ARYABHATTA COLLEGE

UNIVERSIRTY OF DELHI

SOFTWARE ENGINEERING PROJECT

WASTE FOOD MANAGEMENT SYSTEM

By

Saksham Kheterpal (CSC/20/53)

Shubham Kumar (CSC/20/70)

To

DR. Deepak Sharma

ACKNOWLEDGEMENT

Apart from the efforts of team, the success of any project depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. The completion of any inter-disciplinary project depends upon cooperation, co-ordination and combined efforts of several sources of knowledge. We are eternally grateful to our teacher Dr. Deepak Sharma for his even willingness to give us valuable advice and direction under whom we executed this project. His constant guidance and willingness to share his vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the assigned tasks. Although there may be many who remain unacknowledged in this humble note of gratitude, there are none who remain unappreciated.

CERTIFICATE

This is to certify that Software Engineering project report entitled "Waste Food Management System" is the work carried out by Saksham Kheterpal , Shubham Kumar students of B.Sc(H) Computer Science IV Sem, Aryabhatta College, New Delhi , University of Delhi under the supervision of Dr Deepak Sharma .This report has not been submitted to any other organisation/institution for the award any other degree/diploma.

PROBLEM STATEMENT

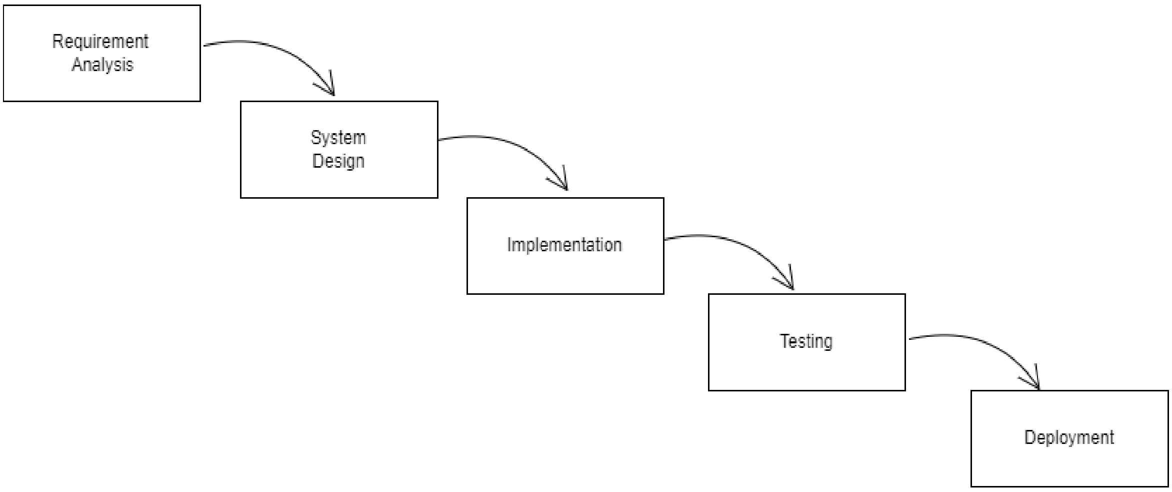
A drastic increase can be seen in food waste. As per recent data given by the Food Waste Index Report 2021, A staggering 50kg of food is thrown away per person every year in Indian homes, that is approximately 68.8 million tonnes of food waste ranking India second in the world for total food waste per year. Hence there is a need to come up with a solution that can avoid food waste & can help feed the needy

This Food Waste Management system can assist in collecting the leftover food from hotels & restaurants to distribute among those in need. NGOs that are helping poor communities to battle against starvation & malnutrition can raise a request for food supply from restaurants through this application. Once the request is accepted, the NGOs can collect the food from the restaurants for its distribution. In this way this android-based food waste management system will help restaurants to reduce food waste and will help in feeding the poor and needy people.

PROCESS MODEL

We choose Waterfall model because of the following reasons:

- It is relatively simple and easier to understand approach as compared to other models
- The requirements are well stated and understood before in hand
- In this model we have to complete one stage before proceeding to the next. So we have clearly defined stages and well understood milestones
- The requirements are fixed and work can proceed to completion in a linear manner
- The waterfall model provides a structured approach



CONTENTS

1.**SOFTWARE REQUIREMENT SPECIFICATION**

1.1 **Overall Description**

1.1.1 **Product Functions**

The Waste Food Management System Software is an independent web based application. There are various user interfaces related with software. These interfaces help the user to interact with the software and provide the necessary information for online food donation/request.

The entire functionality of this software can be divided into modules. The name of the modules involved in the waste food management system are listed below:

1. User Login/Registration
   Using this module, the admin/user provides his details like username and password to login into the application.

2. Food Donation System
   This module allows the user to donate food. Users are required to fill in the type of food, Food name, quantity and Address for pickup of donated food.

3. Register Restaurant/NGO
   This module allows the user to register their restaurant or NGO and mention the food they require.

1.1.2 **User Characteristics**

- The user of the system should be comfortable working with English language.
- The users must have a basic knowledge of computers and internet

1.1.3 **General Constraints**

- The database of the software is not accessible to every user of the software. Making changes in the database requires permissions that have been given to certain specific individuals

1.2.4 **Assumptions and Dependencies**

- The “Waste Food Management System” is assumed to be compatible with the computer system on which it has been loaded for customer use

1.2 **External Interface Requirements**

1.2.1 **User Interfaces**

- Front-end software: HTML and CSS
- Back-end software: PHP and SQL

1.2.2 **Hardware Interfaces**

- Laptop/Desktop PC
- Wi-Fi Router

1.2.3 **SOFTWARE INTERFACES**

Following are the software used for the Waste Food Management application

Software Used	Description
Operating System	We have chosen windows operating system for its best support and user-friendliness
Database	To save user records, Restaurant/NGO details . we have chosen SQL database
MySQL server	Used for Database Connectivity and Management
Visual Studio 1.6.3.2	Used to Write and Test our Web Application Code
XAMPP 1.8.0	To test our web application on a local test server

1.3 **Functional Requirements**

1.3.1 Add User

- The admin has the option to register a new user to the system
- Admin can choose to register either a new user or an admin

1.3.2 View Users

- Admin can view user information such as their user id and role

1.3.3 Delete User

- Admin has the ability to delete specific users from the database

1.3.4 Register NGO

- User can register their own NGO or Restaurant by filling in basic information

1.3.5 Register Food

- Users can Register Food by filling in information such as quantity, food type and food name for collection on the web application

1.3.6 Take Food

- User can take food they require from the web application
- User must enter NGO Id and choose food name and quantity required

1.3.7 View Food

- Users have the option to view food available on the application

1.3.7 View NGOs

- Users can view registered NGOs on the application

1.4 **Performance Requirements**

The product shall be based on web and has to be run from a web server. The product shall take initial load time depending on internet connection strength which also depends on the media from which the product is run. The performance shall depend upon hardware components of the client/customer.

1.5 **Design Constraints**

1.5.1 Standard Development Tools

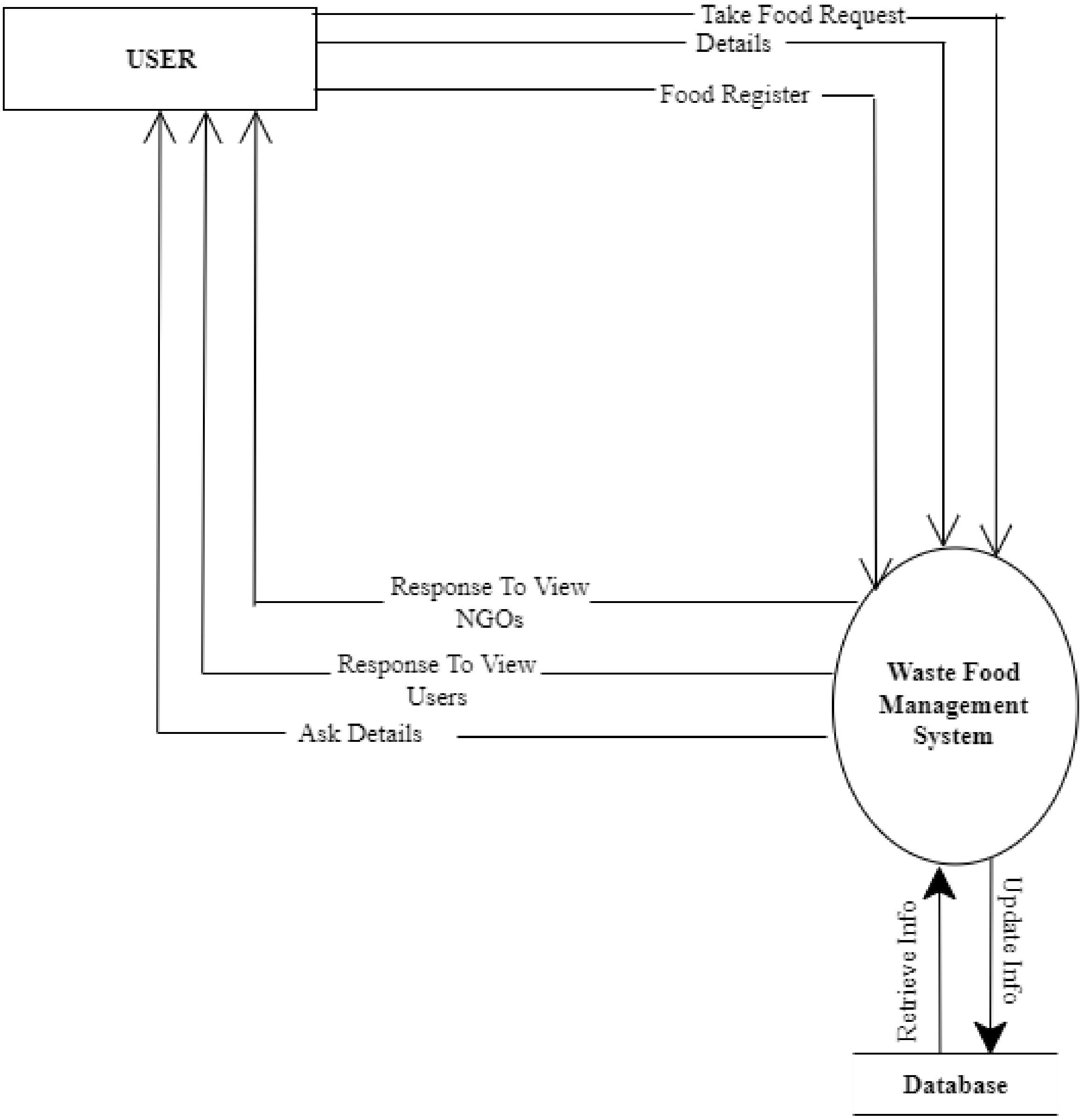
- The system shall be built using a standard web page development tool that conforms to either IBM's CUA standards or Microsoft's GUI standards.

1.5.2 Web Based Product

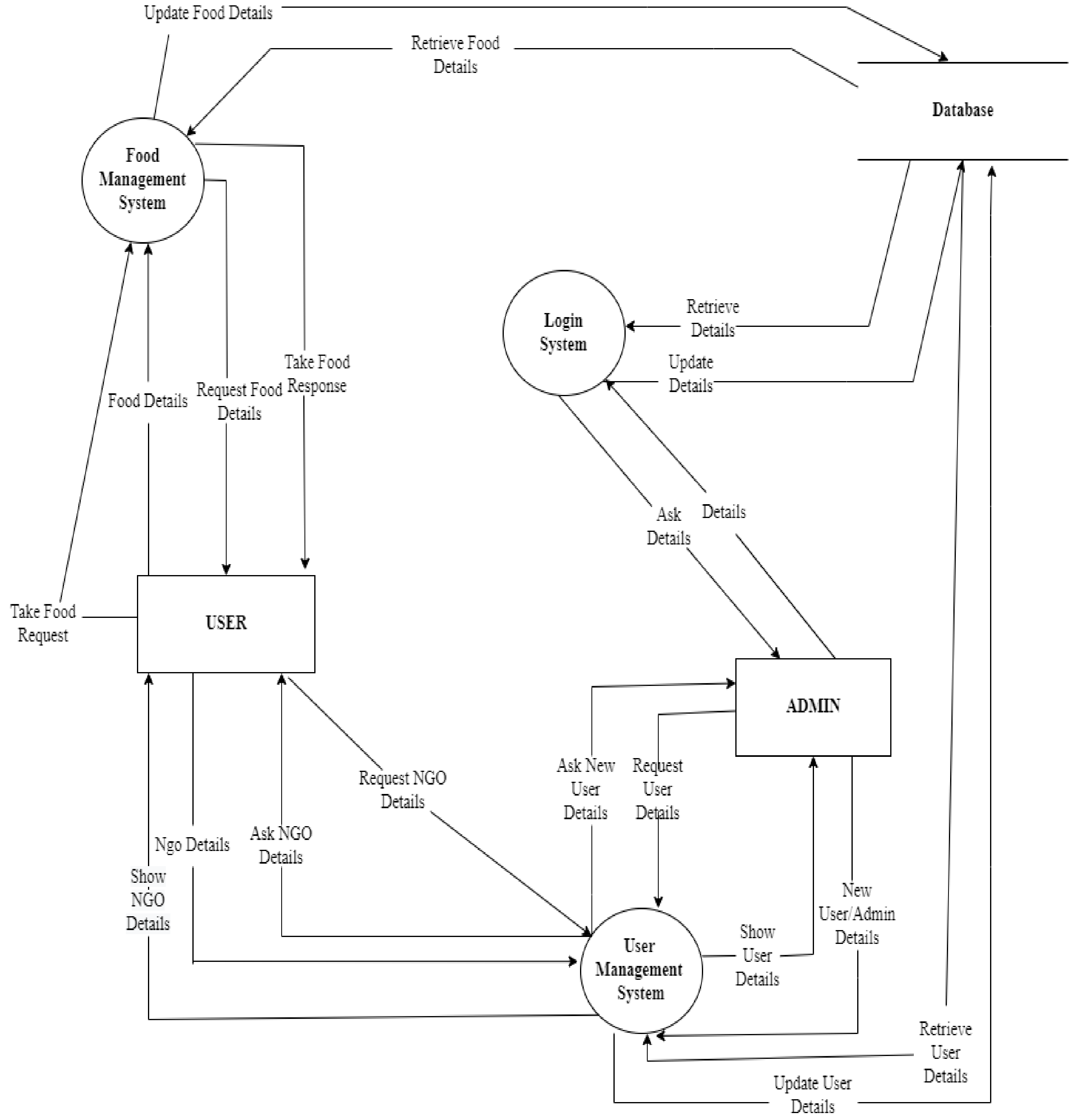
- There are no memory requirements
- The computers must be equipped with web browsers such as Internet explorer.
- The product must be stored in such a way that allows the client easy access to it.
- Response time for loading the product should take no longer than five minutes.
- A general knowledge of basic computer skills is required to use the product

1.6 **Data Flow Diagram**

1.6.1 Context Level DFD



1.6.2 DFD Level 1



1.7 **Data Dictionary**

Field Name	Data Type	Field Size
Food id	Int	10
Food name	Varchar	50
Quantity taken	Int	15
Quantity Left	Int	15
Taken Food Id	int	100
Food Type name	Varchar	50
Food Type id	Int	10
Username	Varchar	50
Password	Varchar	50
Role	Varchar	10
Ngo Name	Varchar	65
Ngo No	Varchar	65
Contact	Int	10
Ngo Id	Int	10
Status	Varchar	10
State	Varchar	100
State Id	Int	50
Reference number	Int	250
User Id	Int	50
Date Entry	Date	30

1.8 **Use Cases and Use Case Diagram**

1.8.1 **Admin Use Case**

Use Case: Login

Description

The admin can login by entering correct username and password

Use Case: Register User

Description

The admin can register a new user and give them admin rights

Use Case: View Users

Description

The admin has access to view all the registered users

Use Case: Delete User

Description

The admin has access to delete an existing user

1.8.2 Non-Admin User Use Case

Use Case: Register NGO

Description

User register a new NGO by entering the required details

Use Case: Take Food

Description

User takes food for their NGO by entering their NGO id and choosing required food and quantity from the list of food available

Use Case: Register Food

Description

If User runs a restaurant, hotel, etc, they can register leftover food by entering food name and quantity

Use Case: View NGOs

Description

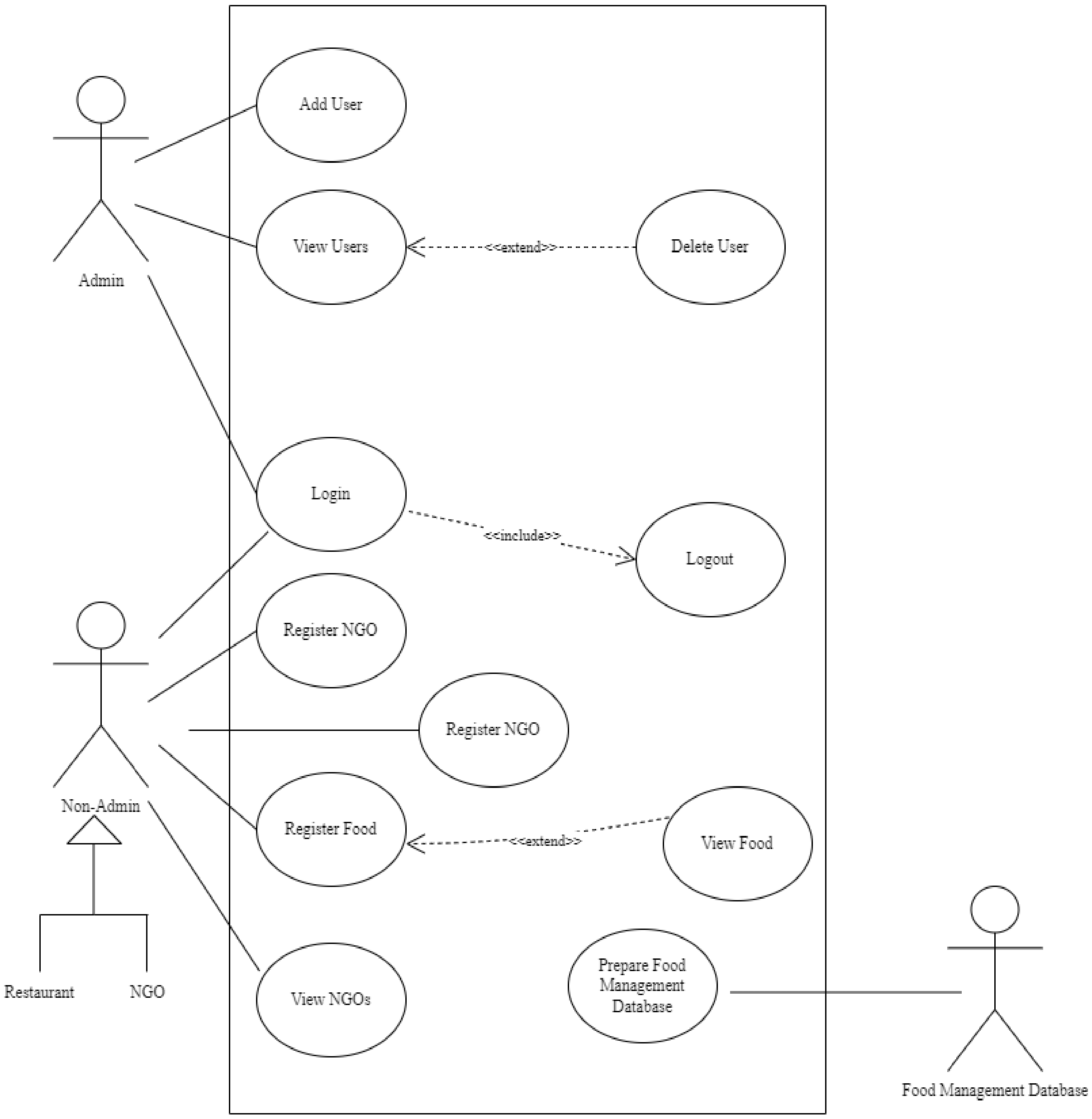
User can view all the names of the registered NGOs along with all their basic information

Use Case: View Food

Description

User can view Food available before registering/taking it

1.8.3 Use Case Diagram



2. Estimations

2.1 Function Points

The function point (FP) metric can be used effectively as a means for measuring the functionality delivered by a system.4 Using historical data, the FP metric can then be used to (1) estimate the cost or effort required to design, code, and test the software; (2) predict the number of errors that will be encountered during testing; and (3) forecast the number of components and/or the number of projected source lines in the implemented system. Function points are derived using an empirical relationship based on countable (direct) measures of software's information domain and qualitative assessments of software complexity. Information domain values are defined in the following manner:

Number of external inputs (EIs): Each external input originates from a user or is transmitted from another application and provides distinct application-oriented data or control information. Inputs are often used to update internal logical files (ILFs). Inputs should be distinguished from inquiries, which are counted separately.

Number of external outputs (EOs): Each external output is derived data within the application that provides information to the user. In this context external output refers to reports, screens, error messages, etc. Individual data items within a report are not counted separately.

Number of external inquiries (EQs): An external inquiry is defined as an online input that results in the generation of some immediate software response in the form of an online output (often retrieved from an ILF).

Number of internal logical files (ILFs): Each internal logical file is a logical grouping of data that resides within the application's boundary and is maintained via external inputs.

Number of external interface files (EIFs): Each external interface file is a logical grouping of data that resides external to the application but provides information that may be of use to the application.

the table in the figure below is completed and a complexity value is associated with each count. Organizations that use function point methods develop criteria for determining whether a particular entry is simple, average, or complex. Nonetheless, the determination of complexity is somewhat subjective.

To compute function points (FP), the following relationship is used:

FP= count total x [0.65 +0.01 x ∑ Fi]

Parameters	Counts	Low	Medium	High
Number of Inputs		3	4	6
Number of Outputs		4	5	7
Number of Files		3	4	6
Number of External Interfaces		5	10	15
Number of User Inquiries		7	7	10

	DATA ELEMENTS		
FTR	**1-4**	**5-15**	**>15**
0-1	Low	Low	Average
2	Low	Average	High
3 or more	Average	High	High

External Inputs

1. Login Module

1.1	Username	Data Elements	FTR	Complexity
1.2	Password	2	1	Low

2. User Registration Module

2.1 Username	Data Elements	FTR	COMPLEXITY
2.2 Password	6	2	AVG
2.3 Confirm Password			
2.4 First Name			
2.5 Last Name			
2.6 Role			

3. NGO Registration Module

3.1 NGO Name	Data Elements	FTR	COMPLEXITY
3.2 NGO No	4	2	LOW
3.3 Contact			
3.4 Address			

4. Food Registration Module

4.1 Food Name	Data Elements	FTR	COMPLEXITY
4.2 Quantity	3	2	LOW
4.3 Food Type			

5. Take Food Module

5.1 NGO No	Data Elements	FTR	COMPLEXITY
5.2 Select Food	3	2	LOW
5.3 Quantity			

Total External Inputs = 18

- 12= Low Complexity
- 6=Average Complexity

External Outputs

Admin

- View Users

User

- View Registered NGO's/Restaurants
- View Taken Food
- View Available Food

All have Simple Complexity

Total External Outputs = 4

Logical Internal Files

- Login File
- Admin File
- User File
- Menu File

Complexity is Simple For all Logical Internal Files

Total Logical Internal Files = 4

External Interface Files = 0

External Inquiries

User – view taken food

Complexity is Simple for all external inquiries

Total External Inquiries = 1

UFP= (12*3+6*4) + 4*4 + 4*7 +1*3

UFP=107

Considering all adjustment factors have average influence

CAF = 0.65 + 0.01*14*3

CAF= 1.07

FP=UFP*CAF

FP=107*1.07

FP=114.49

Hence Function Point Count = 114.49

2.2 Efforts

The average productivity for this kind of system = 13.5 FP/PM

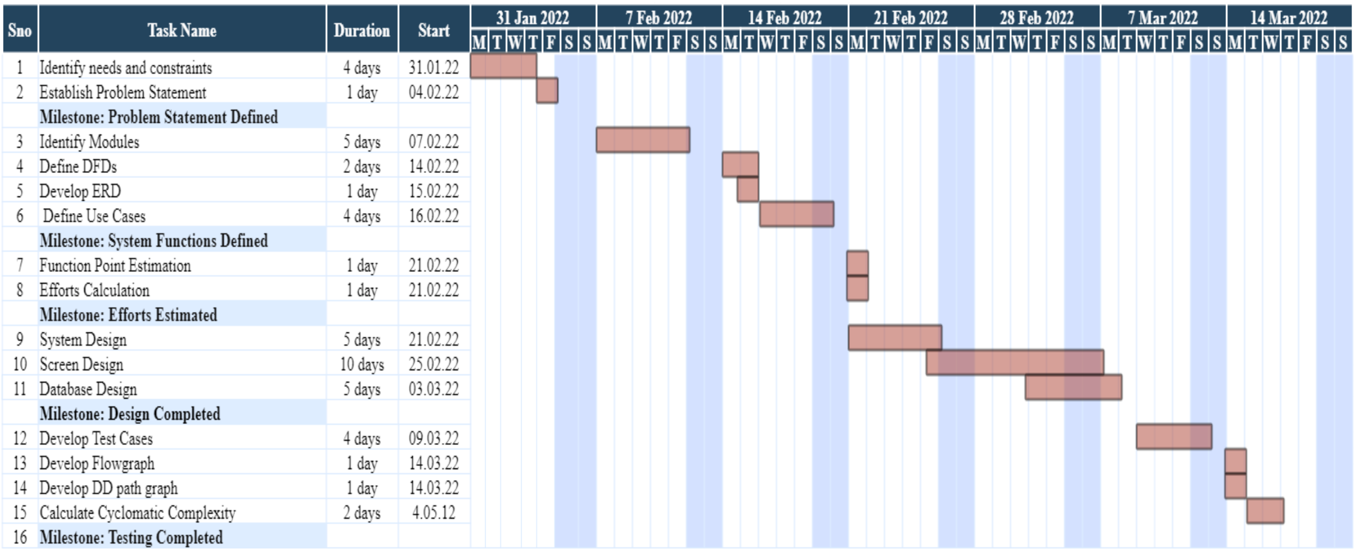
Total Efforts = FP (calculated)/ average productivity

= 114.49/13.5

= 8.48 person months

3. Scheduling

3.1 Gantt Chart



Sno	Task Name	Duration	Start
1	Identify needs and constraints	4 days	31.01.22
2	Establish Problem Statement	1 day	04.02.22
	Milestone: Problem Statement Defined		
3	Identify Modules	5 days	07.02.22
4	Define DFDs	2 days	14.02.22
5	Develop ERD	1 day	15.02.22
6	Define Use Cases	4 days	16.02.22
	Milestone: System Functions Defined		
7	Function Point Estimation	1 day	21.02.22
8	Efforts Calculation	1 day	21.02.22
	Milestone: Efforts Estimated		
9	System Design	5 days	21.02.22
10	Screen Design	10 days	25.02.22
11	Database Design	5 days	03.03.22
	Milestone: Design Completed		
12	Develop Test Cases	4 days	09.03.22
13	Develop Flowgraph	1 day	14.03.22
14	Develop DD path graph	1 day	14.03.22
15	Calculate Cyclomatic Complexity	2 days	4.05.12
16	**Milestone: Testing Completed**		

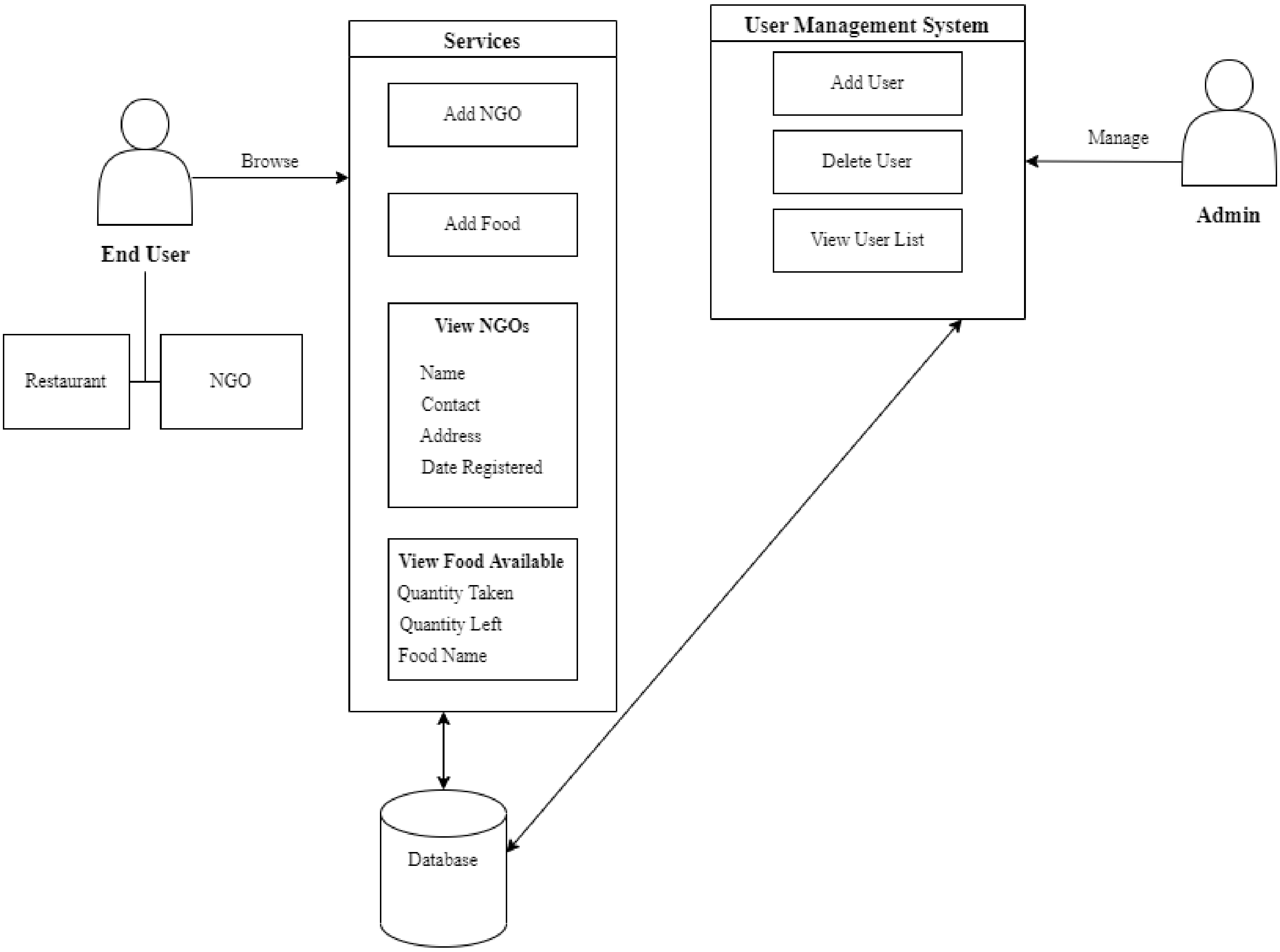
4. Risk Management

4.1 Risk Table

Risks	Category	Probability	Impact	RMMM Plan
Quality not maintained	DE	60%	3	Take up steps to maintain quality at each stage of development
Size estimates may be low	PS	50%	2	Past experiences must be considered and a similar task on a smaller scale may be attempted
Requirements not properly documented and understood	CU	50%	1	Getting the requirements completely verified before finalising them
Delivery deadline will be tightened	BU	40%	2	Review the progress from time to time and take appropriate steps to keep up with the schedule
Lack of Skill	ST	40%	2	External resources might be helpful
Building the wrong product	CU	20%	1	Early and continuous validation is critically important. You need to establish a clear vision and solid justification for the product

5. Design

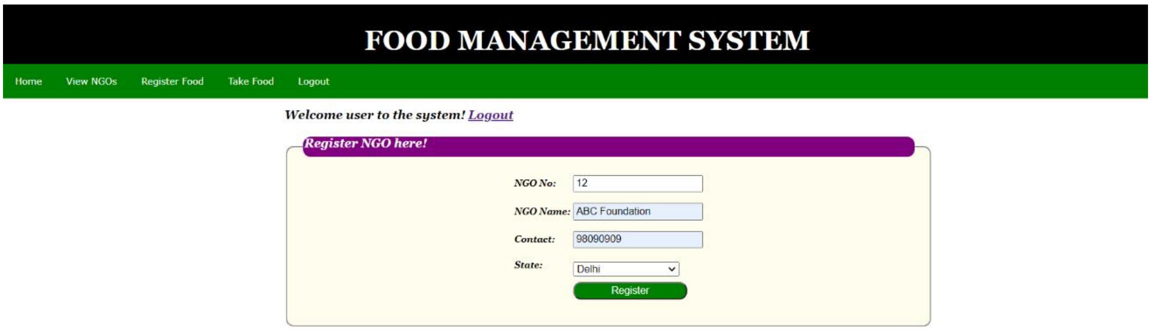
5.1 System Design



5.2 Screen Design

Screenshots of screens visible to the user

USER VIEW



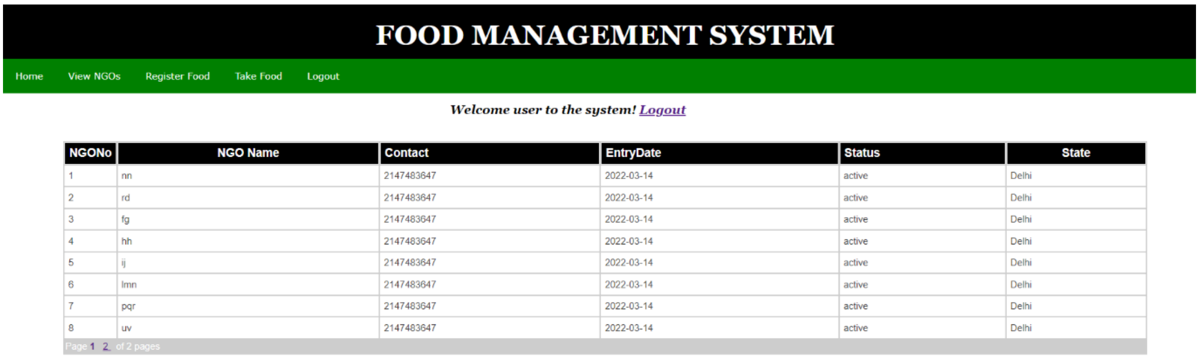
FOOD MANAGEMENT SYSTEM

Welcome user to the system! *Logout*

Register NGO here!

NGO No:	12
NGO Name:	ABC Foundation
Contact:	98090909
State:	Delhi

NGO Registration Page



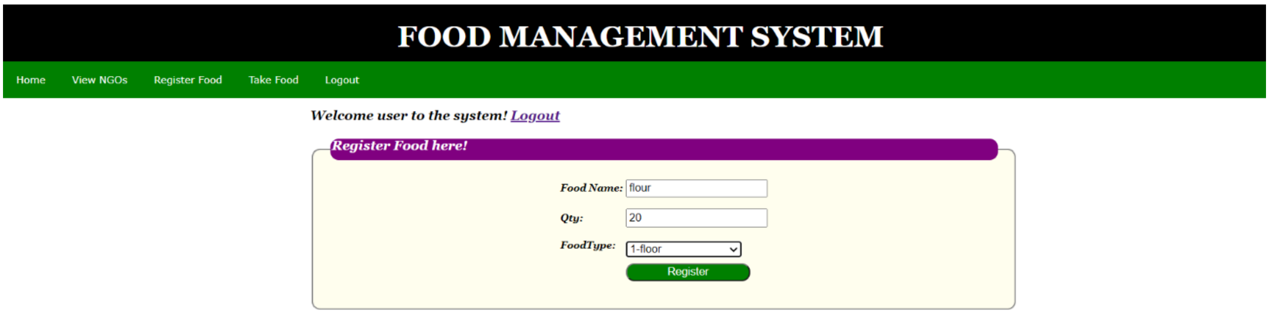
FOOD MANAGEMENT SYSTEM

Welcome user to the system! *Logout*

NGONo	NGO Name	Contact	EntryDate	Status	State
1	nn	2147483647	2022-03-14	active	Delhi
2	rd	2147483647	2022-03-14	active	Delhi
3	fg	2147483647	2022-03-14	active	Delhi
4	hh	2147483647	2022-03-14	active	Delhi
5	ij	2147483647	2022-03-14	active	Delhi
6	lmn	2147483647	2022-03-14	active	Delhi
7	pqr	2147483647	2022-03-14	active	Delhi
8	uv	2147483647	2022-03-14	active	Delhi

Page 1 2 of 2 pages

View NGOs Page



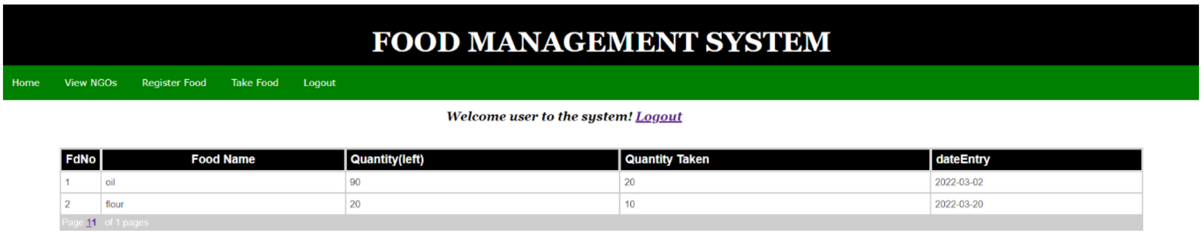
FOOD MANAGEMENT SYSTEM

Welcome user to the system! *Logout*

Register Food here!

Food Name:	flour
Qty:	20
FoodType:	1-floor

Food Registration Page

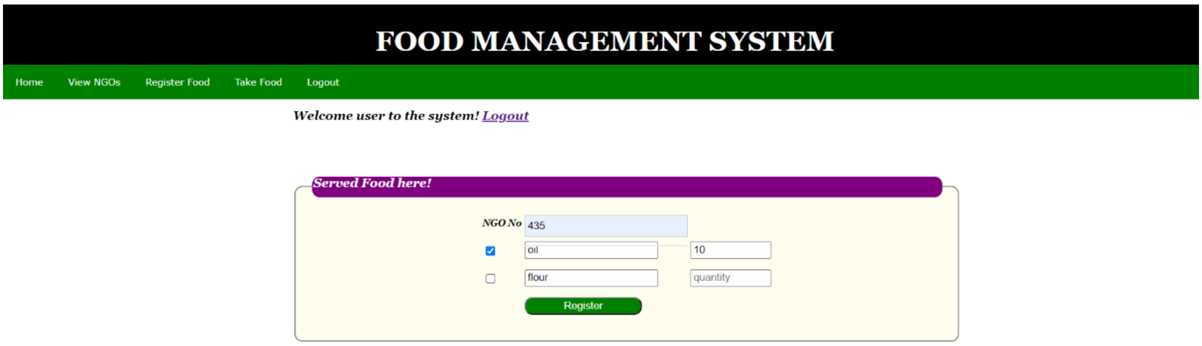


Welcome user to the system! Logout

FdNo	Food Name	Quantity(left)	Quantity Taken	dateEntry
1	oil	90	20	2022-03-02
2	flour	20	10	2022-03-20

Page 11 of 1 pages

View Food Page



Welcome user to the system! Logout

Served Food here!

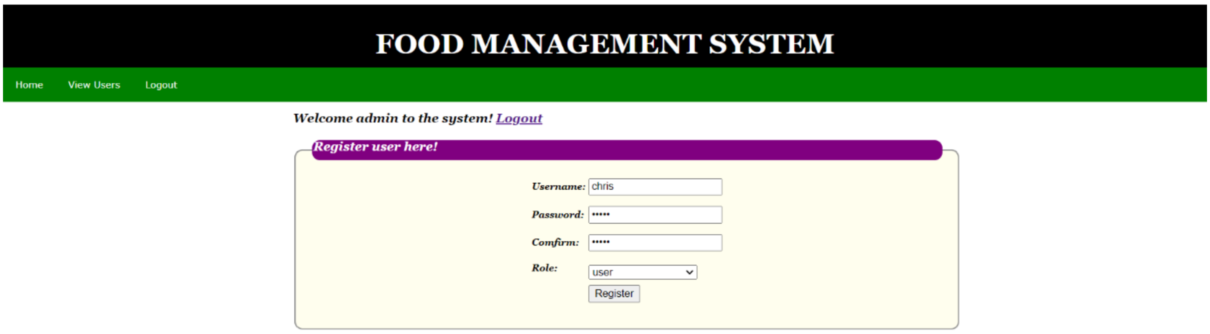
NGO No: 435

☑ oil 10

□ flour quantity

Take Food Page

ADMIN VIEW



Welcome admin to the system! Logout

Register user here!

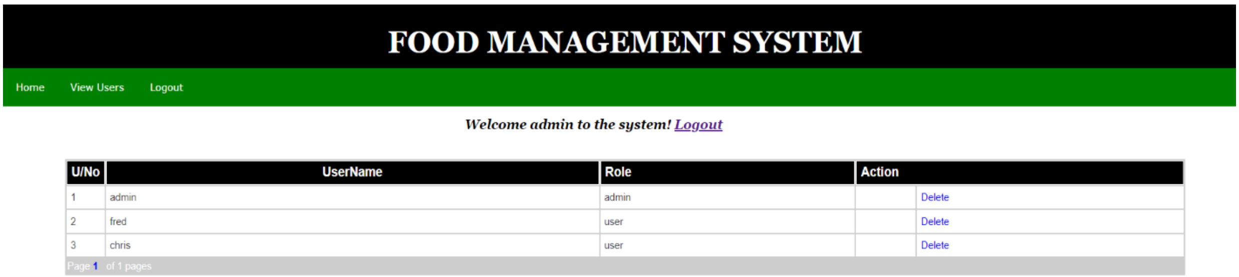
Username: chris

Password: •••••

Comfirm: •••••

Role: user

Registration Page



FOOD MANAGEMENT SYSTEM

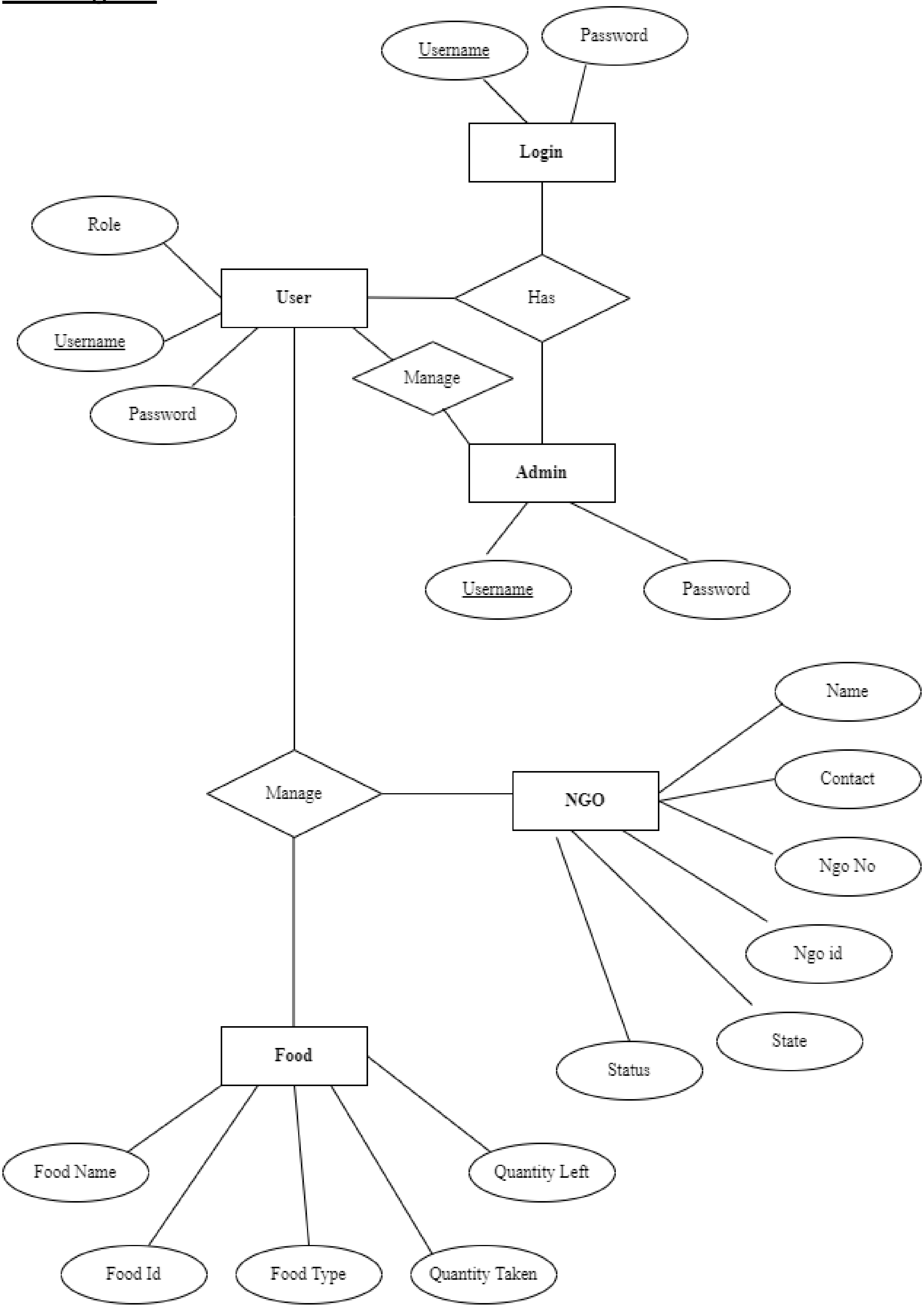
Welcome admin to the system! *Logout*

U/No	UserName	Role	Action	
1	admin	admin		Delete
2	fred	user		Delete
3	chris	user		Delete

Page 1 of 1 pages

View Users Page

5.3 ER Diagram



5.4 Database Design

Database: foodmanagement

- Table: foodreg

Field Name	Data Type
Fdid	int
Foodname	varchar
Qtyleft	int
Qtytaken	int
dateEntry	Date
foodType_typeid	int

- Table: foodtype

Field Name	Data Type
Typeid	int
Typename	varchar

- Table: ngo

Field Name	Data Type
Ngoid	int
Ngoname	varchar
Contact	int
dateEntry	Date
state_sid	int
refNo	int

- Table: users

Field Name	Data Type
Userid	int
Username	varchar
Password	varchar
Role	varchar

- Table: state

Field Name	Data Type
Sid	int
StateName	varchar

- Table: takenfood

Field Name	Data Type
Tfid	int
takenDate	Date
Qty	int
foodreg_ngoid	int
NGO_ngoid	int

6. Coding

```
CREATE TABLE IF NOT EXISTS `foodreg` (
  `fdid` int(11) NOT NULL AUTO_INCREMENT,
  `foodname` varchar(205) NOT NULL,
  `qtyleft` int(205) NOT NULL,
  `qtytaken` int(45) NOT NULL,
  `dateEntry` date NOT NULL,
  `footType_typeid` int(200) NOT NULL,
  PRIMARY KEY (`fdid`),
  KEY `fk_foodreg_footType` (`footType_typeid`)
)
CREATE TABLE IF NOT EXISTS `foodtype` (
  `typeid` int(200) NOT NULL AUTO_INCREMENT,
  `typename` varchar(205) NOT NULL,
  PRIMARY KEY (`typeid`)
)
CREATE TABLE IF NOT EXISTS `state` (
  `sid` int(200) NOT NULL AUTO_INCREMENT,
  `StateName` varchar(45) NOT NULL,
  PRIMARY KEY (`sid`)
)
```

1

```
CREATE TABLE IF NOT EXISTS `NGO` (
  `ngoid` int(11) NOT NULL AUTO_INCREMENT,
  `ngoname` varchar(205) NOT NULL,
  `contact` int(200) NOT NULL,
  `dateEntry` date NOT NULL,
  `status` varchar(205) NOT NULL,
  `state_sid` int(200) NOT NULL,
  `refNo` varchar(205) NOT NULL,
  PRIMARY KEY (`ngoid`),
  KEY `fk_NGO_state` (`state_sid`)
)
CREATE TABLE IF NOT EXISTS `users (
  `userid` int(11) NOT NULL AUTO_INCREMENT,
  `username` varchar(205) NOT NULL,
  `password` varchar(205) NOT NULL,
  `role` varchar(205) NOT NULL ,
  PRIMARY KEY (`staffid`)
)


CREATE TABLE IF NOT EXISTS `takenfood` (
  `tfid` int(11) NOT NULL AUTO_INCREMENT,
  `takenDate` date NOT NULL,
  `qty` int(205) NOT NULL,
  `foodreg_ngoid` int(200) NOT NULL,
  `NGO_ngoid` int(200) NOT NULL,
  PRIMARY KEY (`tfid`),
  KEY `fk_takenFood_foodreg` (`foodreg_ngoid`),
  KEY `fk_takenFood_NGO` (`NGO_ngoid`)
)
```

1

```
<legend>Welcome <?php echo $_SESSION['SESS_LAST_NAME'];?>
 to the system!
<fieldset>
<table align="center">                                                   2
                    <?php
                    $pos=$_SESSION['SESS_LAST_NAME'];

if($pos=='user') {
          include 'admin.php';                                3
          }
          else
                    {
                              include ("addusers.php");             4
}



<li class="dropdown">
<a href="regfood.php" class="dropbtn">Register Food</a>               5

<div class="dropdown-content">
 <a href="viewfd.php">ViewFood</a>                                  6
    </div>
 </li>

</li>

  <li class="dropdown">
<a href="served.php">Take Food</a>                                    7
```

```
</li>


  <li><a href="../logout.php">Logout</a><li>                    8
</ul>


<a href="pagenateNGOList.php">View NGOs </a></li>               9


<?php
}
if($pos=='admin') {
            ?>
  <li><a href="viewusers.php">View Users</a></li>              10
  <li><a href="../logout.php">Logout</a><li>                   11
</ul>
<?php
}
?>
```

```
<tr>
<td><label>Food Name:<label></td>
<td ><input type="text" autocomplete="off" name="a" tabindex="2" required />
</td>
</tr>
<tr>
<td><label>Qty:<label></td>
<td ><input type="text" autocomplete="off" name="b" tabindex="2" required />
</td>
</tr>
<tr>
<td><label>FoodType:<label></td>
<td >
<select name="c" tabindex="2" required />
<option>Select<-------->value</option>
<?php
$result = mysql_query("SELECT * FROM  foodtype ORDER BY typeid  ASC");
while( $row = mysql_fetch_array($result)){
echo '<option value="'.$row['typeid'].'">'.$row['typeid'].'-'.$row['1'].'</option>';
}
?>
</select>
<td><input type="submit" name="Register" value="Register"/></td>
```

12

```
<table class="tablesorter"  align="center">
                  <thead>
                   <th width="5">FdNo</th>
                 <th> <center> Food Name</center></th>
                   <th>Quantity(left)</th>
                   <th >Quantity Taken</th>
                   <th>dateEntry</th>
                   </thead>
                  <tbody>
                   <?php
                     $sno = 0;
            while($e=mysql_fetch_array($result)){
            $sno = $sno + 1;                      ?>
 <tr class = "table_row"><td><?php echo $sno; ?></td>
            <td><?php echo $e['1'];?></td>
            <td ><?php echo $e['2'];?></td>
            <td><?php echo $e['3'];?></td>
            <td><?php echo $e['4'];?></td>
            <?php }}
```

13

```
<tr>
<td><input type="checkbox" name="food[]" value="<?php echo $id;?> " required>
</td>
<td><input type="text" name="a[]" tabindex="2"value="<?php echo $fdname;?>
"readonly required /></td>
<td><input type="text" placeholder="quantity" name="b[]"  size="10"required />
</td>
</td>
</tr>
<?php
}
?>
<tr>
<td></td>
<td><input type="submit"
```

14

```
<?php
session_start();
 if (! isset ( $_SESSION ['auth'] ) || $_SESSION ['auth'] == false) {
            session_cache_expire();
            session_destroy();
            session_unset();
    header ( "Location: index.php" );
    exit ();
```

15

```
}
$per_page =8;
$pages_query =  mysql_query
("SELECT COUNT(ngoid)
FROM  NGO
WHERE NGO.status='active' ");
$pages = ceil(mysql_result($pages_query, 0) / $per_page);
$page = (isset($_GET['page'])) ? (int)$_GET['page'] : 1;
$start = ($page - 1) * $per_page;
$query="select *
 from  NGO, state
WHERE NGO.state_sid= state.sid
AND NGO.status='active'LIMIT $start, $per_page";
```

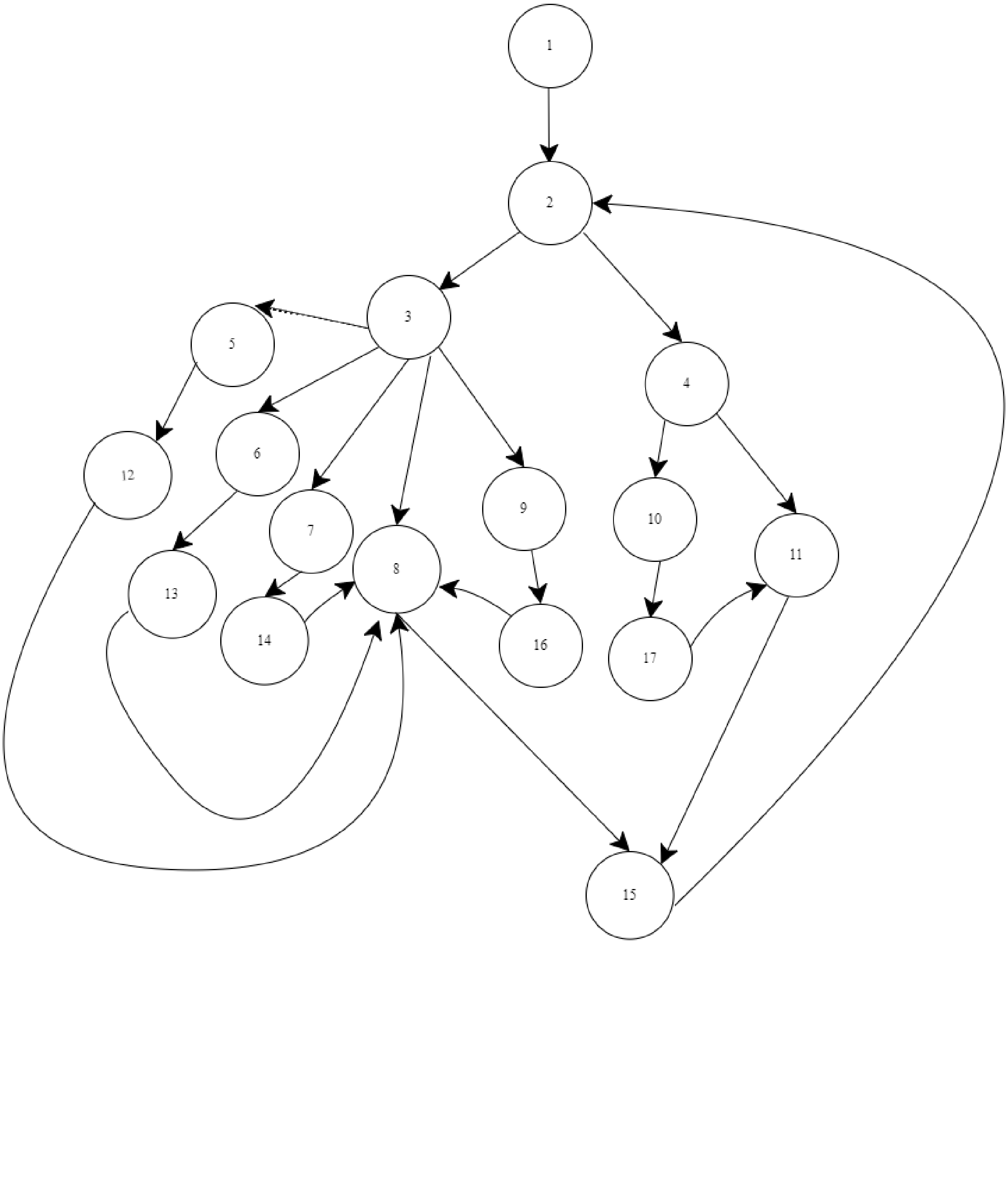
16

```
$per_page =11;
 $pages_query =  mysql_query
("SELECT COUNT(staffid)
FROM  staff");
$pages = ceil(mysql_result($pages_query, 0) / $per_page);
$page = (isset($_GET['page'])) ? (int)$_GET['page'] : 1;
$start = ($page - 1) * $per_page;
$query="select *
 from  staff
 LIMIT $start, $per_page";
```

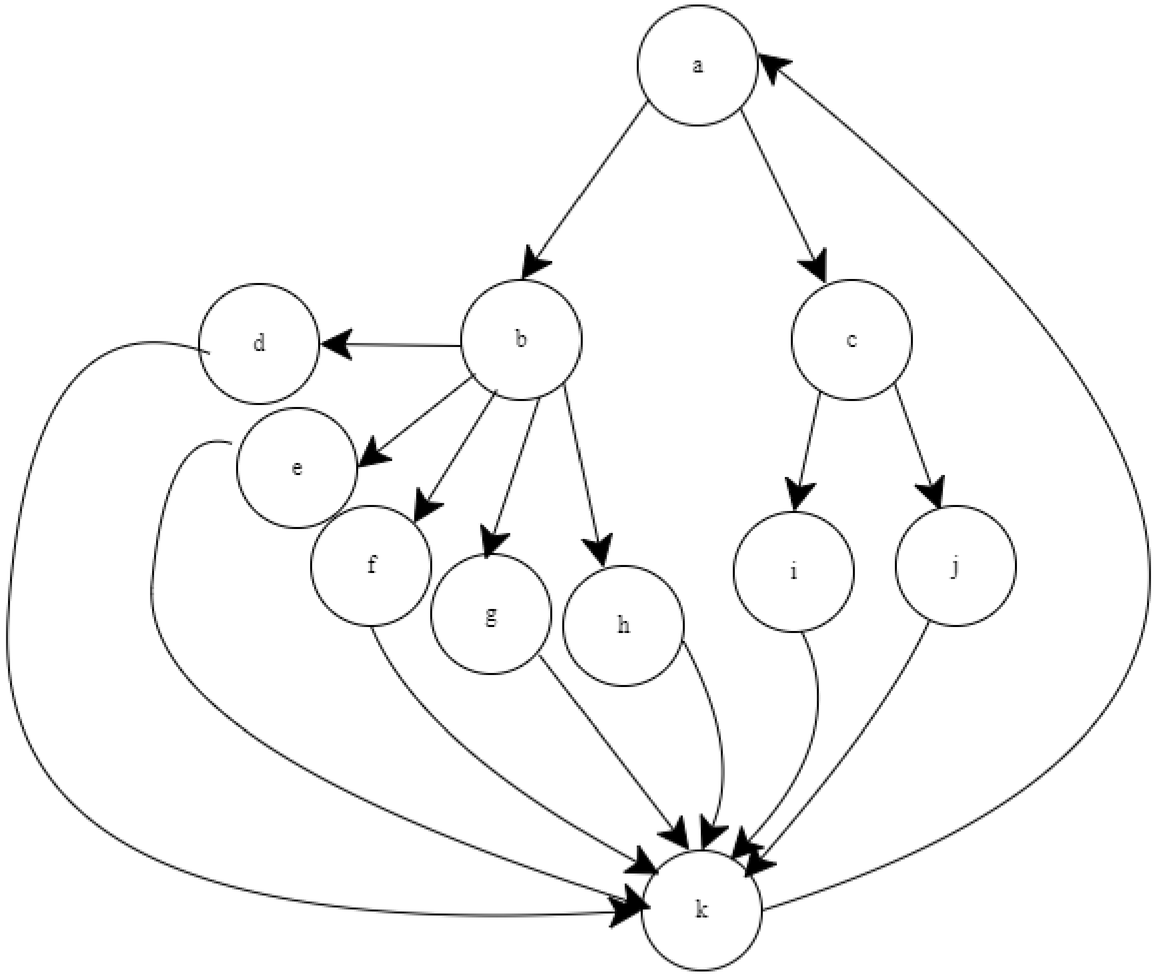
17

7.Testing

7.1 Flowgraph



7.2 DD Path Graph



7.3 Cyclomatic Complexity

V(G)= E-N+2

Where e = Number of edges

N= No of nodes

From the Above Flowgraph

E=22

N=17

Hence

V(G)=22-17+2

=7

OR

V(G) = R

Where R = Number of Regions

Hence V(G) = 7

Cyclomatic Complexity=V(G)=7

Hence Number of Independent Paths will be 7

7.3.1 Independent Paths

Path 1: 1,2,3,5,12,8,15

Path 2: 1,2,3,6,13,8,15

Path 3: 1,2,3,7,14,8,15

Path 4: 1,2,3,8,15

Path 5: 1,2,3,9,16,8,15

Path 6: 1,2,4,10,17,11,15

Path 7: 1,2,4,11,15

7.4 Test Cases

Sno	Description	Username	Password	Confirm Password	Ngo No	Ngo Name	Contact	State	Food Name	Quantity	Expected Error
1	Enter details to create a new account	ABC	abc@123	abc@123	---	-----	--------	----	-------	------	-------
2		------	abcde	abcde	---	-----	---------	----	-------	----	Username Field Cannot be empty
3		Pankaj	pankaj@123	Pankaj@345	---	-----	--------	----	-------	----	Passwords do not match
4	Enter details to register a new NGO	-------	-------	---------	736	XYZ Foundation	9678452380	Delhi	-------	----	------
5		-------	-------	---------	451	-------	9056834230	Delhi	--------	---	Ngo Name Field cannot be empty
6		--------	-------	--------	abc	abc foundation	8526103897	Delhi	-------	---	Ngo No must be a numeric value
7		-------	------	--------	356	AR organization	9856234510586	Delhi	-------	---	Contact field too long
8	Enter Details to register Food	------	------	--------	---	-------	---------	-----	Rice	10Kg	------
9		------	------	--------	---	------	--------	----	-----	5Kg	Food Name cannot be empty

8. Future Scope

- The Project can be used a base for realizing a scheme to be implemented in other projects of same or greater level
- The project itself can be worked on and modified to achieve a complete waste food management system
- A transportation system can be integrated within the system to manage the transport of waste food from one organization to another around the country
- An easier User friendly system can be designed to make it easier to register Ngos, restaurants , leftover food , etc

9. References

- Introduction to Software Engineering by K.K Aggarwal
- En.wikepedia.org
- www.javatpoint.com
- www.geeksforgeeks.org
- Software Engineering a Practitioner's Approach by Roger S. Pressman and Bruce R. Maxim
- P.Jalote , an Integrated Approach to Software Engineering , 3rd Edition
- www.stackoverflow.com
- www.tutorialspoint.com