**MSIS 2602 – Information Systems Analysis and Design**

****

**Food Donation Management System**

Jayalakahsmi Vaidyanathan

Ritika Mathur

Sarah Dihan Fernandes

Varuna Paul Jeeva

TABLE OF CONTENTS

1. Introduction 3
2. Business Description 3
3. Gantt Chart
4. Business Need
5. Business Requirements
6. Business Value
7. Issues / Constraints
8. Functional Requirements
   1. Process oriented
   2. Information oriented
9. Non-functional Requirements
   1. Operational
   2. Performance
   3. Security
   4. Cultural and Political
10. Use Cases
11. Data Flow Diagrams
    1. Context Diagram
    2. Level 0 DFD
    3. Level 1 DFDs
12. Data Dictionary
13. Program Structure Chart
14. User Interface
15. Future Scope
16. Conclusion

**1. Introduction**

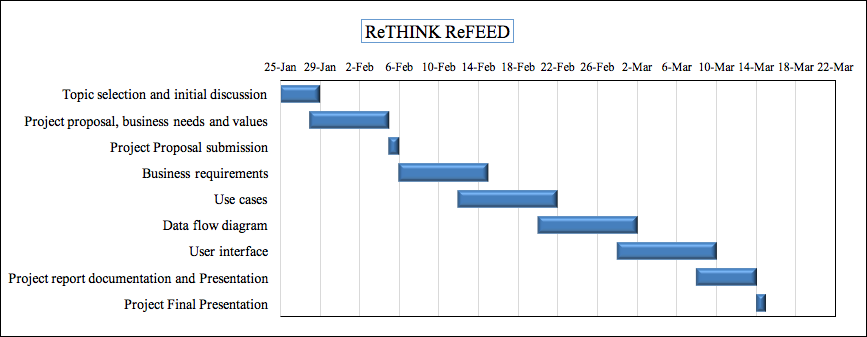
Food waste is one of the biggest problems in the world today. According to the U.N. Food and Agriculture Organization (FAO), at least 1.3 billion tons of food is lost or wasted every year globally—in fields, in storage, at restaurants, and in markets. The United States wastes about 62.5 million tons and spends $218 billion a year in “growing, processing, transporting, and disposal of food that is never eaten.” On the other hand, 1 in 7 people are hungry in the US. According to the United Nations, if we recover all the food that is lost or wasted, we will have enough to feed all those who are hungry, four times over.

**2. Business Description**

‘ReTHINK, ReFEED’ is a web-based ‘marketplace’ that will connect the organizations and people who have excess food with nonprofits, NGOs, charities and others who need good quality food. The Food Donors such as restaurants, cafeterias, supermarkets, farms, etc. will be able to sign up with their details including their location. Aid group organizations like charities, food shelters and other nonprofits that have been carefully vetted for authenticity will also sign up with their details such as location and food preferences. Whenever the Donors have excess food, they can post the details online and the system will send alerts to all the charities and nonprofits in the system based on their location and food preferences. These charities will then be able to book the food. Once the match is made, the food rescue volunteers who work with the organization will recover the surplus food and deliver it to the charities in need. The organization will also accept financial donations to fund the various activities.

**3. Gantt chart**

**Gantt chart showing the proposed work plan**



**4. Business Need**

This application has been developed to reduce food waste and at the same time feed the hungry by diverting excess food to qualified NGOs, charities, nonprofits and food shelters. It will conserve the resources and improve the lives of thousands of people by providing an efficient and effortless way for both the Donors and the Charities to identify each other with just one click.

**5.Business Requirements**

The proposed Information System will support the following operations:

1. Account creation and management for all users including food donors (restaurants, supermarkets, farms, etc.), charities and other nonprofits and volunteers of the organization.

2. Networking services: The System will help to connect the charities and nonprofits with Food Donors based on food preferences and location.

3. Real time updates and alerts about availability of food.

4. Food Recovery and Delivery: Once the volunteer accepts a delivery, the application will provide all the necessary details including the address and navigation details.

5. Financial donation portal: The System will help the organization to receive and manage financial donations and validate them through a third-party gateway.

6. Reporting system: The System will generate monthly/weekly reports of the donations (both food and money) received and distributed.

7. Personnel management system: The System will help the organization manage its employees and volunteers.

8. User Feedback: After completion of delivery, the users can rate the organization and the volunteers and also give their feedback.

**6.Business Values**

We expect that the proposed system will create many tangible and intangible values for all the parties involved:

1. Reduces food wastage and ensures that all of the labor and natural resources that go into growing, processing, and marketing food doesn’t go to waste.

2. Serves the hungry and needy with good quality food.

3. Helps the charities save money. This will allow them to reallocate their funds to other needs such as clothing, education, accommodation, etc.

4. Enhances inventory control for Donors: Donations can reduce overstocking problems and also help to move unsalables from grocery stores, supermarkets, farms, etc.

5. Improves the morale of the Donors and also provides tax benefits.

6. Helps the environment by reducing the amount of food that ends up in landfills.

**7. Issues / Constraints**

* There should be enough food donations every month to sustain the operations.
* There should be sufficient volunteers who are available to pick up food in the particular area where food donation is available.

**8. Functional Requirements**

**8.1. Process oriented:**

**(i)Account creation**

* The system will enable users (Food donors, Charities / NGOs and Volunteers) to create an account.
* The system will collect and store user details.
* The system will create and activate user account after validation.
* If the user is a Charity / NGO, the system will send a notice to the manager to verify it’s authenticity.

**(ii)Update user details**

* The system should allow the user to update/change existing details.
* The system should allow the user to close the account as and when required.

**(iii)Networking services**

* The system should allow the donor to log in and donate food after providing all food details.
* The system will match donors with NGOs/charities based on food specifications and distance of locations.
* The system will send alerts to all charities matched.
* The system will allocate food to a charity/NGO on first come first serve basis.

**(iv)Real time updates and alerts about availability of food**

* The system will allow NGOs / Charities to claim food.
* The system sends an alert to all the other NGOs/charities informing them the food has been claimed now.
* The system sends alerts to all volunteers about the delivery of food from the donor to the NGO/charity.

**(v)Food recovery and delivery**

* The system should send alert message to nearby volunteers.
* The system should provide donor name, charity name and location of pickup and drop.
* Once the volunteer accepts the delivery, the system should record this transaction and send a confirmation message to the volunteer.
* The system should send details of volunteer, pickup time, estimated delivery time to the donor and NGO/Charity.

**(vi)Financial donation**

* The system will collect donation amount details from the Financial donor.
* The system will collect payment information from the donor.
* The system will validate the payment details via a third-party payment clearing house.
* The system will confirm the payment and send a receipt to the donor.

**(vii)Report generation**

* System must accept the duration as input from user and generate report.
* System must have the capabilities to generate weekly and monthly reports for both financial donation and food distribution.
* System must be able to generate reports in different formats (PDF, CSV, RTF, and HTML).
* System must email the reports to the email list if specified.

**(viii)Personnel management system**

* The system should maintain employee details.
* The system should track employee working hours.
* The system should maintain payroll details.

**(ix)User feedback system**

* System must send the feedback form to the NGO/ Charity center once food is delivered to them.
* System must store the user feedback into database once user submits the feedback form.

**8.2. Information oriented:**

* The system should store all food donor details (including name, contact details, address, etc.) in the Food Donor Datastore.
* The system should store all Charity / NGO details (including name, contact details, address, size of the charity, food preferences, etc.) in the Charity / NGO Datastore.
* The system should store all volunteer details (including name, contact details, address, vehicle size, pickup preferences) in the Volunteer Datastore.
* The system will generate unique Transaction ID for every food donation and store the donor details and the recipient charity/NGO details for each donation.
* The system will store the donated food details for every transaction.
* The system should store details of all financial donation transactions in the Financial Donation Datastore.
* The system should retain financial donation records for the current year and past 10 years.
* System must retain all the food transaction history for the current year and past 10 years.
* System must retain feedback from NGO/Charity for 10 years.

**9. Non - Functional Requirements**

**9.1.Operational:**

* The system should be accessible from wide range of devices.
* The system should work on different Web Browsers.
* The system should keep record of every transaction made.
* The system should do real time matching of volunteer, donor and NGO/Charity locations.

**9.2.Performance:**

* Any interaction between users and the system must not exceed 2 seconds.
* The system downloads any status parameter within 5 minutes of change.
* Any interaction between the user, volunteer, employee and the system must be real time.
* System must allow multiple users from different locations to access the application simultaneously.
* System must be available for 24 hours per day, 365 days per year.
* System must support at least 1000 users at a time.

**9.3.Security:**

* Only the manager will have access to all the Financial donor’s information.
* The donors can see all details of the NGOs/Charities and the NGOs/charities can see all details of the donors.
* The system will not allow any changes by anyone except manager in Datastore records of donations made by financial and food donors.
* The system will have all safeguards from virus, trojan etc.

**9.4.Cultural and Political:**

* The system should allow only verified charities to accept the donated food.
* The system should protect the personal information of all users in compliance with the Data Protection Act.
* The system should ensure that the food donors are in compliance with applicable Food Safety Regulations and company enforced quality standards.

**10. Use Cases**

**Use Case 1: Account Creation for Food Donors**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case Name:** Account Creation for Food Donors | | | **ID:** UC-1 | | **Priority:** High |
| **Brief Description:** This use case describes how the system enables Food Donors to create an account. | | | | | |
| **Actor:** Food Donor | | | | | |
| **Trigger:** Food Donor wants to create an account so that he can donate food  **Type**  **🞎 External** 🞎 Temporal | | | | | |
| **Preconditions:** | | | | | |
| **Normal Course**   1. Donor wants to create an account    1. The Donor provides his details    2. System generates a unique User ID    3. System sends a validation link to the Donor for account activation    4. Donor validates the account    5. System activates account | | | **Information for Steps**  Name, Email ID, Phone number, Address  User ID  User Email ID  User validation  Validated User Information | | |
| **Alternative Course(s):** | | | | | |
| **Post conditions:**  1. The Food Donor datastore is updated. | | | | | |
| **Exceptions:** | | | | | |
| **Summary:**  **Inputs Source Outputs Destination** | | | | | |
| Name, Email ID, Phone number, Address  User Email ID  User validation  Validated User Information | User  User  User | New User Information  Validated Account  New Donor Information | | Food Donor datastore  Food Donor datastore | |

**Use Case 2: Account Creation for Charities / NGOs**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case Name:** Account Creation for Charities / NGOs | | | **ID:** UC-2 | | **Priority:** High |
| **Brief Description:** This use case describes how the system enables Charities / NGOs to create an account. | | | | | |
| **Actor:** Charity / NGO | | | | | |
| **Trigger:** Charity / NGO wants to create an account so that they can accept the donated food  **Type**  **🞎 External** 🞎 Temporal | | | | | |
| **Preconditions:** | | | | | |
| **Normal Course**   1. Charity / NGO wants to create an account    1. The Charity provides details    2. System generates a unique User ID    3. The System asks the Charity to provide size of the charity and food preferences    4. System sends a validation link to the Charity / NGO for account activation    5. Charity validates the account    6. System activates account    7. System sends notification to the Manager to check the authenticity of the Charity | | | **Information for Steps**  Name, Email ID, Phone number, Address  User ID  Charity size and Food preferences  User Email ID    User validation  Validated User Information  New Charity Information | | |
| **Alternative Course(s):** | | | | | |
| **Post conditions:**  1. The Charity / NGO datastore is updated | | | | | |
| **Exceptions:** | | | | | |
| **Summary:**  **Inputs Source Outputs Destination** | | | | | |
| Name, Email ID, Phone number, Address  Charity size and Food preferences  User Email ID  User validation  Validated User Information  New Charity Information | Charity  Charity  Charity  Charity | Charity details  Charity Food details  Validated Account  Verified Account | | Charity / NGO datastore  Charity / NGO datastore  Manager  Charity / NGO Datastore | |

**Use Case 3: Account Creation for Volunteers**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case Name:** Account Creation for Volunteers | | | **ID:** UC-3 | | **Priority:** High |
| **Brief Description:** This use case describes how the system enables Volunteers to create an account. | | | | | |
| **Actor:** Volunteer | | | | | |
| **Trigger:** A volunteer wants to create an account so that he can deliver food.  **Type**  **🞎 External** 🞎 Temporal | | | | | |
| **Preconditions:** | | | | | |
| **Normal Course**   * 1. Volunteer wants to create an account   2. The Volunteer provides details   3. System generates a unique User ID   4. The System asks the Volunteer to specify the size of the delivery vehicle and Volunteer’s preferred timings   5. System sends a validation link to the Volunteer for   account activation   * 1. Volunteer validates account   2. System activates account | | | **Information for Steps**  Name, Email ID, Phone number, Address  User ID  Vehicle size and Availability  User Email ID  User validation  Validated User Information | | |
| **Alternative Course(s):** | | | | | |
| **Post conditions:**  1. Volunteer Datastore is updated. | | | | | |
| **Exceptions:** | | | | | |
| **Summary:**  **Inputs Source Outputs Destination** | | | | | |
| Name, Email ID, Phone number, Address  Vehicle size and Volunteer Availability  User Email ID  User validation  Validated User Information | Volunteer  Volunteer  Volunteer  Volunteer | Volunteer details  Vehicle detail and Volunteer Availabilit  Validated Account  New Volunteer Information | | Volunteer Datastore  Volunteer Datastore  Volunteer Datastore | |

**Use case 4: Validate/update**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:** Validate/update | | | **ID:** UC-4 | | | **Priority:** High |
| **Brief Description:** This describes how system will help manage its user updates. | | | | | | |
| **Actor:**  Volunteer or donor or NGO/Charity | | | | | | |
| **Trigger:** whenUser returns interacts to log in or update information.  **Type 🞎 External 🞎 Temporal** | | | | | | |
| **Preconditions:**  1.Application is online.  2.Volunteer database or donor database or NGO/Charity database is online. | | | | | | |
| **Normal Course**   1. User provides username and password to log in 2. System validates the user provided password with the existing password saved on the record. 3. User will provide new information if they want to update existing profile. | | | | **Information for Steps**  Username, Password  User Credentials Account  User information(Contact details, Email address) | | |
| **Alternative Course(s):**   * 1. If the username and password doesn’t match.   2. Password recovery options via e-mail/text method. | | | | | | |
| **Post conditions:**  1. Volunteer datastore or Donor datastore or NGO/Charity datastore is updated. | | | | | | |
| **Summary:**  **Inputs Source Outputs Destination** | | | | | | |
| Username  Password  Information to update | User  User  User | Updated user information | | | Volunteer  Datastore or Donor Datastore or NGO/Charity Datastore | |

**Use Case 5: Networking services**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:** Networking services | | | **ID:** UC-5 | | | **Priority:** High |
| **Brief Description:** The System will help to connect the charities and nonprofits with Food Donors based on food preferences and location. | | | | | | |
| **Actor:** Donor | | | | | | |
| **Trigger:** Food Donor clicks on donate  **Type 🞎 External 🞎 Temporal** | | | | | | |
| **Preconditions:**   1. Food Donor Datastore is updated. 2. Charity/NGO Datastore is updated. 3. Food Donor is authenticated. | | | | | | |
| **Normal Course**   1. Food Donor logs in and clicks on donate. 2. System fetches all details for the donor. 3. System generates Transaction ID for the donation. 4. Food Donor fills details of available food. 5. System matches registered NGOs & charities with similar food requirements located within 10 miles of the donor. 6. Matched charities & NGOs get real time alerts by the system. 7. Charities & NGOs accept food. 8. System accepts request on first come first serve basis. | | | | **Information for Steps**  Donor ID, Food details  Donor details  Transaction ID,Date,Food Donor details, Food details  Food details  Charity/NGO details, Food preferences  Donor details, Food details  Charity/NGO details  Transaction ID, Charity/NGO details,  Acceptance Time, Acceptance status. | | |
| **Alternative Course(s):** | | | | | | |
| **Post conditions:**   1. Food Transaction Datastore is updated. | | | | | | |
| **Exceptions:** | | | | | | |
| **Summary:**  **Inputs Source Outputs Destination** | | | | | | |
| Donor ID, Food details  Donor details  Food details  Charity/NGO details, Food preferences  Charity/NGO details | Food Donor  Food Donor Datastore  Food Donor  Charity/NGO Datastore  NGO/Charity | Transaction ID, Donor details, Food details  Donor details, Food details  Transaction ID, Charity/NGO details,  Acceptance Time, Acceptance status. | | | Food Transaction Datastore  NGOs/Charities  Food Transaction Datastore | |

**Use Case 6: Real time updates**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:** Real time updates and alerts about availability of food. | | | **ID:** UC-6 | | | **Priority:** High |
| **Brief Description:** When food up for donation is accepted by an NGO or charity, the donor, other NGOs/charities and volunteers receive relevant updates by the system. | | | | | | |
| **Actor:** | | | | | | |
| **Trigger:** Acceptance status flag changes to “Accepted”.  **Type 🞎 External 🞎 Temporal** | | | | | | |
| **Preconditions:**   1. Food Transaction datastore is updated. 2. Volunteer datastore is updated. | | | | | | |
| **Normal Course**   1. System fetches Acceptance status and Transaction ID 2. Food Donors are notified their food has been claimed. 3. Food donors receive Information about charity/NGO that accepted their donation. 4. System selects all matched NGOs/Charities 5. Other NGOs/charities will be notified the food has been claimed. 6. System finds relevant registered volunteers 7. Volunteers get alerts about new delivery request. | | | | **Information for Steps**  Acceptance Status, Transaction ID    Food Donor ID, Recipient Charity/ NGO name, Transaction ID  Recipient NGO/Charity name  Food Donor name, Recipient Charity/NGOs details, Transaction ID.  Food Donor name, Recipient Charity/NGO name  Volunteer IDs, Food Donor name, Recipient NGO/Charity ID, Recipient NGO/Charity name, Transaction ID  Food Donor name, Recipient NGOs/Charity details | | |
| **Alternative Course(s):** | | | | | | |
| **Post conditions:** | | | | | | |
| **Exceptions:** | | | | | | |
| **Summary:**  **Inputs Source Outputs Destination** | | | | | | |
| Acceptance Status, Transaction ID  Food Donor ID, Recipient Charity/ NGO name, Transaction ID  Food Donor name, Recipient Charity/NGOs details, Transaction ID.  Volunteer IDs, Donor name, Recipient NGO/Charity ID, Recipient NGO/Charity name, Transaction ID | Food Transaction Datastore.  Food Transaction Datastore.  Food Transaction Datastore.  Employee/Volunteer Datastore.  Food Transaction Datastore | Recipient NGO/Charity name  Food Donor name, Recipient Charity/NGO name  Food Donor name, Recipient NGOs/Charity details | | | Food Donor  Other Charities  Volunteer | |

**Use case 7: Food Delivery**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:** Food Recovery and Delivery | | | **ID:** UC-7 | | | **Priority:** High |
| **Brief Description:** Volunteer accepts a delivery, the application will provide all necessary details. | | | | | | |
| **Actor:** Volunteer | | | | | | |
| **Trigger:** Volunteer gets an alert message on the phone app  **Type 🞎 External 🞎 Temporal** | | | | | | |
| **Preconditions:**  1.Volunteer active.  2.Volunteer receives an alert. | | | | | | |
| **Normal Course**   1. Volunteer receives food delivery alert on the mobile app. 2. Volunteer clicks on the alert. The app displays the pick up and drop location and navigation details. 3. Volunteer accepts the request sets the time for pick up. 4. System sends confirmation to the volunteer. 5. System sends confirmation to the food donor and recipient with volunteer details and pick up and estimated delivery time. 6. Once the food is delivered, Volunteer clicks on delivery completed | | | | **Information for Steps**  Volunteer ID ,Alert message  Donor Name, NGO/Charity name, Donor location, NGO/Charity location, Preferred pick up time.  Volunteer ID, Donor ID, NGO/Charity ID, Pickup time.  Confirmation ID, Donor name, NGO/Charity name, Donor location, NGO/Charity location, Pickup time.  Volunteer name, Pickup time, Estimated delivery time.  Update delivery complete status | | |
| **Alternative Course(s):** | | | | | | |
| **Post conditions:**   1. Delivery datastore is updated. 2. Volunteer and Donor get details of pickup and delivery. | | | | | | |
| **Exceptions:**  E1: Volunteer doesn’t accept the delivery alert.  1.System will resent the alert message to other nearby volunteers. | | | | | | |
| **Summary:**  **Inputs Source Outputs Destination** | | | | | | |
| pick up time | Volunteer | Volunteer ID, Alert message  Volunteer ID, Donor ID, NGO/Charity ID, Pickup time.  Volunteer name, Pickup time, Estimated delivery time.  Confirmation ID, Donor name, NGO/Charity name, Donor location, NGO/Charity location, Pickup time | | | Food Transaction Data  Store  Delivery datastore  Donor and NGO/Charity  Volunteer | |

**Use Case 8: Financial Donation**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Use Case Name:** Financial Donation | | | **ID:** UC-8 | | **Priority:** High |
| **Brief Description:** This use case describes how the Financial Donor donates money to the organization. | | | | | |
| **Actor:** Financial Donor | | | | | |
| **Trigger:** Financial Donor wants to donate money  **Type**  **🞎 External** 🞎 Temporal | | | | | |
| **Preconditions:**  1. Third party payment gateway is set up. | | | | | |
| **Normal Course**   1. Donor wants to donate money    1. Donor clicks on ‘Donate’   1.2 The System requests the Donor to specify the  amount he wants to donate  1.3 System collects Donor’s Payment information   * 1. Donor confirms Payment transaction   2. System obtains Payment Authorization from   Payment clearing house   * 1. System processes donation   2. System sends confirmation and receipt to Donor | | | **Information for Steps**  Donor Name, Contact details  Donation amount  Payment details, Billing address  Payment confirmation  Payment Authorization    Payment details  Payment Receipt | | |
| **Alternative Course(s):** | | | | | |
| **Post conditions:**  1. The Financial Donation datastore is updated. | | | | | |
| **Exceptions:**  E1: Payment is not authorized (occurs at step 1.5)   1. System displays message that payment is not accepted 2. System asks Donor to enter new payment information or exit 3. System terminates use case if Donor specifies exit; otherwise returns to Normal Course (step 1.5)   E2: Donor cancels payment (occurs at step 1.4)   1. Donor cancels payment transaction 2. System terminates use case | | | | | |
| **Summary:**  **Inputs Source Outputs Destination** | | | | | |
| Donor Name, Contact details  Donation amount  Payment details, Billing address  Payment confirmation  Payment Authorization  Payment Acceptance | Donor  Donor  Donor  Donor  Payment clearing house | Donor details  Donation details  Payment details  Payment Receipt | | Financial Donation datastore  Financial Donation datastore  Financial Donation datastore  Donor | |

**Use Case 9: Reporting System**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:** Reporting System | | | **ID: 9** | | **Priority: High** | |
| **Brief Description:**  The system generates report about the quantity of food distributed or amount of financial donations collected weekly/monthly. Weekly report – end of a week, Monthly Report – End of month | | | | | | |
| **Actor: Manager** | | | | | | |
| **Trigger:** Manager wants to generate a report.  **Type ◻ External ◻ Temporal** | | | | | | |
| **Preconditions:**   1. The Manager is authenticated to access the system 2. The food distribution and financial donation details are available up to date in the system 3. The report type includes: PDF, CSV, RTF, and HTML. | | | | | | |
| **Normal Course**  **1.0 Manager accesses the Reporting from the application**  1. System presents report type list – Monthly/ Weekly  2. Manager selects the report duration (Weekly/Monthly)  3. Upon success, system prompts the user to select a file type to render the report in  4. Manager selects a rendering type  5. System generates report as defined by the selected report type  5.1 System displays the report if display the result is selected  5.2 System emails the report to the distribution list if specified | | | | **Information for Steps**  Report type list  Chosen Duration  File type  Chosen File type  Generated report  Report Displayed  Email | | |
| **Alternative Course(s):** | | | | | | |
| **Post conditions:**   * + - 1. Reports are generated which can be published on the website | | | | | | |
| **Exceptions:** | | | | | | |
| **Summary:**  **Inputs Source Outputs Destination** | | | | | | |
| Report type list  Chosen Duration  File type  Chosen File type  Generated report  Report Displayed  Email | Manager  Manager  Financial Donation, Food transaction Datastores | Report type list  Report generation  File type  Report generation  Report  Online Report  Report Attachment | | | | Manager  Manager  Manager/email recipient  Manager  Manager/Email Distribution List |

**Use case 10: Personnel Management**

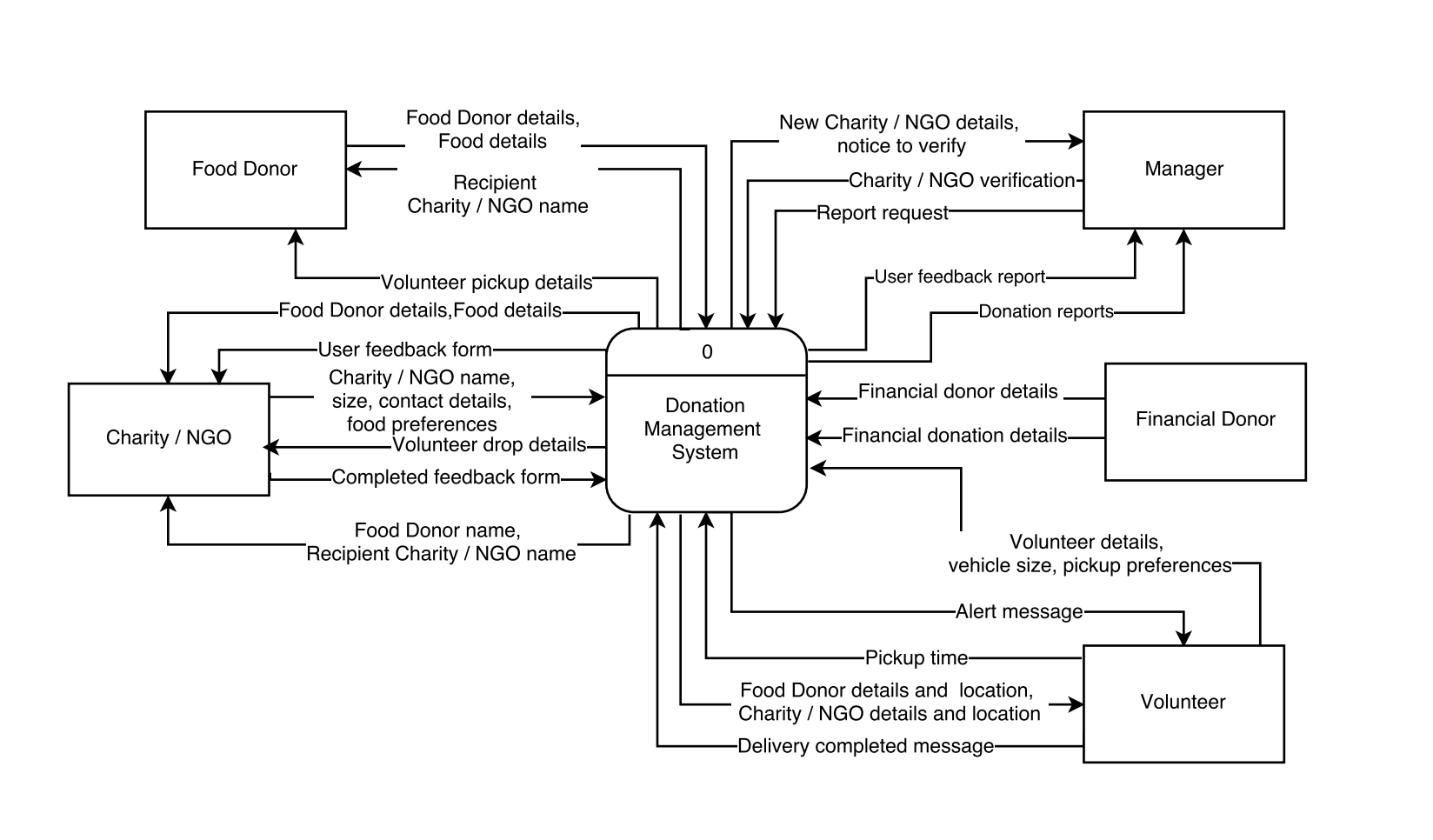
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:** Personnel Management system | | | **ID:** UC-10 | | | **Priority:** High |
| **Brief Description:** This describes how system will help the organization manage its employees | | | | | | |
| **Actor:**  Manager | | | | | | |
| **Trigger:** Manager updates employee details into the system.  **Type 🞎 External 🞎 Temporal** | | | | | | |
| **Preconditions:**  1.Manager is authenticated.  2.Employee Database is available. | | | | | | |
| **Normal Course**   1. Manager enters employee details 2. Manager accepts the timesheet submitted by employee. 3. Manager enters employee pay grade. 4. The system calculates the salary for the employee. 5. The system issues pay check to the employee. 6. The system generates salary record. | | | | **Information for Steps**  Employee name, Contact info, Work details.  Timesheet details.  Pay grade.  Timesheet details, Pay grade.  Pay check details.  Salary record updates. | | |
| **Alternative Course(s):** | | | | | | |
| **Post conditions:**  1. Employee Database is updated. | | | | | | |
| **Summary:**  **Inputs Source Outputs Destination** | | | | | | |
| Employee name, Contact info, Work details.  Timesheet details.  Pay grade  Timesheet details, Pay grade | Manager  Employee  Manager  Employee datastore | Pay check details.  Salary record updates. | | | Employee  Employee datastore | |

**Use Case 11: Feedback system**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Use Case Name:** Feedback system | | | **ID:** UC-11 | | **Priority: High** | |
| **Brief Description:**  The system requests and receives feedback from the NGO/ Charity center regarding the food service | | | | | | |
| **Actor:** NGO/ Charity Center | | | | | | |
| **Trigger:** Food has been delivered and delivery\_status has changed to “Delivered” status in Delivery Datastore  **Type ◻ External ◻ Temporal** | | | | | | |
| **Preconditions:**  1. The food is delivered and recorded in the database  2. The details of the NGO/ Charity center is already available in the database  3. Automatic feedback link sent to the NGO/ Charity Centers upon delivery status change | | | | | | |
| **Normal Course**  **1.0** The food is delivered and recorded in the database  1. Feedback form as a link is sent to the NGO/ Charity center  **2.** NGO/ Charity Centre fills in feedback form and submits, which is stored in database  3. Manager views the feedback report from database | | | | **Information for Steps**  User Feedback Form  Filled feedback form  User Feedback report | | |
| **Alternative Course(s):** | | | | | | |
| **Post conditions:** | | | | | | |
| **Exceptions:** | | | | | | |
| **Summary:**  **Inputs Source Outputs Destination** | | | | | | |
| User Feedback Link  Filled feedback form  User Feedback report | Delivery\_status in Delivery datastore  NGO/ Charity Centre  User Feedback Details | Feedback Link  Completed Feedback form  Feedback Report | | | | NGO/ Charity Centre  User Feedback Datastore  Manager |

**11. Data Flow Diagrams**

**11.1.Context Diagram**

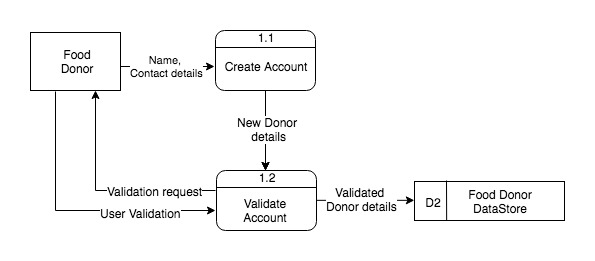


**11.2. Level 0 DFD**

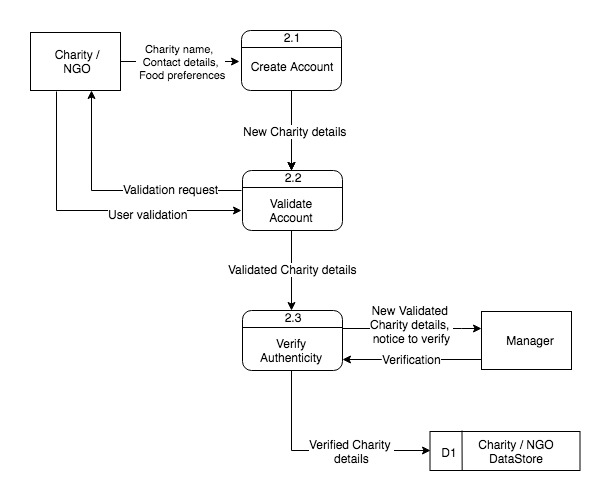


**11.3.Level 1 DFD**

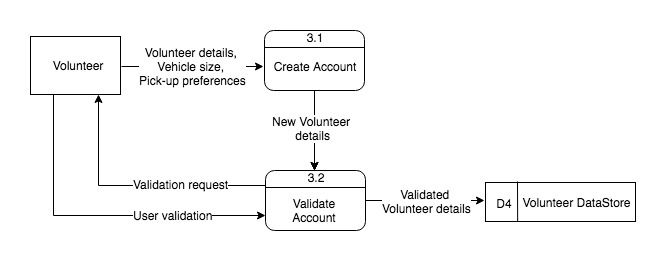
**Process 1: Create and Validate account – Food Donor**

****

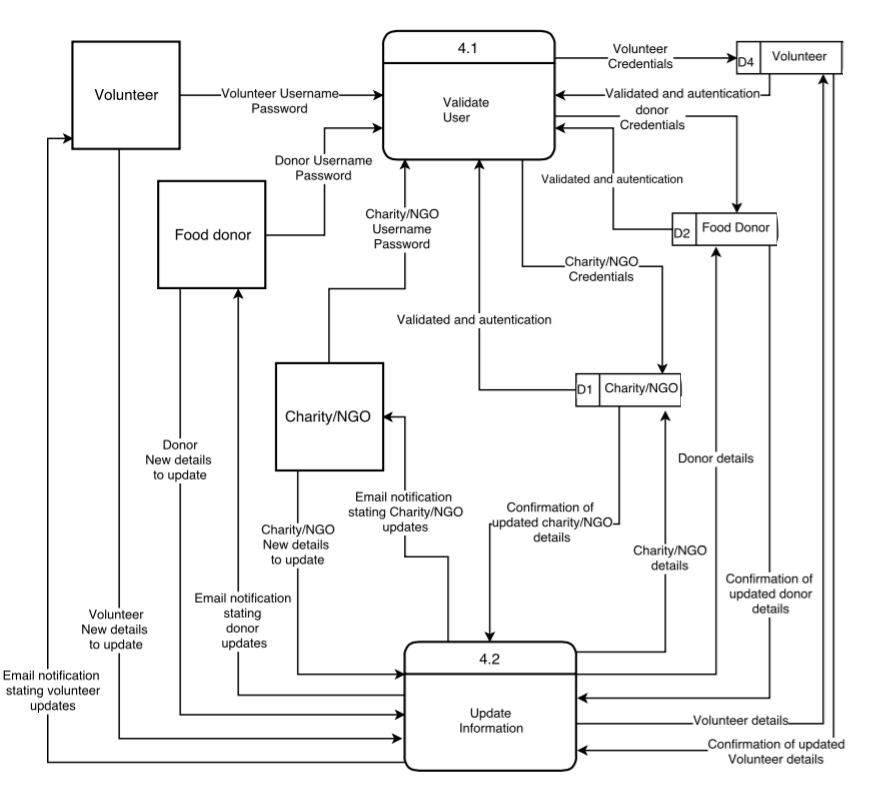
**Process 2: Create, Validate account and verify Authenticity – Charity / NGO**

****

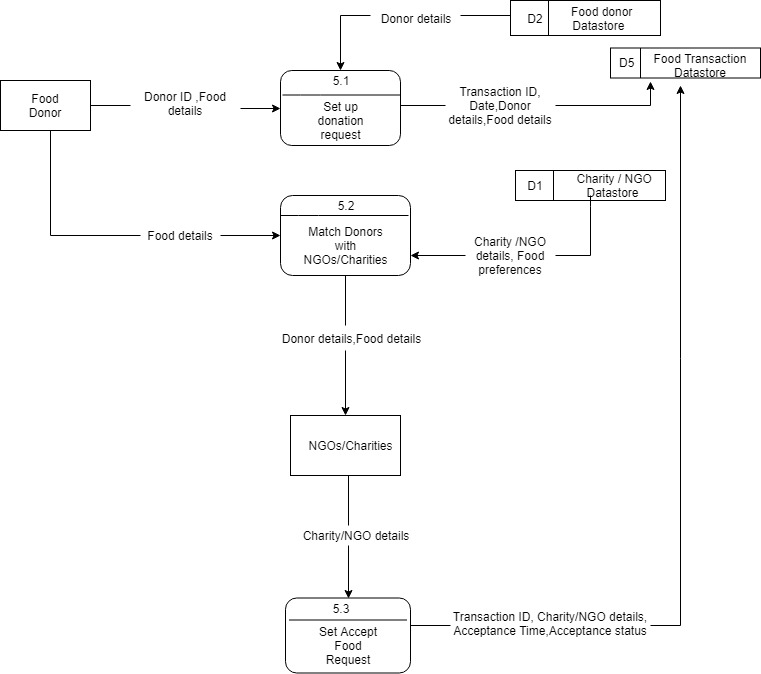
**Process 3: Create and Validate account - Volunteer**



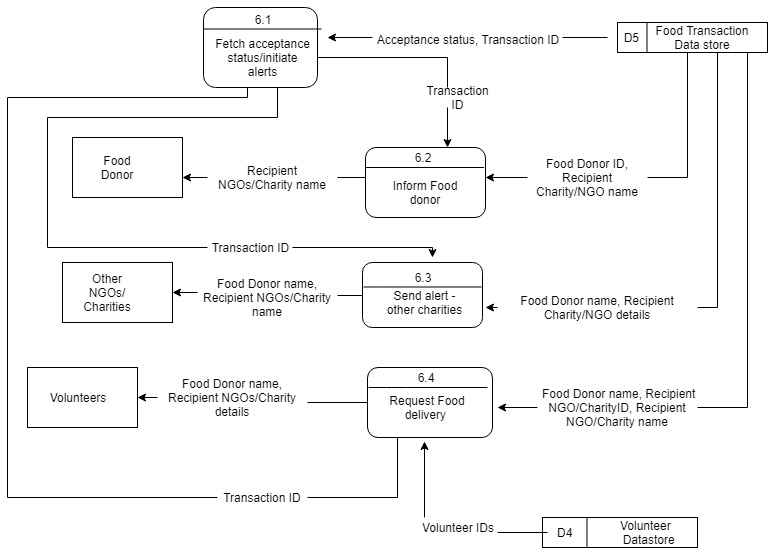
**Process 4: Validate user**



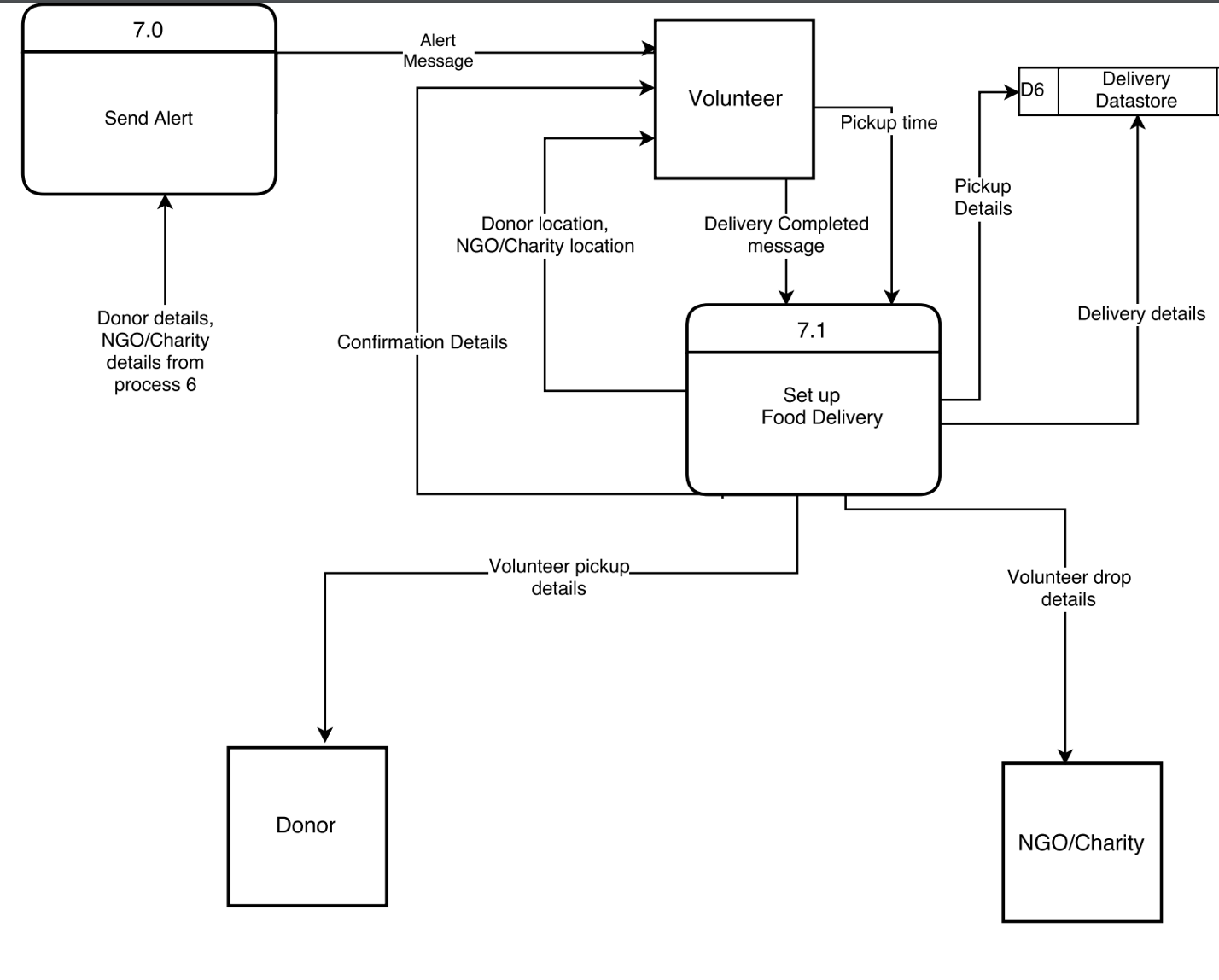
**Process 5: Networking services**



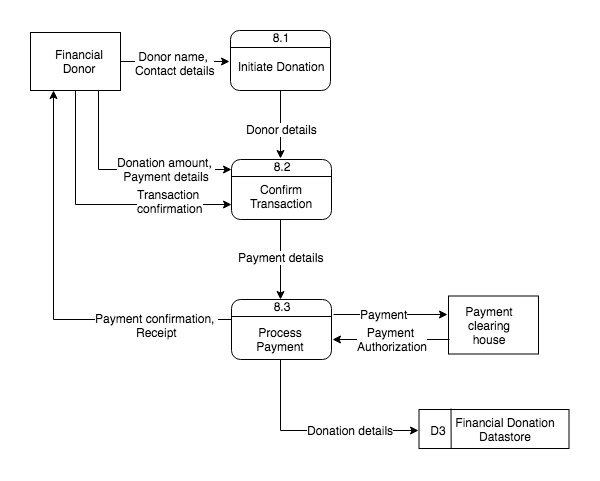
**Process 6: Real time updates and alerts about availability of food**



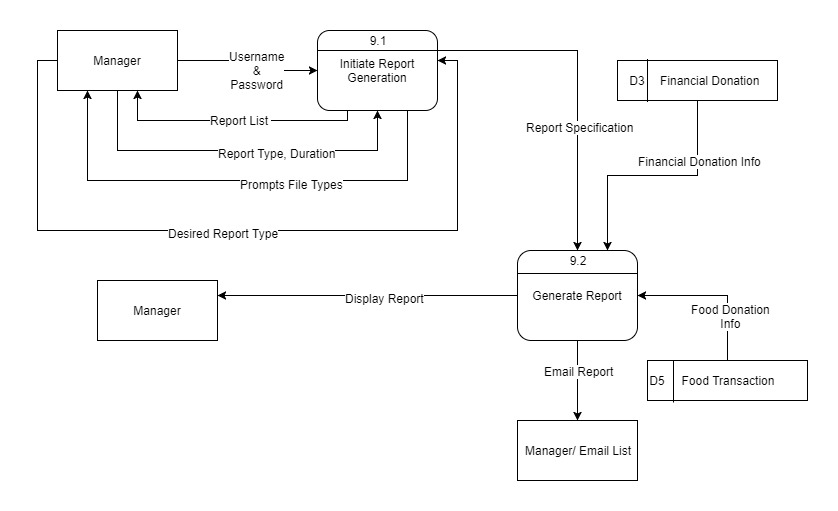
**Process 7: Food Delivery**



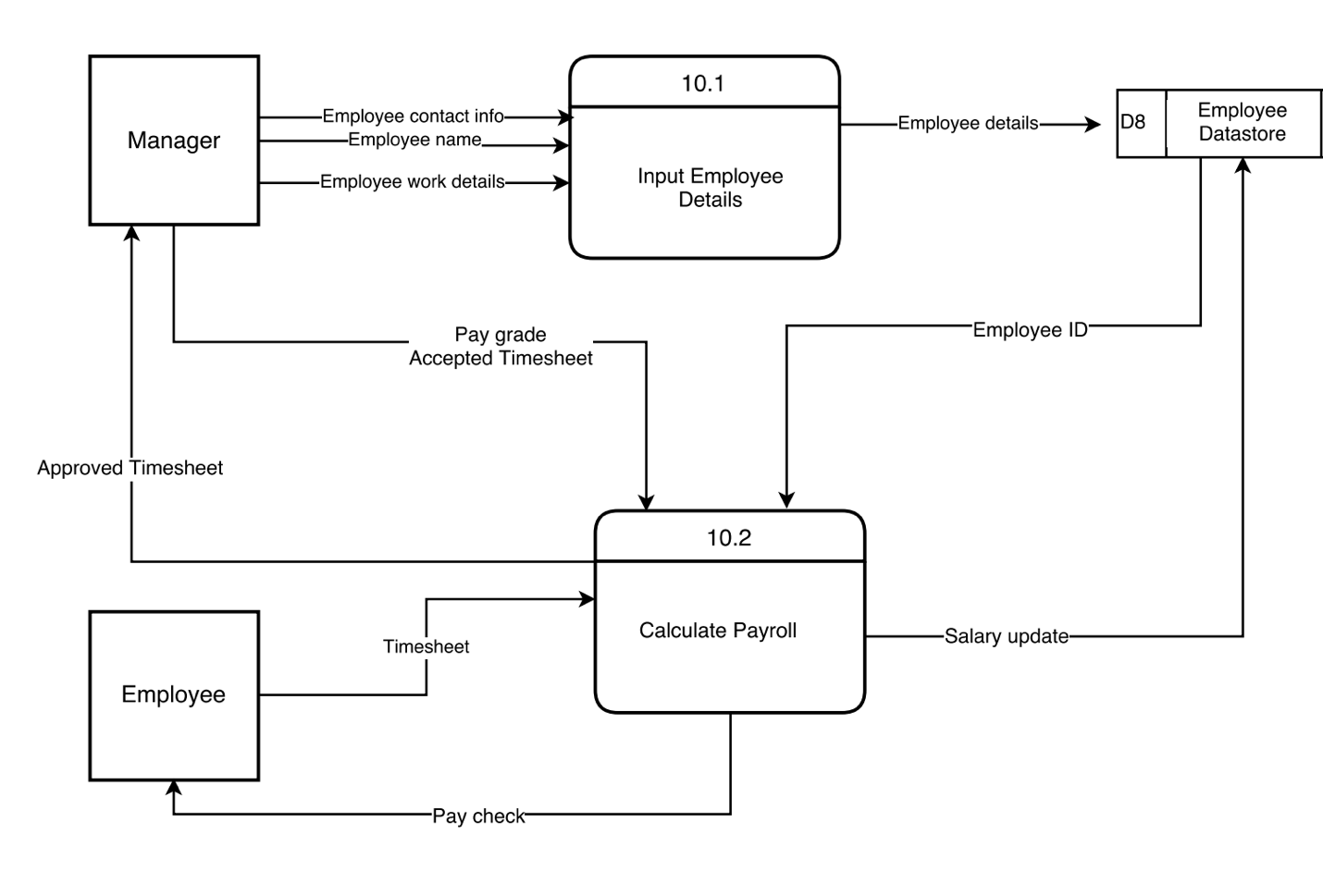
**Process 8: Donate Money**



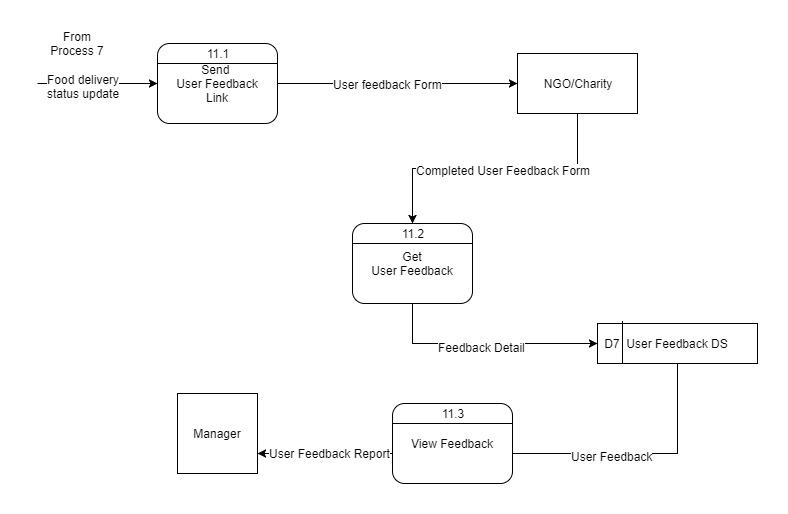
**Process 9: Generate Report**



**Process 10: Personnel management**



**Process 11: Get User Feedback**



**12.Data Dictionary**

**Charity / NGO Datastore**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| charity\_id | charity\_name | address | email\_id | contact\_nr | size | food\_preference |
| Primary Key | Not Null | Not Null | Not Null | Not Null | Not Null |  |
| 1001 | Feeding SF | 1042, Kearny Street, CA 94133 | feedingsf@gmail.com | 650-124-5699 | 100 | Vegetarian |

**Data Dictionary for Charity / NGO Datastore**

|  |  |  |
| --- | --- | --- |
| Column Name | Column Description | Data Type |
| charity\_id | Unique identifier generated by system for every Registered Charity/NGO | INT |
| charity\_name | Name of the Charity/NGO | VARCHAR |
| address | Address of the Charity/NGO | VARCHAR |
| email\_id | Email address of the Charity/NGO | VARCHAR |
| contact\_nr | Contact Phone number of the Charity/NGO | VARCHAR |
| size | Number of beneficiaries in the Charity/NGO | INT |
| food\_preference | Preference of the food desired by the Charity/NGO | VARCHAR |

**Food Donor Datastore**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| food\_donor\_id | donor\_name | address | email\_id | contact\_nr |
| Primary Key | Not Null | Not Null |  | Not Null |
| 2001 | ABC Corporation | 1256, Mission Street, CA 94146 | mike@abc.com | 650-999-0101 |

**Data Dictionary for Food Donor Datastore**

|  |  |  |
| --- | --- | --- |
| Column Name | Column Description | Data Type |
| food\_donor\_id | Unique identifier generated by system for every Food donor | INT |
| donor\_name | Name of the Donor | VARCHAR |
| address | Address of the Donor | VARCHAR |
| email\_id | Email address of the Donor | VARCHAR |
| contact\_nr | Contact Phone number of the Donor | VARCHAR |

**Volunteer Datastore**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| vol\_id | vol\_f\_nm | vol\_l\_nm | vol\_address | vol\_contact | vol\_email | vol\_veh\_type | vol\_veh\_type\_desc | vol\_desc\_info |
| Primary Key | Not Null | Not Null | Not Null | Not Null | Not Null | Not Null | Not Null | Null |
| 50001 | Charles | Kenny | 1033, XYZ Avenue, CA 94133 | 650-456-5689 | CKenny@gmail.com | 1 | Car | I am interested in Social Service |

**Data Dictionary for Volunteer Datastore**

|  |  |  |
| --- | --- | --- |
| Column Name | Column Description | Data Type |
| vol\_id | Unique identifier generated by system for every Volunteer | INT |
| vol\_f\_nm | First Name of the Volunteer | VARCHAR |
| vol\_l\_nm | Last Name of the Volunteer | VARCHAR |
| vol\_address | Address of the Employee | VARCHAR |
| vol\_contact | Contact Phone number of the Employee | VARCHAR |
| vol\_email | Email ID of the Employee | VARCHAR |
| vol\_veh\_type | Vehicle type based on size | INT |
| vol\_veh\_type\_desc | Vehicle type description | VARCHAR |
| vol\_desc\_info | About the Volunteer | VARCHAR |

**Employee Datastore**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| emp\_id | emp\_name | emp\_address | emp\_contact | emp\_email | emp\_dsigntn | emp\_paygrade | emp\_timesheet | emp\_salary |
| Primary Key | Not Null | Not Null | Not Null | Not Null | Not Null | Not Null | Not Null | Not Null |
| 30001 | John Smith | 1035, Walsh Avenue, CA 94133 | 650-124-5699 | Jsmith@gmail.com | Manager | B | 160 | 60,000 |

**Data Dictionary for Employee Datastore**

|  |  |  |
| --- | --- | --- |
| Column Name | Column Description | Data Type |
| emp\_id | Unique identifier generated by system for every Employee | INT |
| emp\_name | Name of the Employee | VARCHAR |
| emp\_address | Address of the Employee | VARCHAR |
| emp\_contact | Contact Phone number of the Employee | VARCHAR |
| emp\_email | Email ID of the Employee | VARCHAR |
| emp\_dsigntn | Designation of the employee | VARCHAR |
| emp\_paygrade | Pay grade of the employee | VARCHAR |
| emp\_timesheet | Number of hours clocked by the employee on timesheet | VARCHAR |
| emp\_salary | Salary of the employee | DOUBLE |

**Food Transaction Datastore**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| transaction\_id | donor\_id | charity\_id | date | food\_item | donor\_food\_type | charity-food\_type | people\_count | accept\_time | acceptance\_status |
| Primary Key | Foreign key | Foreign key | Not Null | Not Null |  |  | Not null | Not Null | Not Null |
| 1001 | 2001 | 3001 | 02/03/2018 | Lentil Soup, Bread Loaf | Vegetarian | Vegetarian | 30 | 20:30 | Accepted |

**Delivery Datastore**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| delivery\_id | transaction\_id | volunteer\_id | pick\_time | est\_del\_time | delivery\_status |
| Primary Key | Forign key | Foreign key | Not null | Not null | Not null |
| 9002 | 1001 | 6001 | 19:00 | 21:00 | Delivered |

**Data Dictionary for Food Transaction Datastore and Delivery Datastore**

|  |  |  |
| --- | --- | --- |
| Column Name | Column Description | Data Type |
| transaction\_id | Unique identifier generated by system for every food donation transaction | INT |
| donor\_id | Unique identifier assigned to food donors on account creation | VARCHAR |
| charity\_id | Unique identifier assigned to food donors on account creation | VARCHAR |
| date | Date of food donation | DATE |
| food\_item | Name of the food items being donated | VARCHAR |
| donor\_food\_type | The specific type of food being donated | VARCHAR |
| charity\_food\_type | The specific type of food preferred by the Charity | VARCHAR |
| people\_count | Number of people the donated food is expected to serve | INT |
| accept\_time | Time the donation is accepted by Charity/NGO | TIME |
| delivery\_id | Unique identifier assigned to the deliver | INT |
| volunteer\_id | Unique identifier assigned to volunteers on account creation | VARCHAR |
| pick\_time | Time given by volunteer for food pick up from donor | TIME |
| est\_del\_time | Estimated pick up time of donated food by volunteer | TIME |
| delivery\_status | Confirmation flag that food has been delivered to Charity/NGO | INT |

**Financial Donation Datastore**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| financial\_donation\_id | donor\_name | billing\_address | email\_id | contact\_nr | donation\_amount | donation\_date |
| Primary Key | Not Null | Not Null | Not Null | Not Null | Not Null | Not Null |
| 8001 | Ashton Paul | 2042, Millard drive, CA 95123 | apaul@gmail.com | 408-124-8800 | 70 | 2018-01-10 |

**Data Dictionary for Financial Donation Datastore**

|  |  |  |
| --- | --- | --- |
| Column Name | Column Description | Data Type |
| financial\_donation\_id | Unique identifier generated by system for every Financial Donation made | INT |
| donor\_name | Name of the Financial donor | VARCHAR |
| billing\_address | Billing Address of the Financial donor | VARCHAR |
| email\_id | Email address of the Financial donor | VARCHAR |
| contact\_nr | Contact Phone number of the Financial donor | VARCHAR |
| donation\_amount | Amount donated in Dollars | FLOAT |
| donation\_date | Date of the financial donation | DATE |

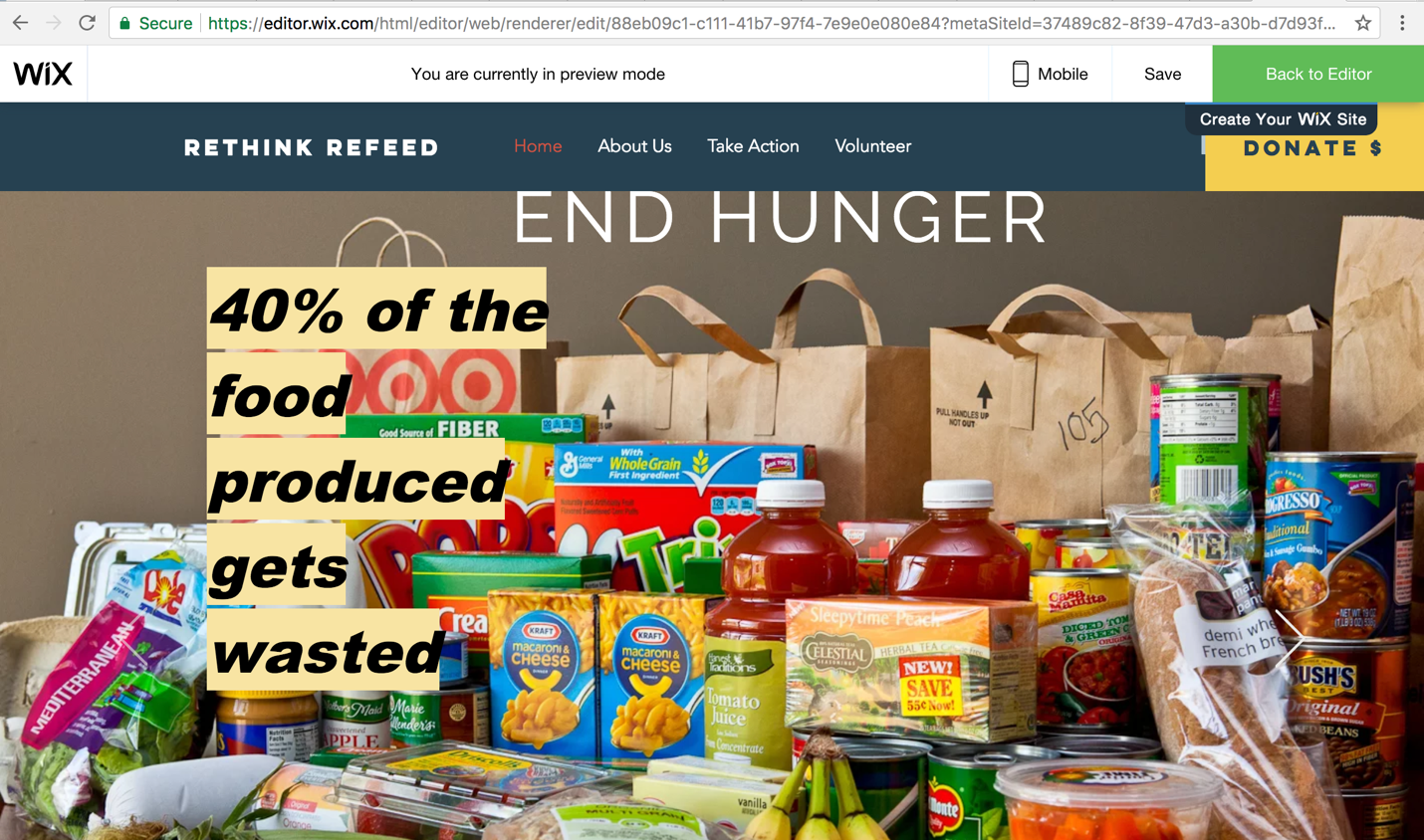
|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| fb\_id | transaction\_id | fb\_date | qn\_1 | qn\_2 | qn\_3 | qn\_4 | overall\_rate | other\_comnts |
| Primary Key | Foreign Key | Not Null | Not Null | Not Null | Not Null | Not Null | Not Null | Null |
| 9001 | 1001 | 03/07/2018 | 3 | 4 | 5 | 3 | 4.5 | Overall a good Service |

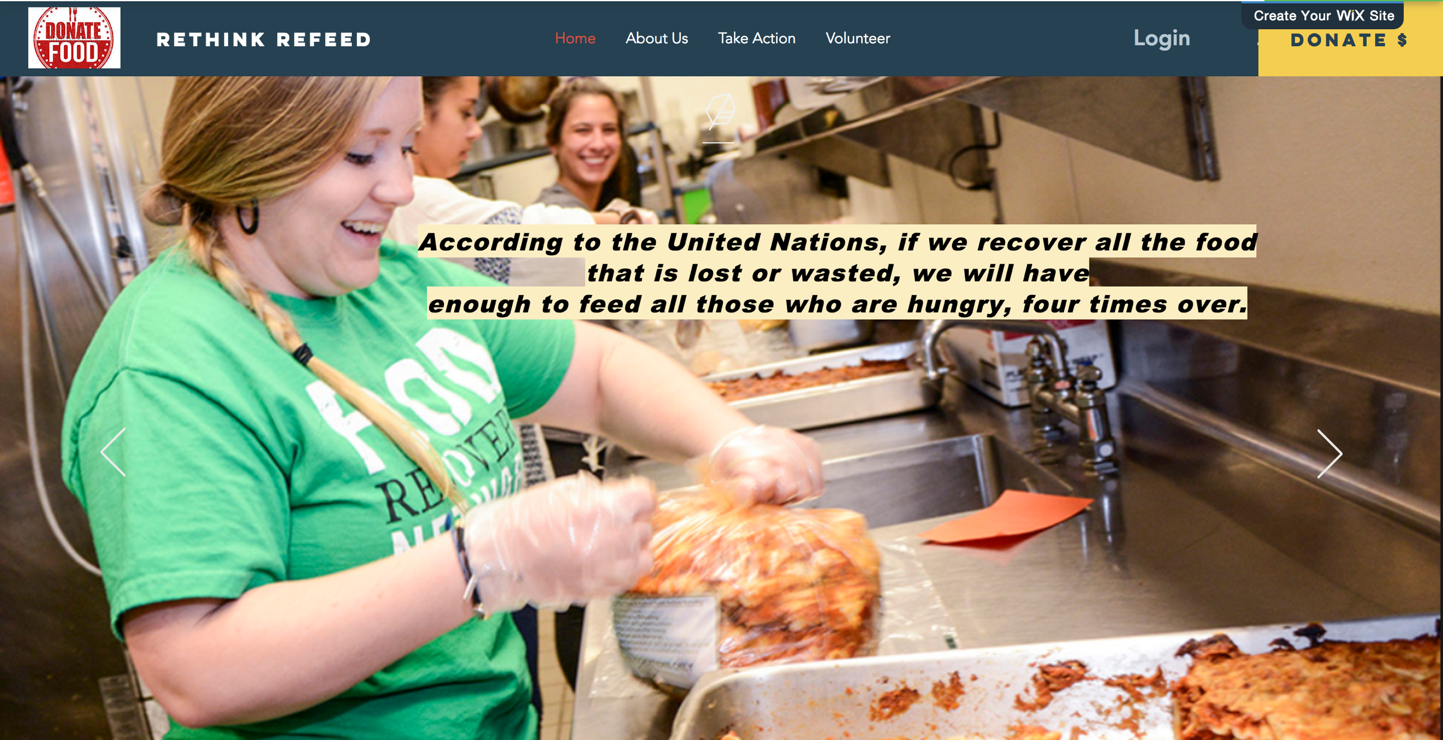
**User Feedback Datastore**

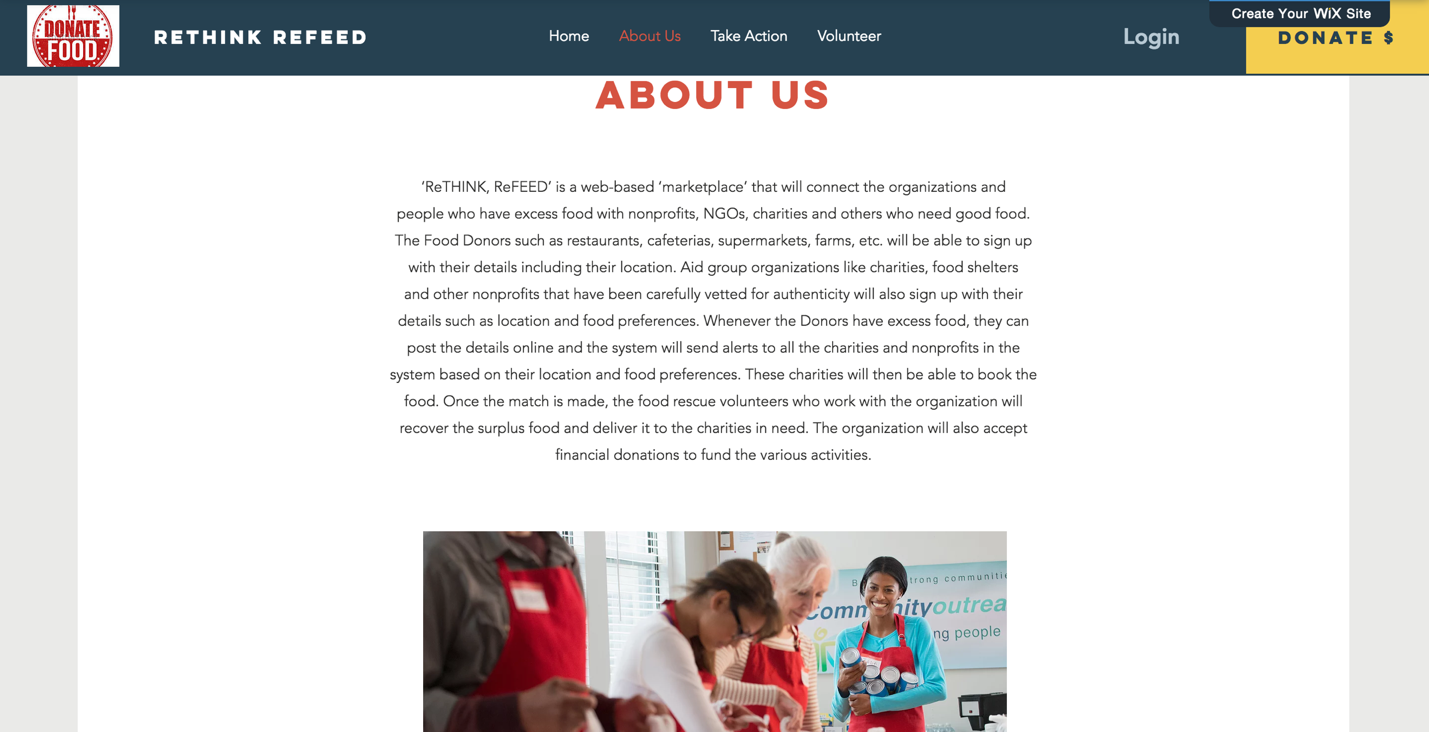
**Data Dictionary for User Feedback Datastore**

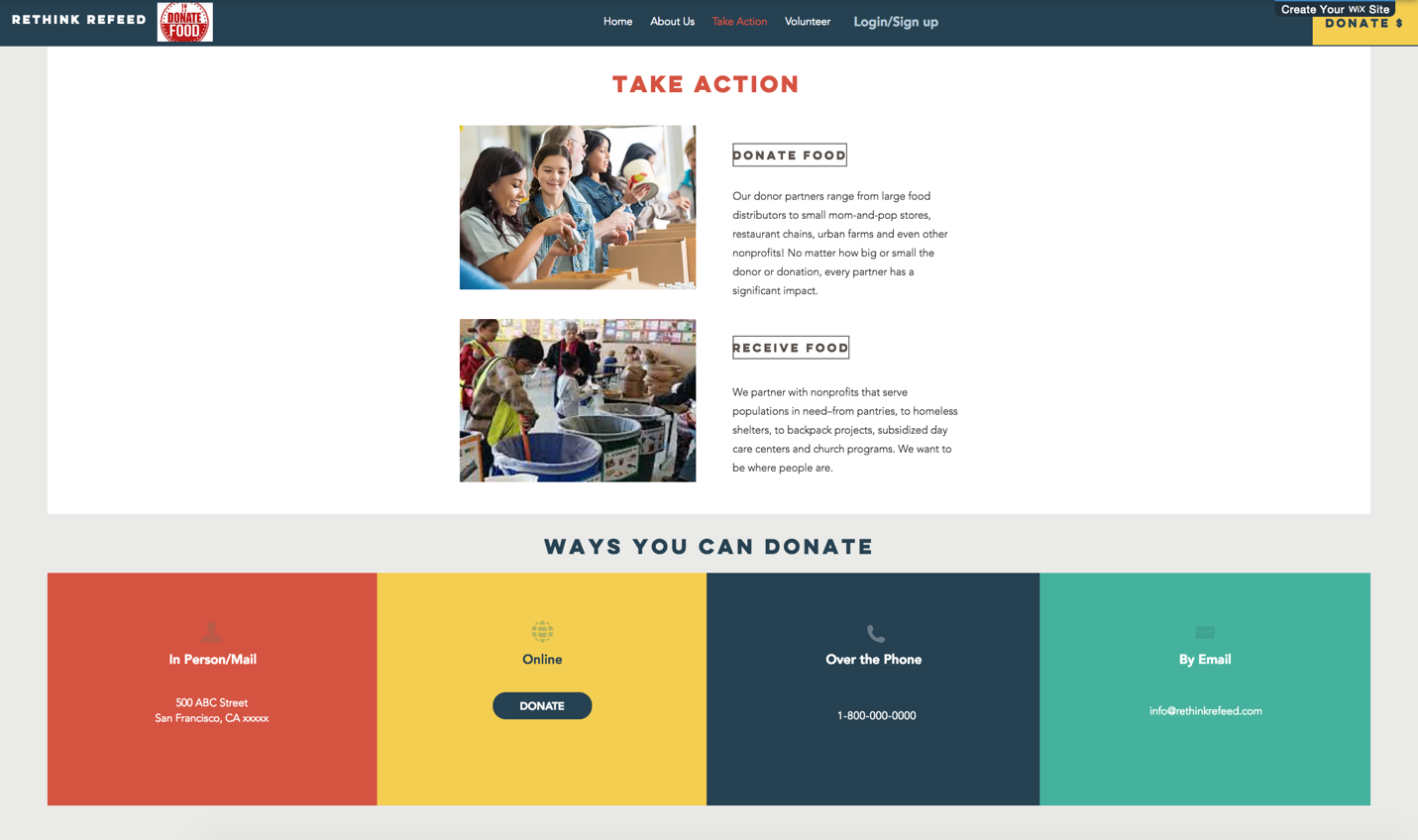
|  |  |  |
| --- | --- | --- |
| Column Name | Column Description | Data Type |
| fb\_id | Unique identifier generated by system for every Feedback | INT |
| transaction\_id | Food Donation Transaction ID | VARCHAR |
| fb\_date | Date when Feedback form is submitted by NGO/Charity | DATE |
| qn\_1 | Rating given for Feedback question 1 | INT |
| qn\_2 | Rating given for Feedback question 2 | INT |
| qn\_3 | Rating given for Feedback question 3 | INT |
| qn\_4 | Rating given for Feedback question 4 | INT |
| overall\_rate | Overall Rating given for the service | INT |
| other\_comnts | Other Feedback Given by NGO/Charity | VARCHAR |

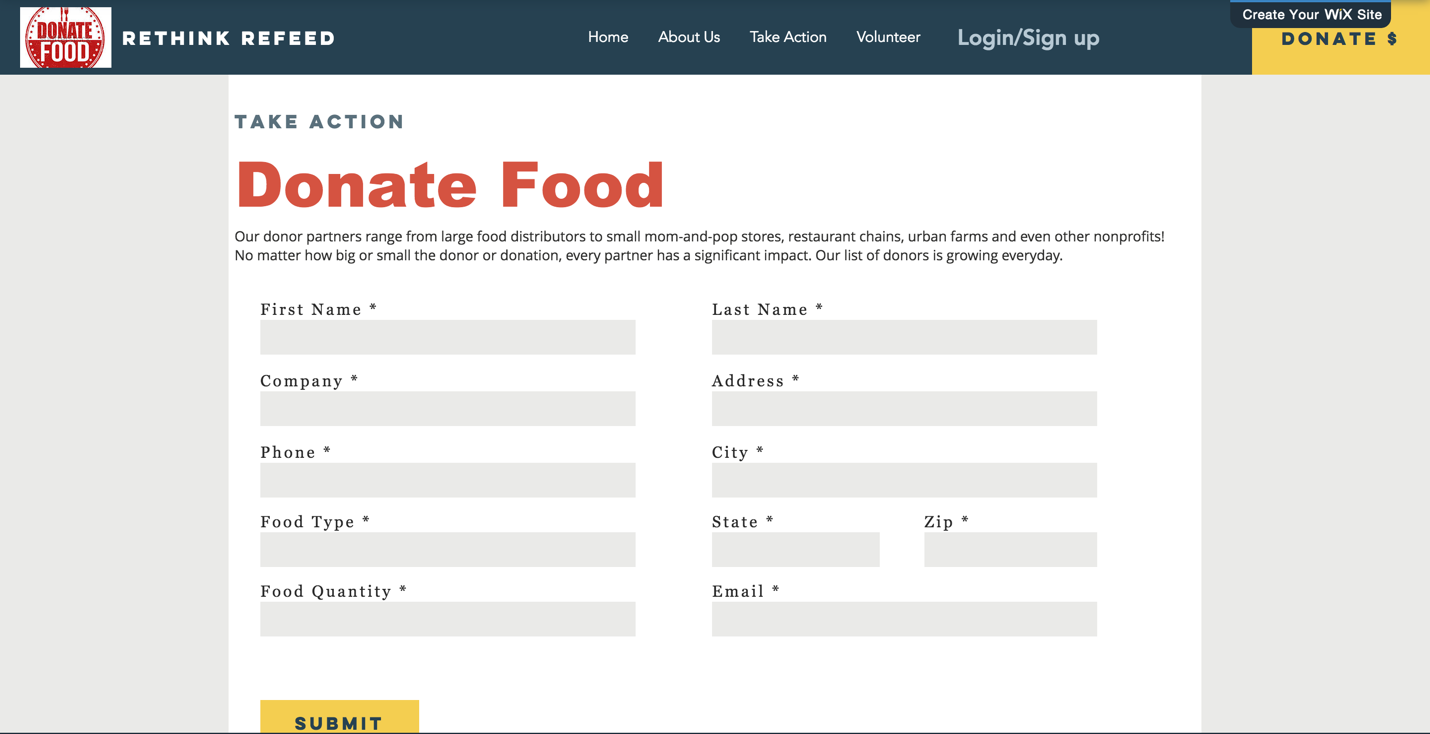
**14.User Interface**

****

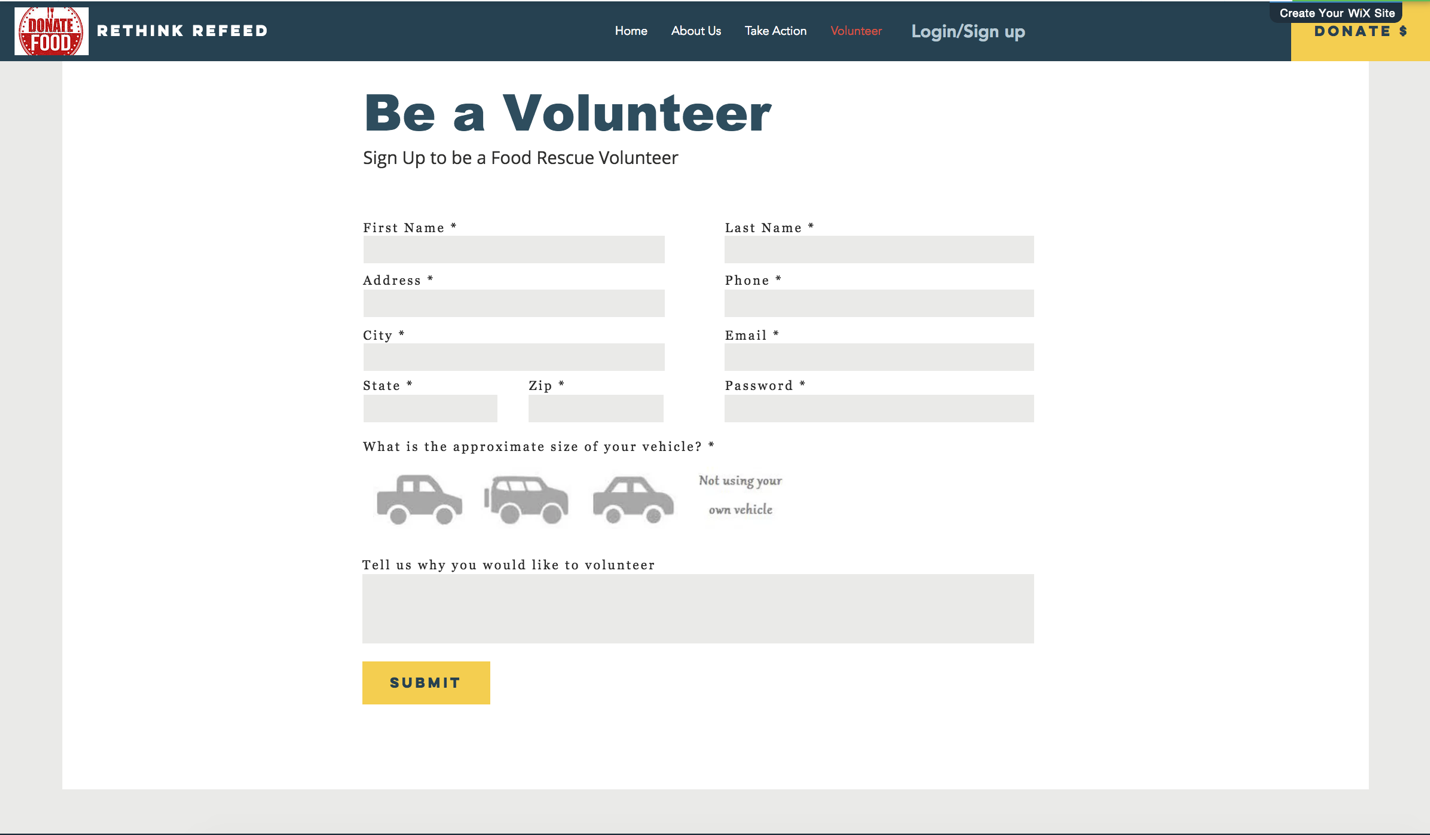
****

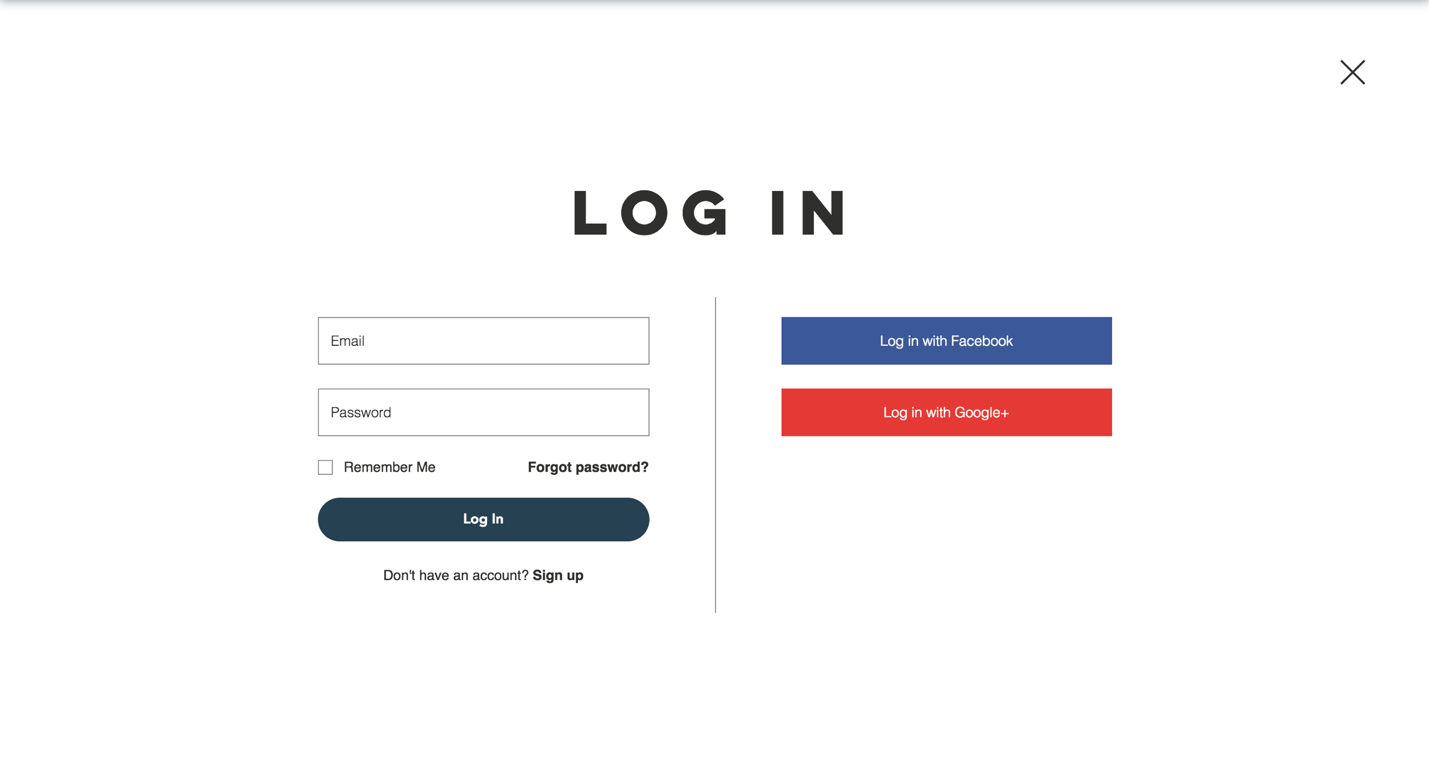
****

****

****

****

****

****

**15.Future Scope**

* Expand to other locations and markets.
* Develop a better-connected system to combine deliveries and match supply and demand efficiently.
* Tie-up with cab services (similar to Uber and Lyft) for food pick-up and delivery.
* In addition to delivering the excess food to charities, the organization can also set up facilities to directly distribute food to those in need (similar to soup kitchens).

**16.Conclusion**

**Our experience with this exercise:**

* Understanding and developing analytical skills and system designing capability.
* Understanding how to gather requirements and convert them into clear diagrammatic representations.
* Importance of dividing tasks and team work.
* Learning about consistency of data and its flow.