INDEX

**1)Requirement gathering**

Problem statement

1.2 Functional requirements

Use case diagram

**2)SRS**

**1.Introduction**

1.1 Purpose

1.2 Scope

1.3 Definitions, Acronyms, and abbreviations

1.4 References

1.5 Overview

2.The overall description

2.1 Product Perspective

2.1.1 System Interfaces

2.1.2 Interfaces

2.1.3 Hardware interfaces

2.1.4 Software interfaces

2.1.5 Communications interfaces

2.1.6 Memory constraints

2.1.7 Operations

2.1.8 Site adaptation requirement

2.2 Product functions

2.3 User characteristics

2.4 Constraints

2.5 Assumptions and dependencies

2.6 Apportioning of requirements

Specific Requirements

3.1 External interfaces

3.2 Functions

3.3 Performance requirement

3.4 Logical database requirement

3.5 Design constraints

3.5.1 Standards compliance

3.6 Software system attributes

3.6.1 Reliability

3.6.2 Availability

3.6.3 Security

3.6.4 Maintainability

3.6.5 Portability

3.7Special Requirements

PROBLEM STATEMENT

Building a digital platform to aggregate secondary resources like used food at house-hold level will help to facilitate distribution to needy through such organizations.

There are various campaigns and websites but it is difficult to find legit and trustworthy platform. Thus a trustworthy, user-friendly and universal platform is needed to bridge such organizations and a donor.

The system should be able to take at least 70 queries in peak hours .

Food would be taken from ones who have excess and given to needy with the help of a big pack of volunteers.

FUNCTIONAL REQUIREMENTS:

User Class Administrator:

FR1

Feature: Administrator log in

In order to administer the system, an administrator should be logged into the web-portal.

Scenario: Successful log-in

Given the administrator wants to log in. When theadministrator logs in with an administrator account, then theadministrator should be logged in as an administrator.

FR2

Feature: Verify NGO and Donors

In order to allow a NGO and donors to use the systemAn administrator should be able to verify the NGO and donors.

Scenario: Verify a NGO or Donor

Given the administrator is logged in. When the administratorverifies a beneficiary or donor then the NGO or donor shouldbe able to log in and the beneficiary or donor should benotified by a confirmation email.

Scenario: Reject a NGO or Donor

Given the administrator is logged in. When the administratorrejects a NGO or donor then the NGO or donor should not beable to log in and the NGO or donor should be notified by a

rejection email.

FR3

Feature: Manage donors

In order to keep track of the donors an administrator should beable to manage the donors.

Scenario: Edit an existing donor’s information

Given the administrator is logged in. When theadministrator edits an existing donor then the donorinformation should be updated.

Scenario: Delete/Inactivate an existing donor

Given the administrator is logged in. When theadministrator deletes an existing donor then the donorshould be deleted.

FR 4

Feature: Create an account

In order to create an account a Donor should register on the web-portal.

Scenario: Required information for registration

Given the donor wants to create an account and the donordoes not have an account when the donor registers on theweb-portal byProviding user-name, password, address, e-mail address,

phone number and a valid identification then the donorshould be able to apply for verification.

Scenario: Full information for registration

Given the donor wants to create an account the donor doesnot have an account when the donor registers on the web portalby providing user name, password, address, e-mailaddress, phone number and a valid identification then thedonor should be able to apply for verification.

Scenario: Confirmed registration

Given the donor has applied for verification and hasreceived a confirmation e-mail after registration Then thedonor should be able to log in.

FR 5

Feature: Donor log-in

In order to use the system a donor should be logged in to the web-portal

Scenario: Successful log-in

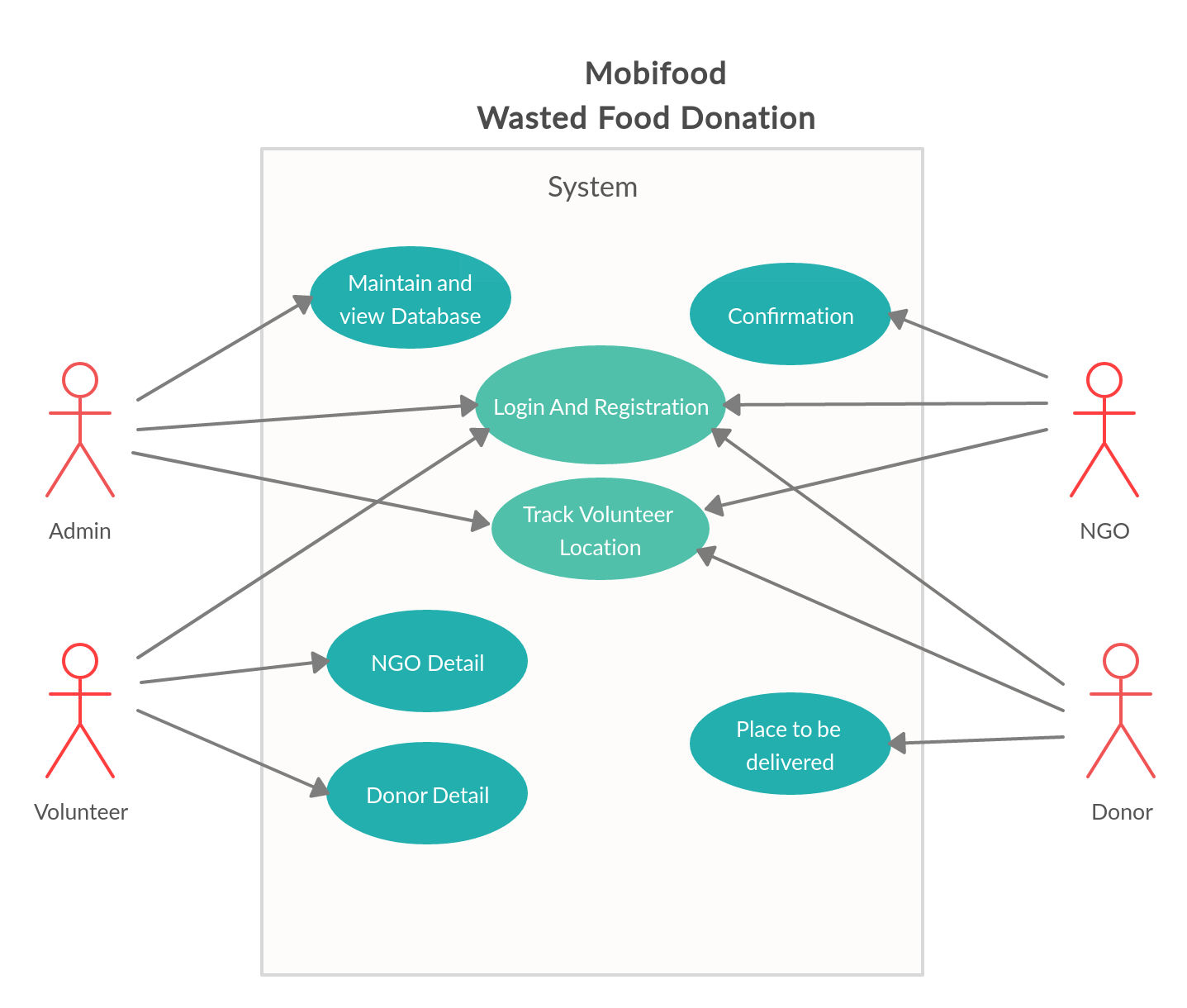
The donor logs in with his/her account.

Scenario: Retrieve password

Given the donor wants to log in and has lost the password.

When the donor enters his/her email address in the“Retrieve password” form and submits the form then thedonor should receive an email.

USE CASE:



CASES

Actors -

The following actors interact and participate in the use case -

Admin

Donor

Volunteer

NGO

Flow of events -

Basic flow

This use case starts when the actor wishes to login to MOBIFOOD.

The system requests that the actor specifies its name, password and user type. The role can be any of Admin, Volunteer , Donor, NGO.

The actor specifies his/her name, password & type.

The system validate the entered name, password and logs the actor into the system.

3,1Alternate Flow

Invalid name / password / role.

If in the basic flow, the actors Enters an invalid name, password or role, the system displays an error message. The actor can choose to eitherreturn to the beginning of the basic flow or cancel the login, at which point the use case ends.

Special Requirements-

None

Pre-Conditions-

All users must have a User Account (i.e user id,Password and Role) created for them in the system(through Admin ) pin to extending the use cases.

Post conditions-

If the use case was successful, the actor is logged into the system. If not, the system state is unchanged.

Extension points-

None

PLACE TO BE DELIVERED-

Brief Description

This use case allows the actor 'Donor' to place food to be donated.

Actor

The following actor interacts in this case 'Donor'

Flow of Events-

basic flow

This use case starts when the Donor wishes to donate food and specifies where food has to be picked up.

Alternate flow.

No change in system is no pick up place specified.

1.4-Special Requirements

None

1.5-Pre Conditions

Donor must be logged onto the system before this use begin

1.6-Post Conditions

If the use case is successful place to be donated is shown on the screen and also received feedback.

1.7-Extension Points

None

2. CONFIRMATION-

2.1 Brief Description

This use case allows the actor 'NGO' to confirm the delivery of food.

2.2 The following actor interact in this use case NGO.

2.3 Flow of events

2.3.1 Basic flow

This use case starts when the NGO wishes to inform it has accepted the delivery.

2.3.2 Alternate flow

The system remains unchanged of delivery not got.

2.4 Special requirements

None

2.5 Pre conditions

NGO must be logged onto the system before this use case begins.

2.6 Post conditions -

If this use case is successful, delivery successful is displayed otherwise system remains unchanged.

2.7 Extension Points-

None

3.MAINTAIN & VIEW DATABASE

Brief Introduction

This use case allows the actor with role 'Admin' to maintain user account. This included adding, changing, deleting user account information from the system.

Actors

The following actor interact in this use case:

Admin

Flow of events

Basic flow

This use case starts when the Admin wishes to add, change and/or delete Use Account information from the system.

The system requests that the admin wishes specify the function/type he would like to perform, ( either Add a User Account', update a user account, orDelete a User Account).

Once the Administration provides the requested info, one of the sub flows is executed.

If the Admin selected "add a user Account" the Add a User Account sub flow is executed.

If the Admin selected "update a user Account" the update a User Account subflow is executed.

If the Admin selected "delete a user Account" the delete a User Account subflow is executed.

3.3.1.1. Add a user

The system requests that Admin enters the user info, this includes

User name

User Id should be unique for each account.

Password

Role

Once the Admin provides the requested info, and Admin verifies the details, the User account information is added to the system and an appropriate message is displayed.

3.3.1.2 Update a User info

The system requests that the administrator enters the user Id.

The Administrator enters the User ID. The system retrieves & display the user account information.

The Admin makes the desired changes to the user account information, This includes any of info specified in add a User account sub flow, the system update the information.

3.3.1.3 Delete a User info

The system requests that the administrator enters the user Id.

The Administrator enters the User ID. The system retrieves & display the user account information.

The system provides the Admin to confirm the deletion of the user account.

The Admin confirms the deletion.

The system deletes the user account record.

Alternate flow

User not Found

If in the update a User Account or delete a User Account sub flows, a user account with the specified user id does not exist, the system displays an error message. The admin can then enter a diff USER ID or cancel theoperation at which point the use case ends.

Update Cancelled

If in update a user sub flow, the Admin decides not to update user account info , the update is cancelled & the basic flow is restarted at the beginning.

2.3.2.3 Delete Cancelled

If in the Delete a user sub flow, the Admin decides not to delete user account info , the delete is cancelled & the basic flow is restarted at the beginning.

Special Requirements

None

Pre - Conditions

The Admin must be logged into the system before this use case begins .

Post Conditions

If the use case was successful, the user account info is added, updated or deleted from the system .otherwisesystem state is unchanged.

Extension Points

None

4 NGO DETAIL-

4.1-Description

This use case allows the actor with role “volunteer”:

4.2-Actor

The following actor interact in the use case:

Volunteer

4.3-Flow of events

4.3.1-Basic flow

This use case starts when the volunteer wants to volunteer to donor request

1)System checks the latest NGO request and gives it the NGO details (Name,Number,Location)and donor details

2)When it gets the details it checks for quality and notifies the admin that he has picked up the delivery

3)It delivers the packet and notifies admin

4.3.2-Alternate flow

If NO request is there at that time ,errormessage will display that will try later has norequest right now.

4.3.4-Special requests

None

4.5-Pre conditions

Volunteer must be logged into the system before they use cases login.

4.6-Post conditions

If the use case is successfully delivery the useful is displayed otherwisesystem remains unchanged .

4.7-Extention points

None

5. DONOR DETAIL

5.1-Description

This use case allows the actor with role “Donor”:

5.2-Actor

The following actor interact in the use case:

Volunteer

Donor

5.3-Flow of events

5.3.1-Basic flow

This use case starts when The donor wishes to donate the food.

1)System checks the latest Donor request and gives the details (Name, Number, Location)to volunteer.

2)When it gets the details it checks for quality and notifies the admin that he has picked up the delivery

3)It delivers the packet and notifies admin.

5.3.2-Alternate flow

If NO request is there at that time error message will display that will try later has no request right now.

5..4-Special requests

None

5.5-Pre conditions

Donor and Volunteer must be logged into the system before they use cases login.

5.6-Post conditions

If the use case is successful, when the volunteer picked up the food otherwise system remains unchanged.

5.7-Extention points

None

6. TRACK VOLUNTEER LOCATION

6.1-Description

This use case allows the actor with role “volunteer”:

6.2-Actor

The following actor interact in the use case:

Admin

Donor

NGO

Volunteer

6.3-Flow of events

6.3.1-Basic flow

This use case starts when the volunteer gives his location.

1)System checks the latest notification of volunteer’s location and gives the location to Admin, NGO and Donor.

2)When they get the details they can track the location of volunteer as well as after completion of delivery volunteer notifies the Admin.

6.3.2-Alternate flow

If thevolunteer will not share his location there at that time error message will display that will try later has no volunteers are there.

6.4-Special requests

None

6.5-Pre conditions

Admin, NGO and Donor must be logged into the system before they use cases login.

6.6-Post conditions

The use case is successful when volunteer successfully delivers the packet to NGO.

6.7-Extention points

None

7.LOGIN AND REGISTRATION

7.1-Description

This use case describes how a user logs in to MOBIFOOD.

7.2-Actor

The following actor interact in the use case:

Admin

Donor

NGO

Volunteer

7.3-Flow of events

7.3.1-Basic flow

This use case starts when the actor wishes to login to MOBIFOOD.

1)The system requests that the actor enter his/her name, password and role. The actor can be any one of Admin, Volunteer, NGO or Donor.

2)The actor enters his/her name, password and role.

3)The System validates the entered name, password and role and logs the actor into the system.

7.3.2-Alternate flow

7.3.2.1 Invalid name/password/role

If in the basic flow, the actor enters an invalid name, password or role. the system displays an error message. The actor can choose to either return to the beginning of the Basic flow or cancel the login, at which point the use case ends.

7..4-Special requirements

None

7.5-Pre conditions

All users must have a user account (user id, password and role) created for them in the system(through the Admin), prior to executing the use cases.

7.6-Post conditions

If the use case was successful, the actor is logged into the system. If not, the system states unchanged .

7.7-Extention points

None

SOFTWARE REQUIREMNET SPECIFICATION:

1 INTRODUCTION

This document aims at defining the overall software requirements for Food Waste Management “Mobifood” .Efforts have been made to define the requirements exhaustively and accurately.The final product will be having only features/functionalities mentioned in this document and assumptions for any additional functionality/feature should not be made by any of the parties involved in developing / testing/implementing / using this product. In case it is required to have some additional features ,a formal change request will need to be raised and subsequently a new release of thus document and/or product will be produced.

PURPOSE

This specification document describes the capabilities that will be provided by the software application Mobifood . It also states the various required constraints by which the system will abide. The intended audience for this document are the development team, testing team and end users of the product.

SCOPE

The software product “Food Waste Management System” ::”MOBIFOOD” will be a platform application for bridging the gap between the ones who have excess food and ones who are needy.It will be a bridge between Donors and needy.

The application will manage the information about various existing donors and NGOs ,and act like a bridge between donors and where delivery has to be done.We will also have a good team of volunteers who would work tirelessly day and night to make the delivery successful and on time.

Printable receipts regarding confirmation of delivery will be generated, also feedback forms will be generated.

The application will greatly simplify the communication and speed up the delivery between the needyand the ones who are donating

DEFINITIONS, ACRONYMS AND ABBREVIATIONS

Following abbreviations have been used throughout this document:

Admin: Administrator

NGO: Non Government Organization

THE DONOR:

An account on the application is created and the details are saved to the database. Along with this donor is verified with the Governmental ID Cards. When donor wants to donate he/she can simply put a notification about his location.

THE VOLUNTEER:

An account on the application is created and the details are saved to the

database. Along with this donor is verified with the Governmental ID Cards .The volunteer is asked about his preference of time and area where he can serve for Mobifood . Whenever a donor puts a notification about donation, nearby volunteer is being notified about the same. And the IDs of both volunteer and donor is being passed to the respective NGO.

THE NGO:

An account on the application is created and the details are saved to the database.The admin registers the NGO’s after checking their authenticity. The NGOs homepage has a section: Donation, which shows the donations made and who made them, who transported them and a confirmation has to be sent by the NGO.

REFERENCES

1)Smart India Hackathon official website

2)IEEE Recommended Practice for Software Requirements Specifications-IEEE Std 830-1993

OVERVIEW

The rest of this SRS document describes the various system requirements ,interfaces, features and functionalities in detail.

THE OVERALL DESCRIPTION

The application will be android based and compatible with all devices fulfilling the basic requirements specified in SRS.

2.1 Product Perspective

This Software is only compatible with android mobile phones.

2.1.1System Interface

None

2.1.2 User Interface

The following screens would be provided :

LOGIN SCREEN:

This will be the first screen to be displayed . It will allow user to access different screens based upon the users role. .

SIGN-UP SCREEN:

This will be accessible by anyone who wants to register himself/herself in order to use this app under the given categories.

DASHBOARD SCREEN:

This screen will be accessible to DONOR , VOLUNTEER and NGO in order to see their past activities such as from when they have been involved in any donation .On this screen ,there will be option for accepting a delivery for VOLUNTEER , for giving Donations there will be option for DONOR and accepting a donation option will be for NGO.

SITE MAINTAINENCE SCREEN:

This screen will be accessible by admin in order to manage site / app.

DONATION SCREEN :

These screens will be to DONOR , VOLUNTEER and NGO. On this screens :

Pick-up Location : Location of the DONOR.

Volunteer Location which is delivering the donation.

NGO location which is accepting the donation .

Type of Food

DONATION STATUS SCREEN :

Donation details will appear on the screens of the given DONOR , VOLUNTEER and NGO involved in the donation if it is successful , else sorry message with the particular reason will appear.

2.1.3 Hardware Interface

1)Android mobile phones.

2)We require a nominal display ; high deficiency screen is not necessary.

3)Also internet connection is required to support login, uploading product image, question’s responses, etc.

4)Ram processing speed: 512mb or more.

5)Android version: Jelly bean(version 4.1) or more

2.1.4 Software Interface

Minimum software requirements for server

1)Operating System : Android

2)Database : MySQL

3)Compiler/Interpreter : PHP

4)Tools : Any browser administration

Minimum software requirements for client

1)An operating system (OS) and a browser.

2.1.5 Communication interfaces

None

2.1.6 Memory Constraints

At least 512mb Ram and 1gb Storage space will be required for running the application.

2.2 Product Functions

This system will consist of two parts: android application and web portal.The mobile application will be used to provide a platform to users (donor orvolunteer) and NGO to interact and view information about them. The webportal will be used for managing the information about the system as a whole.The donors, volunteers and NGOs create their accounts on the application. The admin registers the NGO’s after checking their interest and authenticity. While the donors and volunteers are authenticated through OTP to email and verify his/her ID card. When the user wants to donate any article, they will provide the details of the type of item and their location. The Volunteer checks thesafety and security of the food, confirms the pickup. Thereafter the donation is handed over to the needy NGOs.This is a data-centric product. To store mobile application and web portal will communicate with the database, however in slightly different ways.The mobile application will only use the database to get data while the web portal will also add and modify the database communication will go over the Internet.

2.3 User Characteristics

THE DONOR:

An account on the application is created and the details are saved to thedatabase. Along with this donor isCards. The donor is then asked their preference as what they would like to

donate: Food ,Books, Clothes. A donor can select any of them. When donorwants to donate he/she can simply put a notification about his location anditem he/she wants to donate.

THE VOLUNTEER:

An account on the application is created and the details are saved to thedatabase. Along with this donor is verified with the Governmental ID Cards. Thevolunteer is asked about his preference of time and area where he can serve forMobifood. Whenever a donor puts a notification about donation, nearby volunteeris being notified about the same. And the IDs of both volunteer and donor isbeing passed to the respective NGO.

THE NGO:

An account on the application is created and the details are saved to thedatabase. The admin registers the NGO’s after checking their authenticity.The NGOs also have to choose the kind of item they require from donation.ANGO homepage has a section: Donation, which shows the donations made andwho made them, who transported them and a confirmation has to be sent bythe NGO.

2.4 Constraints

No user would be able to create an account without verification of theiridentities. The donor needs to put the product details like the quality, quantity.Volunteer before picking up the donation from the donor, will confirm thegoodness of product.

2.5 Assumptions and Dependency

The basic login and sign up has been setup and we have a well-built database of users (donor and volunteer) and NGO. The web portal and the application are connected to each other through the database. Wehave collaboration with NITI Aayogthroughhttp://niti.gov.in/content/ngodarpanto check the authenticity of any NGO.We have collaboration with thegovernment through https://eaadhaar.uidai.gov.in to check the authenticity of any aadhaar card.

2.6 Apportioning of Requirement

Not Required

3.0SPECIFIC REQUIREMENTS

This section contains the software requirements to a level of detail sufficient to enable designers to design MOBIFOOD.

3.1 EXTERNAL INTERFACE REQUIREMENTS

3.1.1 USER INTERFACES

The following screens would be provided :

LOGIN SCREEN:

This will be the first screen to be displayed . It will allow user to access different screens based upon the users role. The roles are DONOR , VOLUNTEER and NGO. Various fields that are available on the screen will be :

Type : One of the following options to be selected :DONOR , VOLUNTEER and NGO.

User id : Gmail id/username.

Password : Alphanumeric character upto 8 characters.

SIGN-UP SCREEN:

This will be accessible by anyone who wants to register himself/herself in order to use this app under the given categories.

DASHBOARD SCREEN:

This screen will be accessible to DONOR , VOLUNTEER and NGO in order to see their past activities such as from when they have been involved in any donation .On this screen ,there will be option for accepting a delivery for VOLUNTEER , for giving Donations there will be option for DONOR and accepting a donation option will be for NGO.

SITE MAINTAINENCE SCREEN:

This screen will be accessible by admin in order to manage site / app. Admins can perform following options :

Add / Delete /update a functionality.

Add/Update/Delete a users account due to unavoidable reasons.

DONATION SCREEN :

These screens will be to DONOR , VOLUNTEER and NGO. On this screens :

Pick-up Location : Location of the DONOR.

Volunteer Location which is delivering the donation.

NGO location which is accepting the donation .

Type of Food

DONATION STATUS SCREEN :

Donation details will appear on the screens of the given DONOR , VOLUNTEER and NGO involved in the donation if it is successful , else sorry message with the particular reason will appear.

3.1.2 HARDWARE INTERFACES

Same as in 2.1.3

3.1.3 SOFTWARE INTERFACES

Same as in 2.1.4

3.1.4 COMMUNICATIONs INTERFACES

None.

3.2 SYSTEM FEATURES

SIGN-UP MAINTAINANCE:

There are 3 types of user : DONOR ,VOLUNTEER and NGO. the role of the user would be decided during the sign-up process by which it can access the screens which are available for him.

Validity checks:

-Only admins will be authorized to access the user accounts information.

-Credentials i.e. email,password can not be blank.

-if user is volunteer or donor then an identity proof is required i.e. AADHAAR CARD.

-if user is NGO then name by which they have registered themselves in the government records is required.

Sequencing Information:

In order to access the app , user have to sign in . After sign-in , user can now use its feature .

Error Handling/response to abnormal conditions:

If any of the above validations / sequencing flow does not hold true , appropriate error messages will be provided . if the user wants to retrieve the password the donor enters his/her email address in the “Retrieve password” form and submits the form then the donor should receive an email.

DONATION MAINTAINENCE:

This is the key functionality of app , so in order to use the functionalities it provides these following process.

Validity Checks:

-Atleast one NGO should be available during the donation.

-Location of the donor cannot be left blank.

-The details of the volunteer that accept to deliver should share is location.

-Type of food must be specified by the donor .

Sequencing information:

This can be done after the login procedures.When the food is received and verified by the NGOthen the given process is successful.

Error Handling/response to abnormal conditions:

If anyone of the above validation is not successful then appropriate message will be prompted to the user for doing the needful.

3.3 Performance Requirement

None

3.4 Logical Database Requirement

Volunteer Info-Name, Age, Sex, Time of Availability

NGO Info: Name, Address

Donor Info: Name, Address

3.5 DESIGN CONSTRAINTS

Operating system and hardware constraints:

This management software is only compatible with Android Mobile Phones.

Minimum RAM size 2GB.

Hard disk size must be more than 32GB for storing records.

3.6 Software System Attributes

3.6.1 Reliability

Every user who will be accessing the software will be given a Unique username and password which they have to change after every 45 days. Repeated entry of false username or password will lead to software halt state and all the records will be locked, and software can only be revived by admin

3.6.2 Availability

Food quality will be done by the Volunteers and the receiving NGOs.

3.6.3 Security

Record encryption: Records are encrypted with Algorithm which splits records into blocks and encrypts them individually.

Records of every day activity will be maintained which can be viewed only by the admin.

Access controls: An user cannot access the data which is not relevant. Only relevant records will be visible to the user.

3.6.4 Maintainability

Admin maintains the record of the food donated.

3.6.5 Portability

1.MySQL used for maintaining Database.

2.PHP used as compiler/interpreter.

3.Runs on Android Operating System.

OTHER REQUIREMENTS

None.

Appendix

ANALYSIS OF GOOGLE FORMS (VOLUNTEER)

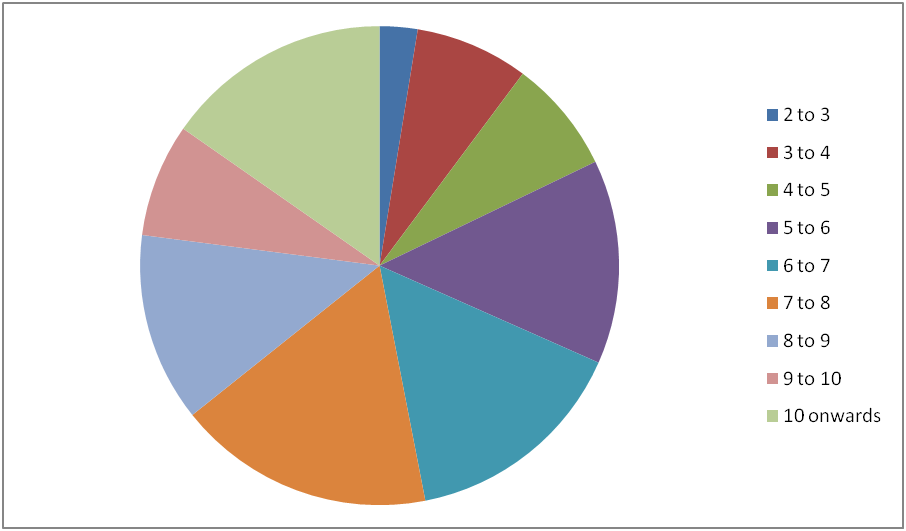
"If you cannot feed a hundred people , the feed just one ".

With this in mind our volunteers should great enthusiasm in filling up the form. Some people even volunteered for the festive season as well.

Analyzing the data , it was seen that the time when most volunteers were free was coinciding with when most of the donations were being made and before any of the child homes and orphanages are closed . Canteens were available to make donation after 5 pm in the evening and volunteers are available at that time. Most of the canteens we went to were in the North campus and students/volunteers also resides in Malkaganj , from where it is easy for them to take donations . Also, as these are student canteens , the quality of food is quite good .

Weddings usually end by 10 to 11 pm in the night . Leftover food from these can also be collected by our volunteers as there are many volunteers which are available after 10 pm . as they are free from their classes and other chores .

Also 2-4 volunteers also had experience of working in other NGO's .



Timing is in PM.