Software Requirements Specification

for

Food Bank System

Version 1.0 approved

Friday, March 1, 2024

Team Members:

Omar Mahmoud 2201216

Omar Sameh 21P0204

Abdulrahman Owais 21P0300

Alaa Yasser Fathy 21P0408

Table of Contents

Table of Contents ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

2. Overall Description 1

2.1 Product Perspective 1

2.2 Product Functions 1

2.3 Use Case Diagram 3

2.4 Operating Environment 4

2.5 User Documentation 4

3. External Interface Requirements 4

3.1 User Interfaces 4

3.2 Hardware Interfaces 4

3.3 Software Interfaces 4

4. System Features 5

4.1 View Total Month Donations 5

4.2 Add Distributor 5

4.3 Edit Distributor 6

4.4 Remove Distributor 6

4.5 Add program 7

4.6 Edit Program 7

4.7 View Items 8

4.8 Remove Program 8

4.9 Add Supplier 9

4.10 Edit Supplier 9

4.11 Remove Supplier 10

4.12 View Suppliers 10

4.13 View Programs 11

4.14 View Distributors 11

4.15 Add Item 12

4.16 Remove Item 12

4.17 Edit Item 13

4.18 Add Donor 13

4.19 Delete Donor 14

4.20 Edit Donor 14

4.21 Get Donor By Email 15

4.22 Send Confirmation Email 15

4.23 Request to Volunteer 16

4.24 Update Request Status 16

4.25 Send Volunteer Email 17

4.26 Request Food 17

5. Other Nonfunctional Requirements 18

5.1 Performance Requirements 18

5.2 Security Requirements 18

5.3 Usability Requirements 18

Appendix B: Analysis Models 19

# Introduction

## Purpose

The project aims to design a Food Bank System. This is the first version (version 1.0). This SRS document covers the whole system and not only a part of it.

## Document Conventions

This document follows the IEEE convention, all rights are reserved.

## Intended Audience and Reading Suggestions

This document is intended for developers, project managers, users, testers, and documentation writers. Refer to [table of contents](#contents) to know the contents of this document and how it is organized. Users are suggested to read only the following sections: “[Overall Description](#_Overall_Description)”, “[System Features](#_System_Features)”.

## Product Scope

This product’s goal is to facilitate the daily processes of the bank system and help gather donations for a Food Bank.

# Overall Description

## Product Perspective

The product is supposed to serve a non-profit organization which is the Food Bank. The product should facilitate the daily processes of the bank system.

## Product Functions

* Warehouse Coordinator
  1. Add Item
  2. Remove Item
  3. Edit Item
  4. View items
  5. Send volunteer email
* Procurement Coordinator
  1. Add Supplier
  2. Edit Supplier
  3. Remove Supplier
  4. View Suppliers
* Executive director
  1. View Total Month Donations
  2. Add Distributor
  3. Edit Distributor
  4. Remove Distributor
  5. View Distributors
* Donor
  1. Donate Money
  2. Request to Volunteer
* Recipient
  1. Request Food
* Programs Coordinator
  1. Add Program
  2. Edit Program
  3. Remove Program
  4. View Programs
  5. View Items

## Use Case Diagram

A diagram of a diagram

Description automatically generated

## Operating Environment

A robust and secure operating environment is needed to ensure optimum functionality.

The following are key components:

* **Web Server:** A web server is required to host the website.
* **Database Management System:** A database management system (DBMS) is required to store and manage the data related to the Food Bank. Mysql will be used.
* **Web Framework:** Php will be used for this product.
* **Operating System:** An operating system is required to run the web server. For example, Windows.

## User Documentation

No user documentation will be available. Tutorials on how to use the product can be made later if necessary.

# External Interface Requirements

## User Interfaces

The user interface should be user-friendly, and easy to use and navigate. Details of the user interface design should be documented in a separate user interface specification.

## Hardware Interfaces

The product will be supported through web browsers on multiple platforms such as: Windows, macOS, Android.

## Software Interfaces

The database engine to be used for this product is MySQL.

# System Features

## View Total Month Donations

|  |  |
| --- | --- |
| Code | R01 |
| Name | ViewTotalMonthDonations |
| Description | Allows the executive director to view the total of all donations for any of the months in the current year. The function should return the total money for all the donations for the entered month by getting a query from the "donations" database and calculating the sum of all donations. |
| Cost | 1h |
| Criticality | Med |
| Dependencies | No dependency |
| Pre-conditions | No conditions |
| Post-conditions | No conditions |
| Input | Selected Month of type Integer |
| Output | Total Donations of type Float |

## Add Distributor

|  |  |
| --- | --- |
| Code | R02 |
| Name | AddDistributor |
| Description | Allows the executive director to add a new distributor to the "distributors" database.  Check first if the distributor’s name is not already in the database.  Return 0 if inserted successfully, else return 1. |
| Cost | 1h |
| Criticality | High |
| Dependencies | No dependency |
| Pre-conditions | Distributor does not exist in the "distributors" database. |
| Post-conditions | A new record is added in the "distributors" database. |
| Input | A Distributor object |
| Output | Flag of type Boolean |

## Edit Distributor

|  |  |
| --- | --- |
| Code | R03 |
| Name | EditDistributor |
| Description | Allows the executive director to edit an existing distributor in the "distributors" database.  If the distributor does not exist in the database return 1. Else update all the attributes of the existing distributor and return 0. |
| Cost | 1h |
| Criticality | High |
| Dependencies | No dependency |
| Pre-conditions | Distributor exists in "distributors" database |
| Post-conditions | A record is updated in the "distributors" database. |
| Input | No input |
| Output | Flag of type Boolean |

## Remove Distributor

|  |  |
| --- | --- |
| Code | R04 |
| Name | RemoveDistributor |
| Description | Allows the executive director to remove an existing distributor from the "distributors" database.  If the distributor does not exist return 1. Else delete the distributor from the database and return 0. |
| Cost | 1h |
| Criticality | High |
| Dependencies | No dependency |
| Pre-conditions | Distributor exists in "distributors" database |
| Post-conditions | A record is removed from the "distributors" database. |
| Input | No input |
| Output | Flag of type Boolean |

## Add program

|  |  |
| --- | --- |
| Code | R05 |
| Name | AddProgram |
| Description | Allows the program coordinator to add a new program to the "programs" database.  Check first if the program’s name is not already in the database.  Return 0 if inserted successfully, else return 1. |
| Cost | 1h |
| Criticality | High |
| Dependencies | No dependency |
| Pre-conditions | Program does not exist in "programs" database |
| Post-conditions | A new record is added in the "programs" database. |
| Input | A Program object |
| Output | Flag of type Boolean |

## Edit Program

|  |  |
| --- | --- |
| Code | R06 |
| Name | EditProgram |
| Description | Allows the programs coordinator to edit an existing program in the "programs" database.  If the program does not exist return 1. Else update all the attributes of the existing program from the database and return 0. |
| Cost | 1h |
| Criticality | High |
| Dependencies | No dependency |
| Pre-conditions | Program exists in "programs" database |
| Post-conditions | A record is updated in the "programs" database. |
| Input | No input |
| Output | Flag of type Boolean |

## View Items

|  |  |
| --- | --- |
| Code | R07 |
| Name | ViewItems |
| Description | Allows the warehouse coordinator and programs coordinator to see all items in the "items" database.  Return a PDO object with all rows form the database by using SELECT statement. |
| Cost | 1h |
| Criticality | High |
| Dependencies | No dependency |
| Pre-conditions | No condition |
| Post-conditions | No condition |
| Input | No Input |
| Output | PDO object |

## Remove Program

|  |  |
| --- | --- |
| Code | R08 |
| Name | RemoveProgram |
| Description | Allows the programs coordinator to remove an existing program from the "programs" database.  If the program does not exist return 1. Else delete the program from the database and return 0. |
| Cost | 1h |
| Criticality | High |
| Dependencies | No dependency |
| Pre-conditions | Program exists in "programs" database |
| Post-conditions | A record is removed from the "programs" database. |
| Input | No input |
| Output | Flag of type Boolean |

## Add Supplier

|  |  |
| --- | --- |
| Code | R09 |
| Name | AddSupplier |
| Description | Allows the procurement coordinator to add a new supplier to the "suppliers" database.  Check first if the supplier is not already in the database.  Return 0 if inserted successfully, else return 1. |
| Cost | 1h |
| Criticality | High |
| Dependencies | No dependency |
| Pre-conditions | Supplier’s name does not exist in "suppliers" database |
| Post-conditions | A new record is added in the "suppliers" database. |
| Input | A Supplier object |
| Output | Flag of type Boolean |

## Edit Supplier

|  |  |
| --- | --- |
| Code | R10 |
| Name | EditSupplier |
| Description | Allows the procurement coordinator to edit an existing supplier in the "suppliers" database.  If the supplier does not exist return 1.  Else update all the attributes of the existing supplier from the database and return 0. |
| Cost | 1h |
| Criticality | High |
| Dependencies | No dependency |
| Pre-conditions | Supplier exists in "suppliers" database |
| Post-conditions | A record is updated in the "suppliers" database. |
| Input | No input |
| Output | Flag of type Boolean |

## Remove Supplier

|  |  |
| --- | --- |
| Code | R11 |
| Name | RemoveSupplier |
| Description | Allows the procurement coordinator to remove an existing supplier from the "suppliers" database.  If the supplier does not exist return 1.  Else delete the supplier from the database and return 0. |
| Cost | 1h |
| Criticality | High |
| Dependencies | No dependency |
| Pre-conditions | Supplier exists in " suppliers" database |
| Post-conditions | A record is removed from the "suppliers" database. |
| Input | No input |
| Output | Flag of type Boolean |

## View Suppliers

|  |  |
| --- | --- |
| Code | R12 |
| Name | ViewSuppliers |
| Description | Allows the procurement coordinator to see all suppliers in the "suppliers" database.  Return a PDO object with all rows form the database by using SELECT statement. |
| Cost | 1h |
| Criticality | High |
| Dependencies | No dependency |
| Pre-conditions | No condition |
| Post-conditions | No condition |
| Input | No Input |
| Output | PDO object |

## View Programs

|  |  |
| --- | --- |
| Code | R13 |
| Name | ViewPrograms |
| Description | Allows the programs coordinator to see all programs in the "programs" database.  Return a PDO object with all rows form the database by using SELECT statement. |
| Cost | 1h |
| Criticality | High |
| Dependencies | No dependency |
| Pre-conditions | No condition |
| Post-conditions | No condition |
| Input | No Input |
| Output | PDO object |

## View Distributors

|  |  |
| --- | --- |
| Code | R14 |
| Name | ViewDistributors |
| Description | Allows the executive director to see all distributors in the "distributors " database.  Return a PDO object with all rows form the database by using SELECT statement. |
| Cost | 1h |
| Criticality | High |
| Dependencies | No dependency |
| Pre-conditions | No condition |
| Post-conditions | No condition |
| Input | No Input |
| Output | PDO object |

## Add Item

|  |  |
| --- | --- |
| Code | R15 |
| Name | addItem |
| Description | This function is called to add a new item to the "items" database. An “INSERT INTO” statement is used to add an item with the corresponding attributes to the database.  Check first if the item is not already in the database.  Return 0 if inserted successfully, else return 1. |
| Critically | High |
| Cost | 1h |
| Dependencies | No dependencies |
| Pre-conditions | Item’s name does not exist in "items" database |
| Post-conditions | A new item added to the "items" database |
| Input | An Item object |
| Output | Flag of type Boolean |

## Remove Item

|  |  |
| --- | --- |
| Code | R16 |
| Name | removeItem |
| Description | This function is called to remove an existing item from the "items" database. If the item does not exist in the database return 1. Else, use the “DELETE FROM” statement to delete the database and return 0. |
| Critically | High |
| Cost | 1h |
| Dependencies | No dependencies |
| Pre-conditions | Item must exist in "items" database. |
| Post-conditions | Item removed from the program’s list. |
| Input | No input |
| Output | Flag of type Boolean |

## Edit Item

|  |  |
| --- | --- |
| Code | R17 |
| Name | editItem |
| Description | This function is called to edit an existing item in the "items" database.  If the item does not exist return 1. Else update all the attributes of the existing item and return 0. |
| Critically | 1h |
| Cost | High |
| Dependencies | No dependency |
| Pre-conditions | Item exists in "items" database |
| Post-conditions | A record is updated in the "items" database. |
| Input | No input |
| Output | Flag of type Boolean |

## Add Donor

|  |  |
| --- | --- |
| Code | R18 |
| Name | addDonor |
| Description | This function is called to add a new donor to the donor database. An “INSERT INTO” statement is used to add a donor with the corresponding attributes to the database.  Check first if the donor is not already in the database.  Return 0 if inserted successfully, else return 1. |
| Critically | high |
| Cost | 1h |
| Dependencies | No dependencies |
| Pre-conditions | Donor’s username does not exist in "donors" database |
| Post-conditions | The database is updated if the function returns true. Otherwise, it returns false (no changes) |
| Input | A Donor object |
| Output | Boolean flag indicating whether adding the donor was successful or not |

## Delete Donor

|  |  |
| --- | --- |
| Code | R19 |
| Name | deleteDonor |
| Description | This function is called to delete an existing donor in the donor database. If donor’s username does not exist return 1.  Else, use the “DELETE FROM” statement to delete the donor from the database and return 0. |
| Critically | high |
| Cost | 1h |
| Dependencies | No dependencies |
| Pre-conditions | Donor exists in “Donor” database |
| Post-conditions | The database is updated if the function returns true. Otherwise, it returns false (no changes) |
| Input | No input |
| Output | Boolean flag indicating whether deleting the donor was successful or not |

## Edit Donor

|  |  |
| --- | --- |
| Code | R20 |
| Name | editDonor |
| Description | This function is called to update the attributes of an existing donor in the donor database. An “UPDATE” statement is used to update the attributes of the donor in the database. The donor is found using the “WHERE” statement. If the donor does not exist return 1. Else update all the attributes of the existing item and return 0. |
| Critically | high |
| Cost | 1h |
| Dependencies | No dependencies |
| Pre-conditions | Donor exists in “Donor” database |
| Post-conditions | The database is updated if the function returns true. Otherwise, it returns false (no changes) |
| Input | No input |
| Output | Boolean flag indicating whether updating the donor was successful or not |

## Get Donor By Email

|  |  |
| --- | --- |
| Code | R21 |
| Name | getDonorByEmail |
| Description | The function executes an SQL query to retrieve donor data from the donors table based on the provided email using the "SELECT" statement.  If a donor with the given email exists in the database, create a donor object from the returned associative array containing the donor's details.  Else if no donor with the given email is found, return null to indicate that no donor was found. |
| Critically | high |
| Cost | 1h |
| Dependencies | No dependecies |
| Pre-conditions | Donor exists in “Donor” database |
| Post-conditions | No post-conditions |
| Input | String email |
| Output | A Donor object |

## Send Confirmation Email

|  |  |
| --- | --- |
| Code | R22 |
| Name | sendConfirmationEmail |
| Description | The email subject and message are constructed (stored in 2 variables, 1 for the subject and the other for the message) using the name of the provided donor and the donation amount. The subject contains the string “Donation Confirmation”, and the message contains a thank you message that specifies the amount and the donation’s expiration date. A headers variable is created to store the header of the email which includes the sender's email address and reply-to address. The mail() function is used to send the email. This function returns true if the email is sent successfully and false otherwise. |
| Critically | high |
| Cost | 1h |
| Dependencies | No dependencies |
| Pre-conditions | No pre-conditions |
| Post-conditions | An email is sent to the donor. |
| Input | No input |
| Output | Boolean flag indicating whether an email has been successfully sent or not. |

## Request to Volunteer

|  |  |
| --- | --- |
| Code | R23 |
| Name | requesttoVolunteer |
| Description | The function adds a new request to the database using the “INSERT INTO” statement. The requestStatus is set to create. Call the function sendRequestEmail() and return 0 if the creation was successful. Otherwise return 1. |
| Critically | high |
| Cost | 1h |
| Dependencies | No dependencies |
| Pre-conditions | No pre-conditions. |
| Post-conditions | The “Volunteer Request” database is updated if the function returns 0. Otherwise, it returns 1 (no changes). |
| Input | No input. |
| Output | Boolean flag indicating whether creating the donation was successful or not |

## Update Request Status

|  |  |
| --- | --- |
| Code | R24 |
| Name | updateRequestStatus |
| Description | The SQL query updates the requestStatus attribute in the “Volunteer Request” table for the specified requestID. If the query execution is successful, the function returns 0, indicating that the request status was updated successfully. Otherwise, it returns 1. |
| Critically | High |
| Cost | 1h |
| Dependencies | requesttoVolunteer |
| Pre-conditions | Request exists in “Volunteer Request” database. |
| Post-conditions | The status of the request is updated. |
| Input | The new status of RequestStatus Type. |
| Output | Boolean flag indicating whether updating the request status was successful or not |

## Send Volunteer Email

|  |  |
| --- | --- |
| Code | R25 |
| Name | sendVolunteerEmail |
| Description | The email subject and message are constructed (stored in 2 variables, 1 for the subject and the other for the message) using the name of the provided donor and the program. The subject contains the string “Confirmation of Volunteer Application”, and the message contains a thank you message that includes the application status and other contact credentials. A headers variable is created to store the header of the email which includes the sender's email address and reply-to address. The mail() function is used to send the email. This function returns true if the email is sent successfully and false otherwise. |
| Critically | high |
| Cost | 1h |
| Dependencies | No dependencies. |
| Pre-conditions | No pre-conditions. |
| Post-conditions | an email is sent to the food bank system. |
| Input | No input |
| Output | Boolean flag indicating whether an email has been successfully sent or not. |

## 

## Request Food

|  |  |
| --- | --- |
| Code | R26 |
| Name | requestFood |
| Description | The email subject and message are constructed (stored in 2 variables, 1 for the subject and the other for the message) using the name of the provided recipient. The subject contains the string “Request Food Application”, and the message contains all attribute of the recipient. A headers variable is created to store the header of the email which includes the sender's email address and reply-to address. The mail() function is used to send the email. This function returns true if the email is sent successfully and false otherwise. |
| Critically | high |
| Cost | 1h |
| Dependencies | No dependencies |
| Pre-conditions | No pre-conditions |
| Post-conditions | No post-conditions |
| Input | No input |
| Output | Boolean flag indicating whether an email has been successfully sent or not. |

## Calculate Total Value Donated

|  |  |
| --- | --- |
| Code | R27 |
| Name | calcTotalValueDonated() |
| Description | The function gets all donations from the “donation details” database, filter the donations by the donor username. Calculate the total of donations by summing the total cost attribute of each donation. Return the sum, or return 0 if no donations were made by the donor. |
| Critically | high |
| Cost | 1h |
| Dependencies | No dependencies |
| Pre-conditions | Donor must be logged in |
| Post-conditions | No post-conditions |
| Input | No input |
| Output | Total value of donations of type float |

## Request Time Off

|  |  |
| --- | --- |
| Code | R28 |
| Name | RequestTimeoff() |
| Description | The employee enters the day they wish to have time off, enter the reason for this. An email is sent to the executive at [fbankexecutive@gmail.com](mailto:fbankexecutive@gmail.com) with the subject “Day Off”. The message is constructed using the day and reason entered by the employee.  Return 0 if sent successfully, else return 1. |
| Critically | Med |
| Cost | 1h |
| Dependencies | No dependencies |
| Pre-conditions | No pre-conditions |
| Post-conditions | An email is sent to mentioned address above |
| Input | No input |
| Output | Flag of type boolean |

# Other Nonfunctional Requirements

## Performance Requirements

The system should be able to perform queries without significant performance degradation. The system will use a database management system (MySQL) instead of text files.

## Security Requirements

The system should be secure, protecting against sql injections by using the prepare and execute methods. Also, user authentication is required to access the system.

## Usability Requirements

The system should be easy to use and navigate, with a user-friendly interface. The system should provide a user interface that is available in English, in addition to, clearly labeled menus and buttons.

Arabic interface may be added later but it is not urgent.

Appendix B: Analysis Models

A computer screen shot of a computer

Description automatically generatedClass Diagram