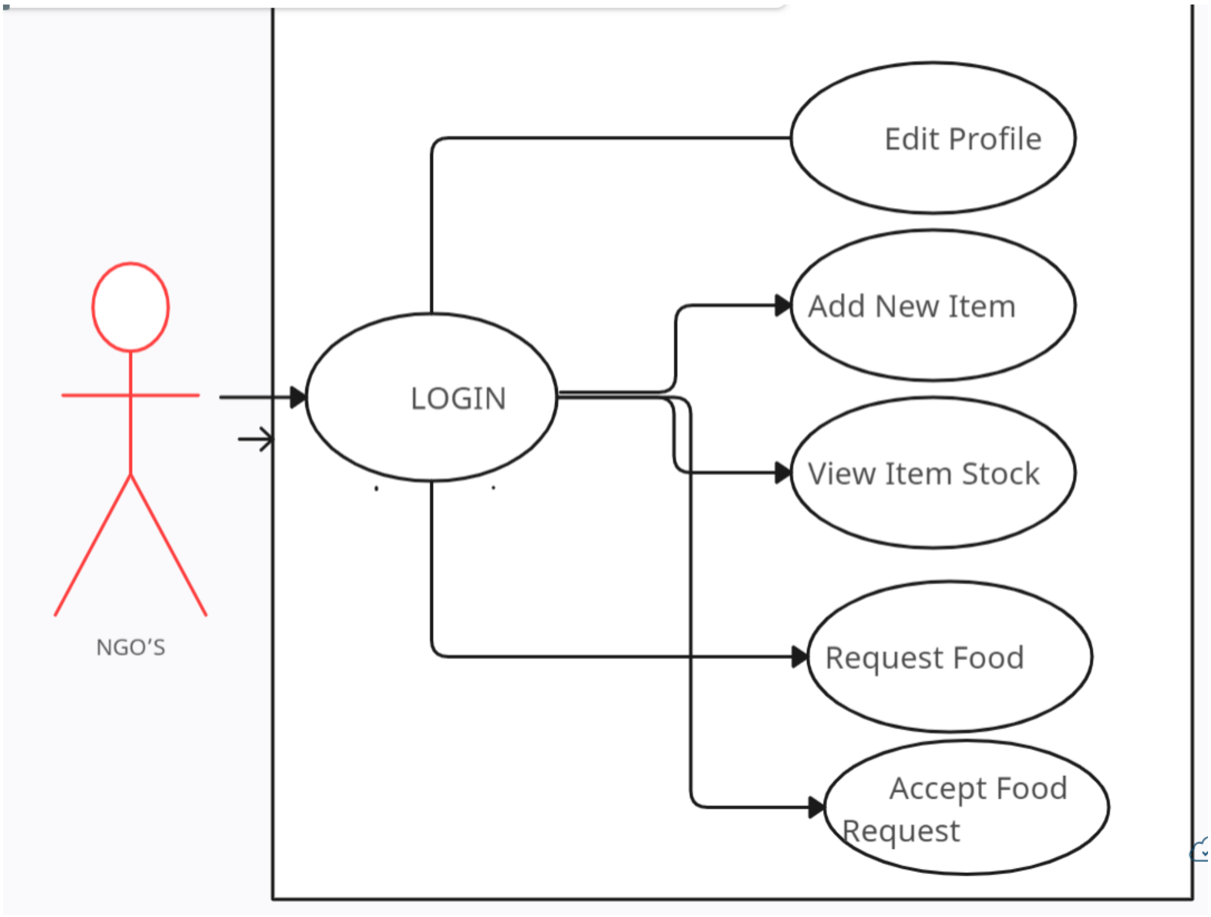
1. **Google Cloud Platform:**
    - **1.1.**Google Places Api.
2. **Firebase Functions:**
    - **2.1.**Firebase messaging service.

# CHAPTER 3

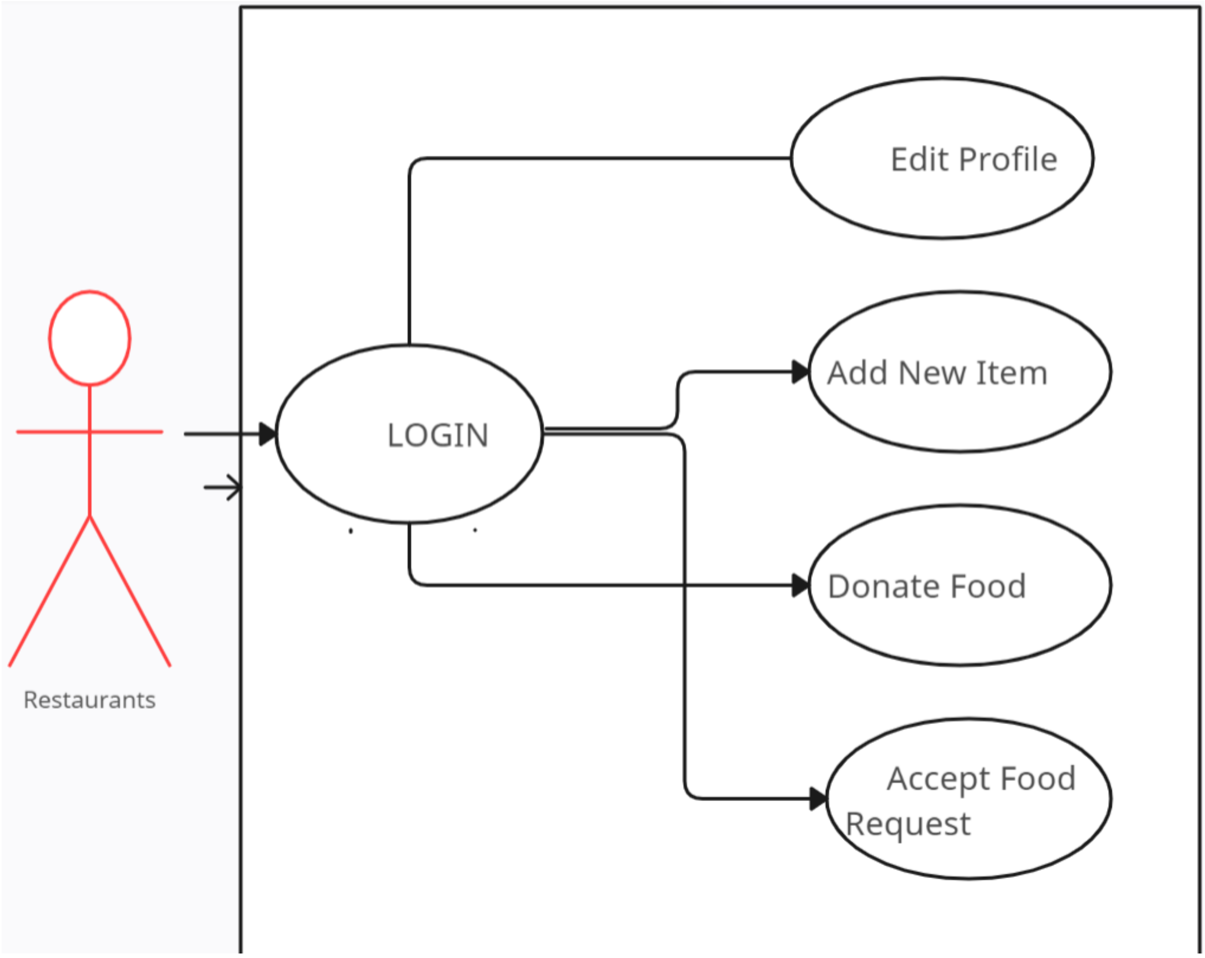
## 3. Projects diagrams

Based on the above literature review and project scope here are some diagrams, which illustrates that what will be our project or the system is capable to reach the desired results.

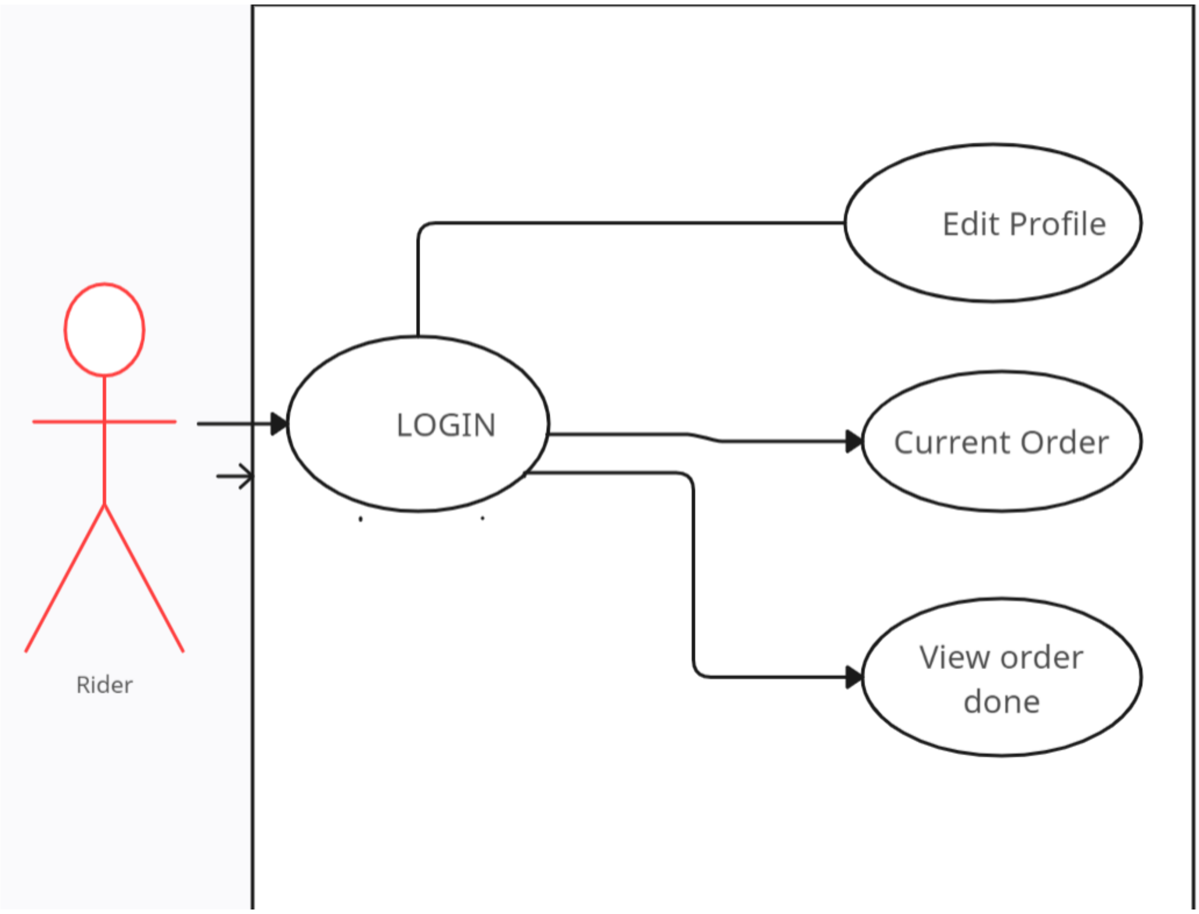
### 3.1. Use Case Diagrams



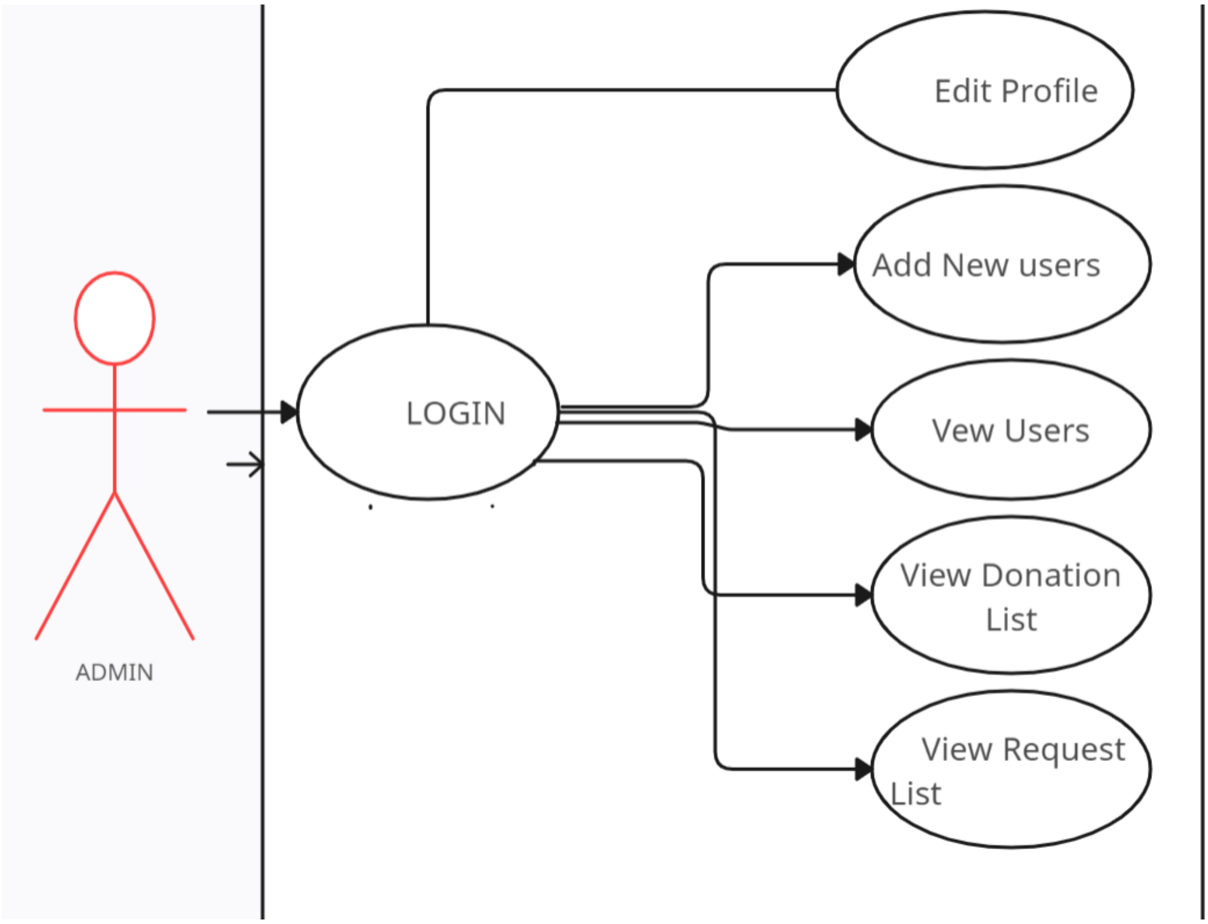
Ngos can perform its functionality only after they have login in into application. After login they can perform certain functionality through this app which are send food request to restaurants, add new items, view item in stock, accept food request of restaurants and edit their profile.



Restaurants s can perform its functionality only after they have login in into application. After login they can perform certain functionality through this app which are send food request to ngos, add new items, accept food request of ngos and edit their profile.

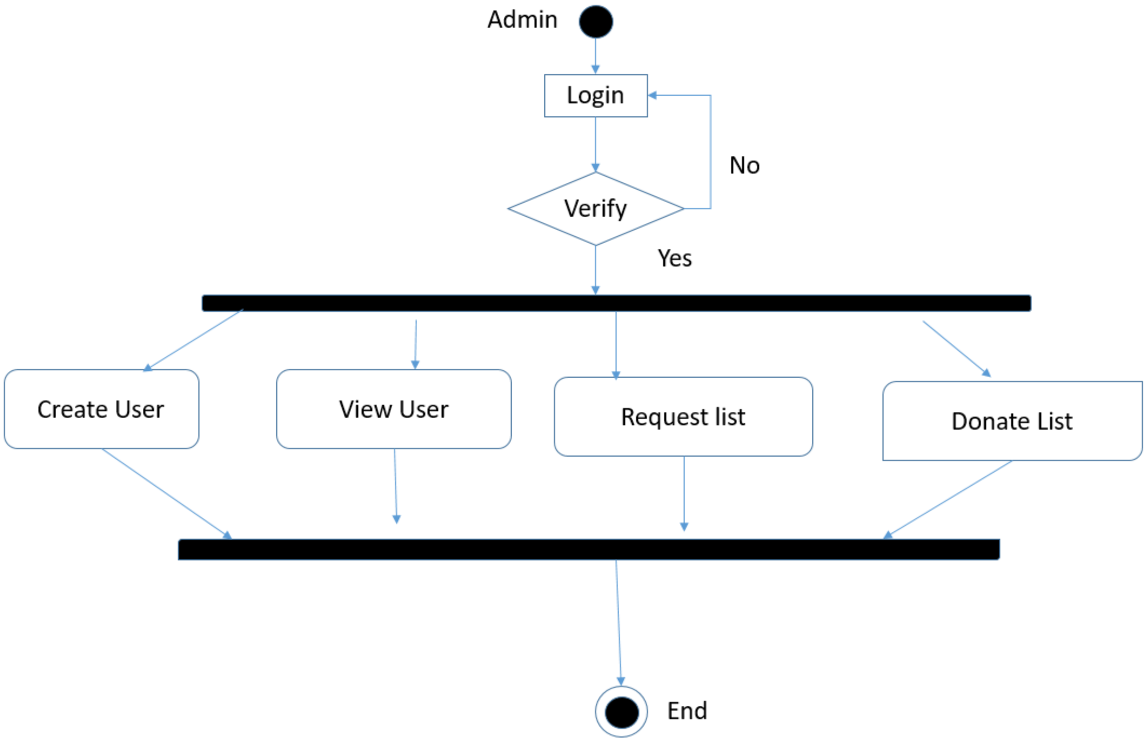


Rider can perform its functionality only after they have login in into application. After login they can perform certain functionality through this app which are view current orders, view orders that are done and edit their profile.

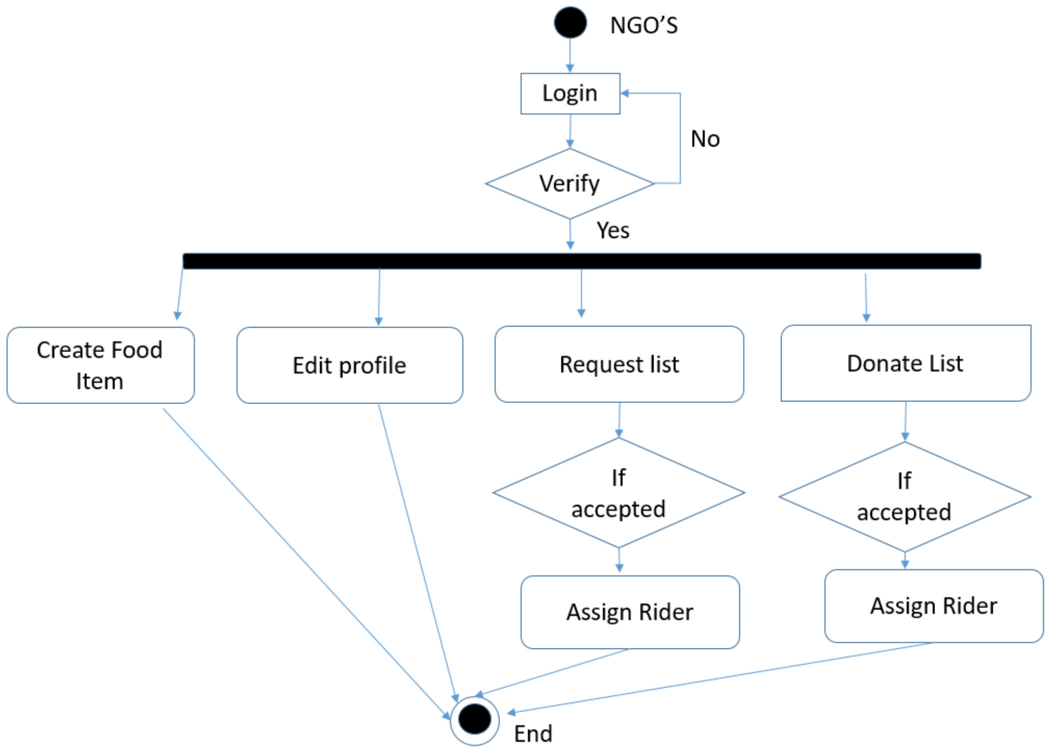


Admin can perform its functionality only after they have login in into application. After login they can perform certain functionality through this app which are add new users, view current users, view donation list and edit their profile.

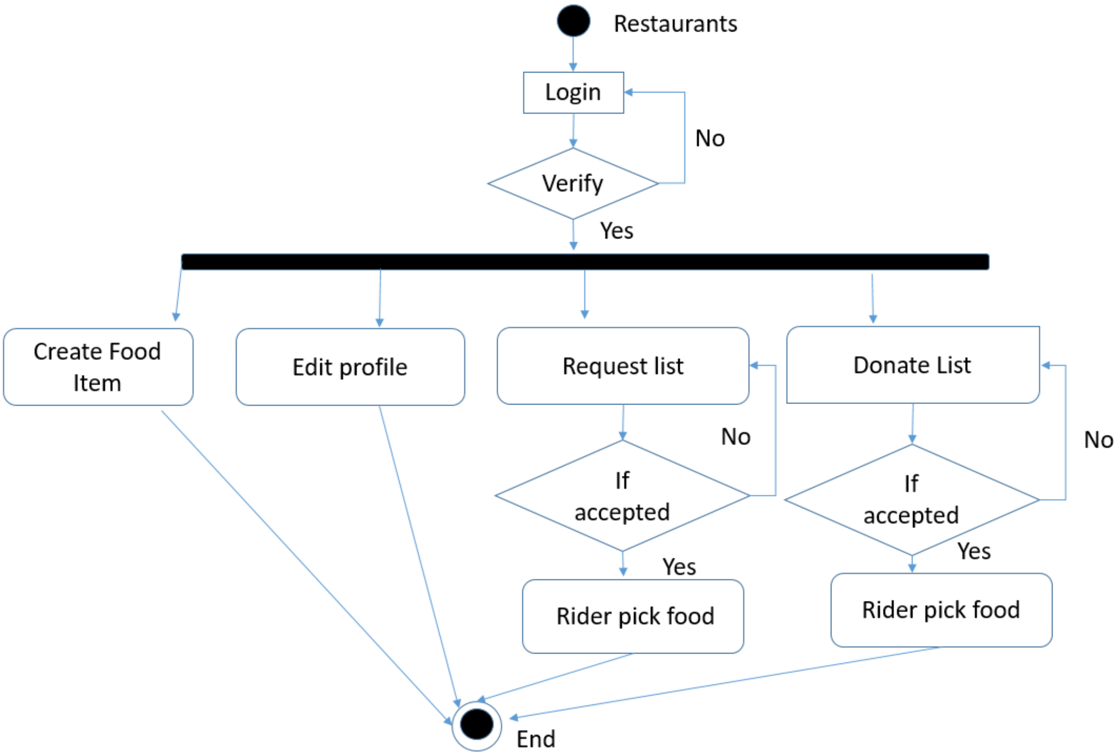
## 3.2 Activity diagram (diagram correction)



At first admin will login and if its username and password are right he will be redirected to the main page of application and if its incorrect he will remain on login page. After login in he can perform its functionality of creating users, view current users, view donation list and view request list.



At first ngo will login and if its username and password are right he will be redirected to the main page of application and if its incorrect he will remain on login page. After login in he can perform its functionality of creating food items, view donation list and view request list.If in request list he accept the request then he can assign the rider and if he rejects it he doesn't need to assign the rider.



At first restaurnats will login and if its username and password are right he will be redirected to the main page of application and if its incorrect he will remain on login page. After login in he can perform its functionality of creating food items, view donation list and view request list.If its request is accepted by  the ngo then rider will pick up food from restaurant.

## 1.1. Inside Project

### 3.4.1.Android Studio

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance your productivity when building Android apps.

### 3.4.2.Google Firebase

Firebase is an app development platform that helps you build and grow apps and games users love. Backed by Google and trusted by millions of businesses around the world.

# CHAPTER 4

## 2. Project Planning

### 2.1. Project Timeline Summary

### 2.2. Project Timeline Details

| S.no | Start Date | End Date | Week | Task | Status |
|---|---|---|---|---|---|
| 1 | 9/10/2021 | 23/10/2021 | Meeting # 01 | Introduction And Discussion | Complete |
| 2 | 23/10/2021 | 6/11/2021 | Meeting # 02 | admin Login and user Registration | Complete |
| 3 | 6/11/2021 | 20/11/2021 | Meeting # 03 | User Profile Design | Complete |
| 4 | 20/11/2021 | 4/12/2021 | Meeting # 4 | Admin Panel | Complete |
| 5 | 4/12/2021 | 18/12/2021 | Meeting # 05 | Ngo send request | Complete |
| 6 | 18/12/2021 | 1/1/2022 | Meeting # 06 | Restaurant receive request | Complete |
| 7 | 19/2/2022 | 5/3/2022 | Meeting # 07 | Restaurant send request | Complete |
| 8 | 5/3/2022 | 19/3/2022 | Meeting # 08 | Ngo recieve request | Complete |
| 9 | 19/3/2022 | 2/4/2022 | Meeting # 09 | Rider profile | Complete |
| 10 | 2/4/2022 | 16/4/2022 | Meeting # 10 | Assign order to rider to pick up food | Complete |
| 11 | 16/4/2022 | 30/4/2022 | Meeting # 11 | Mobile application for rider | Complete |
| 12 | 30/4/2022 | 14/5/2022 | Meeting # 12 | Finalizing of Application and Project Report | Complete |

## 2.3. Black-box Testing

Black Box Testing, also known as Behavioral Testing, is a software testing method in which the internal structure/ design/ implementation of the item being tested is not known to the tester. These tests can be functional or non-functional, though usually functional. Testing, either functional or non-functional, without reference to the internal structure of the component or system.

This technique is the procedure to derive and/or select test cases based on an analysis of the specification, either functional or non-functional, of a component or system without reference to its internal structure.

Black Box testing method is applicable to the following levels of software testing:

- Integration Testing
- System Testing
- Acceptance Testing

The higher the level, and hence the bigger and more complex the box, the more black box testing method comes into use.

## 2.4. Test Cases

Test cases are a crucial component of software testing and quality assurance. They are detailed, specific scenarios or conditions that are designed to evaluate whether a software application, system, or component functions correctly and meets its intended requirements. Test cases serve as a set of instructions for testers to follow, providing a structured approach to validate that the software performs as expected.

**TEST CASE # 1**

**Test Case Title: Create User**

Preconditions: User needs to register first.

Actions:

Take Steps:

1) Admin click on the Create user
2) Fill the details and click on save

| | |
|---|---|
| Expected Results: User Registered Successfully | |
| Tested By: Arqum | |
| Result | Pass |

**TEST CASE # 2**

**Test Case Title: User Login**

| | |
|---|---|
| Preconditions: User must be registered | |
| Actions<br>Take Steps<br>1) User clicks on the login button<br>2) User Enter their credentials<br>3) User clicks on sign in | |
| Expected Results: User get home screen of profile | |
| Tested By: Mustajab amin | |
| Result | Pass |

**TEST CASE # 3**

**Test Case Title:   NGO SendFood Request**

| | |
|---|---|
| Preconditions: User must be logged In | |
| Actions:<br>1) NGO click on create new order button<br>2) NGO enter food items and quantity<br>3) NGO send request | |
| Expected Results: Request Sent  to restaurant | |
| Tested By: Arqum | |
| Result | Pass |

**TEST CASE # 4**

**Test Case Title: Restaurant Accept food Request**

| | |
|---|---|
| Preconditions: User must be logged In | |
| Actions:<br>4) Restaurant click on view order<br>5) Restaurant click on edit order<br>6) Restaurant accept request | |
| Expected Results: Ngo get notification of accepted order | |
| Tested By: Arqum | |
| Result | Pass |

**TEST CASE # 5**

**Test Case Title: NGO Assign Rider**

| | |
|---|---|
| Preconditions: User must be logged In | |
| Actions:<br>7) NGO click on view order<br>8) NGO click on edit order<br>9) NGO assign rider | |
| Expected Results: rider get notification of order | |
| Tested By: Mustajab | |
| Result | Pass |

**TEST CASE # 6**

**Test Case Title: Rider deliver Food**

| | |
|---|---|
| Preconditions: User must be logged In and online | |
| Actions:<br>10) Rider click on view order<br>11) Rider click on map for location<br>12) After delivering food rider click on mark as deliver button. | |
| Expected Results: all user get notification of complete order. | |
| Tested By: Mustajab | |
| Result | Pass |

**TEST CASE # 7**

**Test Case Title: Restaurant Send Food Request**

| | |
|---|---|
| Preconditions: User must be logged In | |
| Actions:<br>13) Restaurant click on create new order button<br>14) Restaurant enter food items and quantity<br>15) Restaurant send request | |
| Expected Results: Request Sent to ngo | |
| Tested By: Arqum | |
| Result | Pass |

**TEST CASE # 8**

**Test Case Title: Ngo Accept food Request**

| | |
|---|---|
| Preconditions: User must be logged In | |
| Actions:<br>16) NGO click on view order<br>17) NGO click on edit order<br>18) NGO accept request | |

| | |
|---|---|
| Expected Results: Restaurant get notification of accepted order | |
| Tested By: Arqum | |
| Result | Pass |

**TEST CASE # 9**

**Test Case Title: Ngo Assign Rider**

| | |
|---|---|
| Preconditions: User must be logged In | |
| Actions:<br>19) NGO click on view order<br>20) NGO click on edit order<br>21) NGO assign rider | |
| Expected Results: rider get notification of order | |
| Tested By: Mustajab | |
| Result | Pass |

**TEST CASE # 10**

**Test Case Title: Rider deliver Food**

| | |
|---|---|
| Preconditions: User must be logged In and online | |
| Actions:<br>22) Rider click on view order<br>23) Rider click on map for location<br>24) After delivering food rider click on mark as deliver button. | |
| Expected Results: all user get notification of complete order. | |
| Tested By: Mustajab | |
| Result | Pass |

**TEST CASE # 11**

**Test Case Title: Restaurant reject Food Request**

| | |
|---|---|
| Preconditions: User must be logged In and online | |
| Actions:<br>25) Restuarst click on view order<br>26) Restaurant  select an orders and rejects it. | |
| Expected Results: The NGO should receive a notification indicating that the restaurant has rejected the food request. | |
| Tested By: Mustajab | |
| Result | Pass |

**TEST CASE # 12**

**Test Case Title:   Ngo reject Food Request**

| | |
|---|---|
| Preconditions: User must be logged In and online | |
| Actions:<br>27) Ngo click on view order<br>28) Ngo select an orders and rejects it. | |
| Expected Results: The Restaurant should receive a notification indicating that the ngo has rejected the food request. | |
| Tested By: Mustajab | |
| Result | Pass |

**TEST CASE # 13**

**Test Case Title:   Rider Rejects to deliver Food**

| | |
|---|---|
| Preconditions: User must be logged In and online | |
| Actions:<br>29) Rider click on view order<br>30) Rider selects the assigned delivery and rejects it. | |
| Expected Results: All relevant parties (NGO, restaurant, and rider) should receive | |

| | |
|---|---|
| notifications indicating that the rider has rejected the delivery. | |
| Tested By: Mustajab | |
| Result | Pass |

**TEST CASE # 14**

**Test Case Title: User Profile Update**

| | |
|---|---|
| Preconditions: User must be logged In and online | |
| Actions:<br>31) User clicks on "Edit Profile."<br>32) User updates their profile information.<br>33) Clicks on the "Save" button. | |
| Expected Results: User's profile information should be updated successfully. | |
| Tested By: Mustajab | |
| Result | Pass |

**TEST CASE # 15**

**Test Case Title: Invalid User Registration**

| | |
|---|---|
| Preconditions: None | |
| Actions:<br>34) User attempts to register with invalid or incomplete details.<br>35) Clicks on the "Save" button. | |
| Expected Results: User receive an error message indicating that registration was unsuccessful due to invalid or incomplete details. | |
| Tested By: Arqum | |
| Result | Pass |

**TEST CASE # 16**

**Test Case Title: User Logout**

| | |
|---|---|
| Preconditions: User must be logged In and online | |
| Actions:<br>36) User clicks on the "Logout" button. | |
| Expected Results: User should be logged out and redirected to the login page. | |
| Tested By: Mustajab | |
| Result | Pass |

**TEST CASE # 17**

**Test Case Title: Invalid User Login**

| | |
|---|---|
| Preconditions: User must be registered. | |
| Actions:<br>37) User enters incorrect login credentials.<br>38) Clicks on the "Sign In" button. | |
| Expected Results: User should receive an error message indicating that the login attempt failed due to incorrect credentials. | |
| Tested By: Mustajab | |
| Result | Pass |

**TEST CASE # 18**

**Test Case Title: Password Reset**

| | |
|---|---|
| Preconditions: User must be registered and logged out. | |
| Actions:<br>39) User clicks on the "Forgot Password" link.<br>40) User enters their email address and requests a password reset. | |
| Expected Results: ser should receive a password reset email with instructions on resetting the password. | |
| Tested By: Mustajab | |
| Result | Pass |

### TEST CASE # 19

| | |
|---|---|
| **Test Case Title: Order History**<br>Preconditions: User must be logged in. | |
| Actions:<br>41) User clicks on "Order History." | |
| Expected Results: user should see a list of their past orders. | |
| Tested By: Mustajab | |
| Result | Pass |

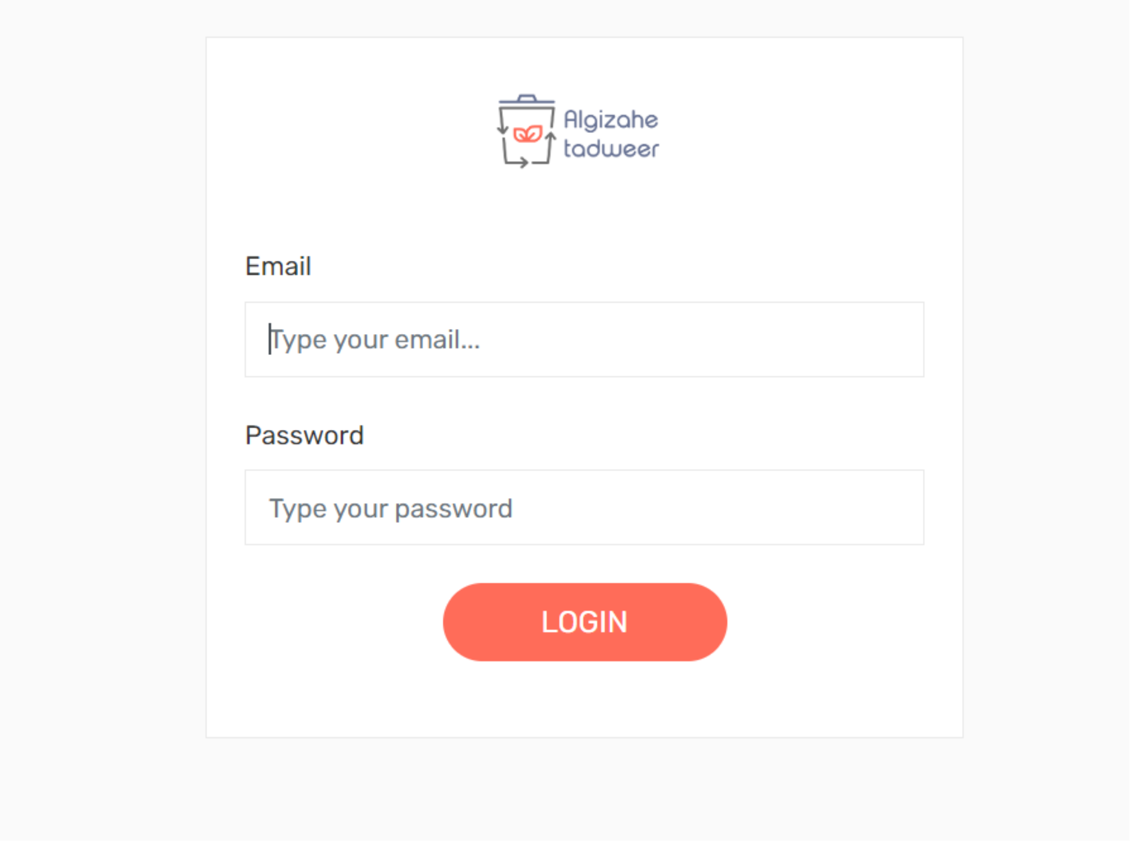
### TEST CASE # 20

| | |
|---|---|
| **Test Case Title: User Profile Picture Upload**<br>Preconditions: User must be logged in. | |
| Actions:<br>42) User clicks on "Edit Profile."<br>43) User uploads a profile picture.<br>44) Clicks on the "Save" button. | |
| Expected Results: The user's profile picture should be successfully uploaded and displayed in their profile. | |
| Tested By: Mustajab | |
| Result | Pass |

# CHAPTER 5

## 3. GUI of Web Application

### 3.1. Login Interface



Algizahe tadweer

Email

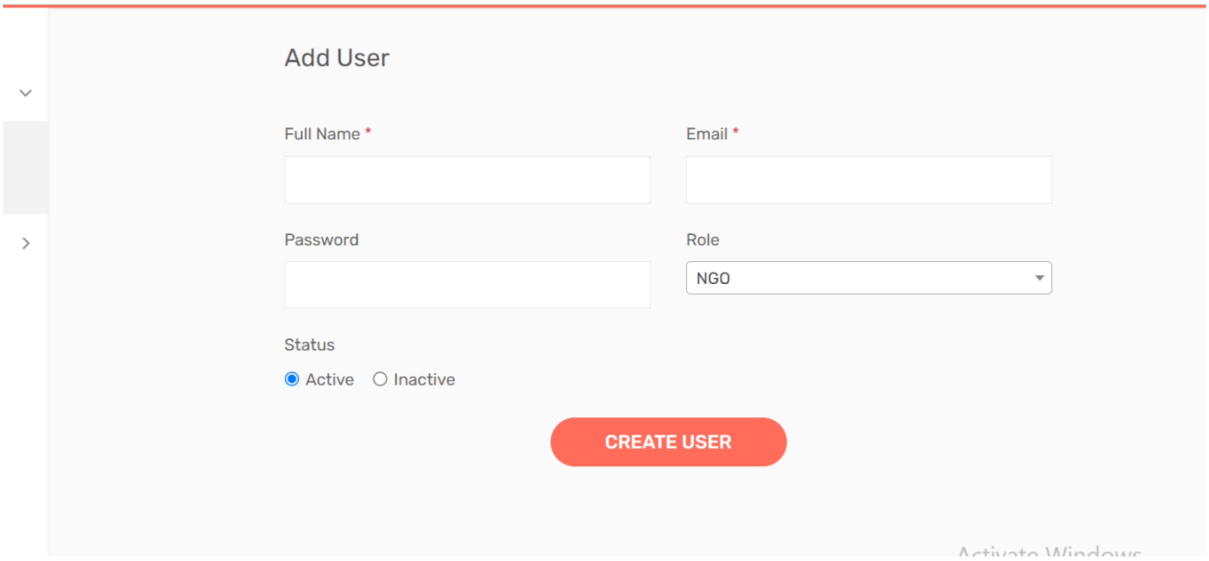
Type your email...

Password

Type your password

Login interface will be same for all users either you are ngo , restaurant or rider

### 3.2. Sign up Interface


Add User

Full Name *

Email *

Password

Role

NGO

Status

● Active ○ Inactive

Signup interface will also be same for all users they can be differentiated into different users by selecting their roles such as ngo, restaurant or rider.

# CHAPTER 6

## 4. Conclusion and Future Work

### 4.1. Limitation

- This project is deployed locally not globally.
- This system will counter one request at a time cannot handle multiple request.
- This web app will only work for ngos and restaurants.
- Prior to registration they cannot either ask for food or donate it.

### 4.2. Conclusion

In conclusion, we have developed a web app which allow the users (i.e., Ngos or Restaurants) to get on a single platform which is Algizae tadweer through which users can request leftover food from anywhere which can be deliver by rider at their respective location. With our app ngo don’t need to check out with every restaurant to ask for food and in a same manner restaurant doesn’t have to search for ngo to donate food.

### 4.3. Future Works

There is always a chance of improvement, following are the aspects where the system requires some time to be analyzed and modified

- Improve quality of UI Design.
- Improve quality of Database.
- Enhance Application Functionality and Features.
- Allow multiple request at a time.
- Donate food without registration.

# APPENDIX

## **Admin.php:**

```
<?php defined('BASEPATH') OR exit('No direct script access allowed');

class Admin extends MY_Controller {
public $data = [];
function __construct() {
        parent::__construct();
        if(!$this->session->userdata('id') || $this->session->userdata('role') != 'admin'){
                return redirect(base_url('login'));
        }
        $this->data["current_user_id"] = $this->user_id = $this->session->userdata('id');
}//end construct

public function index(){
        redirect(base_url('admin/profile'));
}//index

public function dashboard(){
        $this->load->view('dashboard', $this->data);
}//index

public function profile(){
        $this->data['current_slug'] = 'My Profile';

        if($_POST){
                if($this->form_validation->run('profile') == FALSE){
                        $this->data['user_profile']     =       $this->users->view_profile($this-
>user_id);
                        $this->load->view('profile', $this->data);
                }else{
                        $post     = $this->input->post();
```

```
            $user_pass = $this->input->post('user_pass');
            $img_id    = $this->input->post('img_id');

            if(!empty($user_pass)){
                if(strlen($user_pass) >= 6){
                    $post['user_pass'] = md5($this->input->post('user_pass'));
                }else{
                    $this->session->set_flashdata('error', 'Password must be graterthan 5 digits');
                    redirect(base_url('admin/profile'));
                }
            }else{
                if(isset($_POST['user_pass'])){
                    unset($post['user_pass']);
                }
            }
            $profile_pic = $_FILES['profile_pic'];

            if(!empty($profile_pic['name'])){
                $img_path = 'profile/admin/';
                if($new_name = $this-> _upload_file($profile_pic['name'], $profile_pic['type'], $profile_pic['tmp_name'], $profile_pic['size'], $img_path)){
                    $post['profile_pic'] = $new_name;
                    unlink('./assets/'.$img_path.$img_id);
                }else{
                    $post['profile_pic'] = $img_id;
                }
                unset($post['img_id']);
            }else{
                $post['profile_pic'] = $img_id;
                unset($post['img_id']);
            }
            if($this->users->update_profile($post, $this->user_id)){
                $this->session->set_flashdata('feedback',"Profile Updated
Successfully.");
                redirect(base_url('admin/profile'));
            }else{
                $this->session->set_flashdata('error',"Profile Not Update. Plz
Try Again!.");
                redirect(base_url('admin/profile'));
            }//model
        }//validation
    }else{
        $this->data['user_profile'] = $this->users->view_profile($this->user_id);
        $this->load->view('profile', $this->data);
    }//post
}//profile

public function users($action="", $id=0){
    if($action=="add"){
        if($_POST){
            if($this->form_validation->run('create_user') == FALSE){
                $this->session->set_flashdata('error',validation_errors());
                redirect(base_url('admin/users/add'));
            }else{
                $post = $this->input->post();
                $post['user_pass'] = md5($post['user_pass']);

                if($this->users->create_user($post)){
                    $this->session->set_flashdata('feedback',"Create User
Successfully.");
                    redirect(base_url('admin/users/manage'));
                }else{
                    $this->session->set_flashdata('error',"User Not Create.
Plz Try Again!.");
                    redirect(base_url('admin/users/manage'));
                }//end if else model
            }//end else valid
        }else{
            $this->data['current_slug'] = 'Add User';
            $this->data['current_page'] = 'add_user';
            $this->load->view('admin/users/add', $this->data);
        }//end else post
    }//add

    if($action=="edit"){
        if($_POST){
            $post = $this->input->post();
            unset($post['user_pass']);
            $user_pass = $this->input->post('user_pass');

            if(!empty($user_pass)){
                if(strlen($user_pass) >= 6){
                    $post['user_pass'] = md5($user_pass);
                }else{
                    $this->session->set_flashdata('error', 'Password must be
graterthan 5 digits');
                    redirect(base_url('admin/users/edit'));
                }
            }

            if($this->users->update_profile($post, $id)){
                $this->session->set_flashdata('feedback',"User Updated
Successfully.");
                redirect(base_url('admin/users/manage'));
            }else{
                $this->session->set_flashdata('error',"User Not Update. Plz Try
Again!.");
                redirect(base_url('admin/users/manage'));
            }//end if else model
        }else{
            $this->data['rs_view'] = $this->users->view_profile($id);
            $this->data['current_slug'] = 'Edit User';
            $this->data['current_page'] = 'users';
            $this->load->view('admin/users/edit', $this->data);
        }//end else post
    }//edit

    if($action=="manage"){
        $this->data['user_id']   = $this->user_id;
        $this->data['rs_manage'] = $this->users->view_all_user();
        $this->data['current_page'] = 'user_list';
        $this->data['current_slug'] = 'User List';
        $this->load->view('admin/users/manage', $this->data);
    }//manage
}//users

public function signout(){
    $this->_signout();
}//signout

}
```

## A. Gantt chart

| Task | Oct | Nov | Dec | Jan | Feb | Mar | April | May |
|---|---|---|---|---|---|---|---|---|
| Do the Literature Survey | 🟩 | | | | | | | |
| Design the Application | 🟩 | 🟩 | | | | | | |
| Design the Database | | 🟩 | 🟩 | 🟩 | | | | |
| Design User Interface | | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | | |
| Develop Application | | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 | |
| Built Database | | 🟩 | 🟩 | | | | | |
| Built User Interface | | | 🟩 | 🟩 | | 🟩 | | |
| Test Application | | | 🟩 | 🟩 | | | | 🟩 |
| Test Database | | | 🟩 | 🟩 | | 🟩 | | 🟩 |
| Test User Interface | | 🟩 | | 🟩 | 🟩 | 🟩 | 🟩 | 🟩 |
| Maps Integration | | 🟩 | | 🟩 | 🟩 | | | |
| Application Integration | | | | | 🟩 | 🟩 | | |
| Perform Integration Testing | | | | | 🟩 | 🟩 | 🟩 | 🟩 |
| Write the Proposal | | | | | | | | 🟩 |
| Write Final Report | | | | | | | | 🟩 |
| Presentation | | | | | | | | 🟩 |