



Software requirement specification document for project Charity Management System

Mohamed Ashraf, Farida Sherif, Farah Ashraf, Basma Tarek, Nada
Nady

Supervised by: Dr.

June , 8 2022

1 Introduction

1.1 Purpose of this document

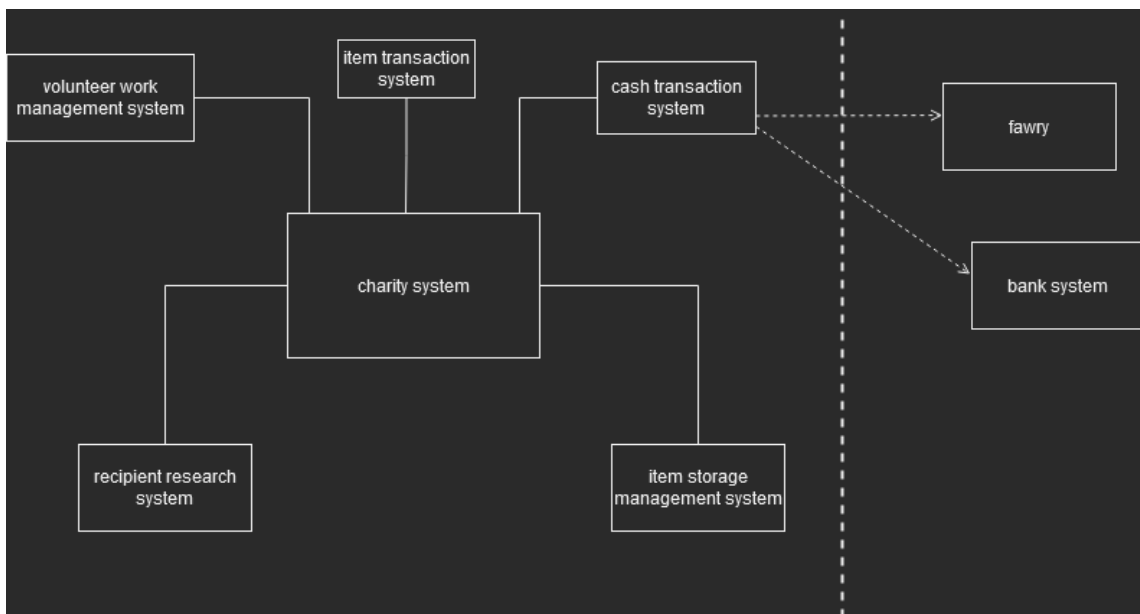
The purpose of this document is to build an online system to manage charity organization needs and to ease its use.

1.2 Scope of this document

The purpose of the charity management system is to ease the charity management for warehouse managers, accountants, executive directors, volunteers, volunteer managers, researchers, donors, recipients, and administrators.

1.3 Overview

This document contains all the requirements for donations that occur in the charity organization. The charity management system will manage the types of donations and how the donations are delivered.



1.4 Business Context

Reports creation about donations describing at what times of the year does the charity organization receive most of its donations to help the charity decide when it will give out the donations to the recipients.

2 General Description

2.1 Product Functions

Accountant

- Compare Expected & Received donations
- Create Annual Budget
- Calculate total income each year
- Prepare monthly cash donation report
- Report Monthly Expenses
- Update cash in ledger

Volunteer Manager

- Update number of volunteers
- Check number of volunteers in each volunteer work

Researcher

- Check if national id of recipient already exists
- Check if the recipient is in the excluded list
- Calculate difference between recipient's monthly income & expenses

Executive Director

- Create Annual Budget

Volunteer

- Register in volunteer work

Donor

- Make Donation
- Check Donation Status

2.2 Similar System Information

Mersal Charity is one of the systems similar to this system, but our system allows volunteers to choose the type of volunteer work they want to participate in and allows the donor to track his donations until they reach the recipients

2.3 User Characteristics

The users who will use the system can write in english and arabic language. The users are aware of the technological advances and will be able to make efficient use of the system. The users that will use the systems are the accountant, warehouse manager, volunteer, volunteer manager, executive director, admin, and donor.

2.4 User Problem Statement

The main user problem is that all the records of the donation details and users are documented on paper which are prone to damage and are difficult to edit.

2.5 User Objectives

The main user objective is to be able to store data efficiently in a database which will provide faster data access and will provide more security. Another objective is to be able to easily edit data.

2.6 General Constraints

software requirements like ease of use and speed constraints for example that the system should be able to handle the huge amount of data without the speed getting slower

3 Functional Requirements

1-

Code	p.1
Name	Compare expected & received donations

Type	Functional Requirement
Criticality	High
Input	An array of received donations each year and an integer that contains the expected amount of donated money each year.
Output	An integer containing the Difference between the expected and received donations each year and an integer that represents the total amount of donations in this year
Description	The function will add all the received donations from the array during the year and then subtract it from the expected yearly donation to find the difference.
Priority	9/10
Expected risks	If the amount expected or amount received is with null value
Preconditions	Difference between the expected and received is equal to null value login
Postconditions	There will be a new variable for the difference between the expected and received.
Dependences	This is depending on the amount received and expected donations being set.
Cost and schedule	1 hour
Technical issues	

2-

Code	p.2
Name	Create Annual Budget
Type	Functional Requirement
Criticality	High
Input	An integer containing Received donations
Output	5 integers

	<p>An integer containing the amount of donations going for cash</p> <p>An integer containing the amount of donations going for the purchase of appliances</p> <p>An integer containing the amount of donations going for salaries</p> <p>An integer containing amount of donations going for food supplies</p> <p>An integer containing the amount of donations going for medicines.</p>
Description	<p>30% of received donations to be donated as cash for recipients</p> <p>30% of received donations for purchasing appliances</p> <p>10% of received donations for salaries</p> <p>20% for food supplies</p> <p>10% for medicines</p>
Priority	9/10
Expected risks	That received donations may be with null value
Preconditions	<p>Donations going as cash are set to null</p> <p>Donations going for appliances are set to null</p> <p>Donations going for salaries are set to null</p> <p>Donations going for food supplies are set to null</p> <p>Donations going for medicines are set to null</p> <p>login</p> <p>Screen has only a title create annual budget</p>
Postconditions	<p>5 integers each containing amount of donation for each thing</p> <p>The screen has cash donation and what they're going for</p>
Dependences	This depends on setting the received donations
Cost and Schedule	2 hours
Technical Issues	

3-

Code	p.3
------	-----

Name	Report monthly expenses
Type	Functional requirement
Criticality	medium
Input	A file (expenses.txt) containing a date, description, amount of expenses. An integer representing a specific month.
Output	Returns the sum of all expenses during the month.
Description	The function will search in the file for a date that contains the same month that the function received and will then display each integer containing expenses with its corresponding description and date during the specified month. This function will also add all the values of the expenses of the specified month and display them.
Priority	8/10
Expected risks	Some of the expenses may be null values. Some of the date values may be null values. Some of the descriptions may be null values.
Preconditions	The screen will be blank. login
Postconditions	expenses, date of expense, description, and total expenses will be displayed on the screen.
Dependences	The dates of each expense.
Cost and Schedule	2 Hours
Technical Issues	

4-

Code	p.4
Name	Update warehouse capacity
Type	Functional requirement

Criticality	high
Input	A file (warehouse.txt) ,integer that contains the updated capacity value
Output	The output will be a file containing the updated warehouse capacity.
Description	The function will receive the object warehouse and put the integer value of the new capacity in the integer that represents capacity in the warehouse.
Priority	5/10
Expected risks	None
Preconditions	Login, the file warehousewith the un-updated integer that represents capacity in the object warehouse.
Postconditions	The integer representing the warehouse capacity will be updated to its current state.
Dependences	Current capacity of the warehouse
Cost and schedule	1 hour
Technical issues	

5.

Code	p.5
Name	calculate the difference between the recipient's monthly income and monthly expenses.
Type	Functional Requirement

Criticality	High
Input	A file (recipient.txt) containing Recipient Id An integer that carries the value of Recipient's monthly income An integer that carries the value of Recipient's monthly expenses
Output	Return the integer that representing the Difference between monthly expenses and monthly income
Description	The function will open the recipient file and search for the recipient id to get his/her monthly income and monthly expenses, then calculate the difference between monthly expenses and monthly income and put the calculated value in an integer.
Priority	9/10
Expected risks	That monthly expenses or monthly income might be with null values
Preconditions	Difference between monthly expenses and monthly income is set to null login
Postconditions	The difference between them will be set in files
Dependences	the received donations
Cost and schedule	1 hour
Technical issues	

6-

Name	Update quantity of an item
Type	Functional requirement
Code	p.6
Criticality	high
Input	(itemswarehouse.txt) containing the item ID, description, quantity, and location.

	An Item Id
Output	True in case the item quantity was updated successfully False in case an error occurred or the item quantity wasn't updated.
Description	The function will take the Id then will search for id of the item then the function will update the item data in the (itemswarehouse.txt) file with the new data.
Priority	6/10
Expected risks	The item may not exist
Preconditions	Login, old item data not updated in the file.
Postconditions	The (file itemswarehouse.txt) will have the updated data of the item.
Dependences	The item already exists in the file.
Cost and schedule	1 hour
Technical issues	

7-

Code	p.7
Name	Check if national id of recipient already exists
Type	Functional requirement
Criticality	high
Input	National id and file (recipient.txt)
Output	True or False True if it exists False if it doesn't
Description	The function will search for the received id in the file and will output true if it found it and false if it didn't.

Priority	7/10
Expected risks	National id record in the file may be a null value.
Preconditions	Login
Postconditions	none
Dependences	National id of recipient and previous records of all recipients.
Cost and schedule	45 minutes
Technical issues	

8-

Code	p.8
Name	Check if recipient id is in the excluded file
Type	Functional requirement
Criticality	high
Input	A file (excluded.txt) containing Id of excluded recipients
Output	True or False True if it exists False if it doesn't
Description	The function will search for the id in the (excluded.txt) file and check if the id of the recipient exists in the excluded file or not.

Priority	10/10
Expected risks	none
Preconditions	Login
Postconditions	none
Dependences	id of recipient that is sent to the function
Cost and schedule	45 minutes
Technical issues	

9.

Code	p.9
Name	Check number of volunteers in volunteer work
Type	Functional requirement
Criticality	low
Input	A file(AllvolunteerWorks.txt) containing id of volunteer work , the number of volunteers, the date of volunteer work and string.
Output	If the volunteer work has more than 20 volunteers on the same date; return false.

	If the volunteer work has not more than 20 volunteers on the same date; return true.
Description	The function will search for a volunteer work ID in the (volunteerWorks.txt) file and check if the volunteer has no more than 20 volunteers on the same date.
Priority	2/10
Expected risks	The volunteer work is not recorded in the file.
Preconditions	the screen is going to have a message displaying check if volunteer work is available
Postconditions	The screen is going to have either a message of is available or not available
Dependences	the functions (Register in volunteer work & update numbers of volunteers)
Cost and schedule	2 hours
Technical issues	

10-

Code	p.10
Name	Register in volunteer work
Type	Functional requirement
Criticality	low
Input	Integer carries Id of the volunteer who wants to register ,Integer carries Id of the volunteer work chosen by the volunteer, string carries returned value, and file(volunteersWork.txt) contains volunteer ID, volunteer work id.

Output	Return true if the received values are recorded Return False if the received values aren't recorded
Description	The function will record the volunteer id and volunteer work (received values) in (volunteersWork.txt) file.
Priority	4/10
Expected risks	The received id is null or volunteer work id does not exist.
Preconditions	The received id and volunteer work are not recorded yet in the file (volunteersWork.txt)
Postconditions	The received id and volunteer work are recorded in the file (volunteersWork.txt)
Dependences	
Cost and schedule	2 hours
Technical issues	

11-

Code	p.11
Name	Check the current status of items
Type	Functional requirement
Criticality	low
Input	Item id and file(itemwarehouse.txt) containing item ID, description, quantity, and location.
Output	A string containing the location of the item.

Description	The function will search for the item Id in the itemwarehouse.txt and display its location.
Priority	5/10
Expected risks	none
Preconditions	login
Postconditions	The item status will be displayed on the screen.
Dependences	Item data
Cost and schedule	45 minutes
Technical issues	

12.

Code	p.12
Name	Check if the recipient is currently being helped
Type	Functional requirement
Criticality	Medium
Input	A file (recipient.txt) containing id of recipient and a boolean representing if the person is currently being helped or not. Id of recipient

Output	True or False True if it exists False if it doesn't
Description	The function will search for the id of recipient in the(recipient.txt) file and check the boolean representing if the person is currently being helped or not.
Priority	7/10
Expected risks	Id is with null value
Preconditions	Login
Postconditions	none
Dependences	id of recipient sent to the function
Cost and schedule	45 minutes
Technical issues	

13-

Code	p.13
Name	Report monthly items donated to the organization
Type	Functional requirement
Criticality	medium
Input	A specific month

Output	none
Description	The function will search in the file for a date that contains the same month that the function received and will then display each item name and its corresponding date and description that got donated during the required month
Priority	4/10
Expected risks	Some of the donation details may be null values.
Preconditions	The screen will be blank. login
Postconditions	Item name, date of donation, description, and will be displayed on the screen.
Dependences	The dates of each donation.
Cost and schedule	2 hours
Technical issues	

14.

Code	p.14
Name	Update number of volunteer
Type	Functional Requirement
Criticality	High
Input	Volunteer work that needs to be updated, A file(AllvolunteerWorks.txt) containing id of volunteer work , the number of volunteers, the date of volunteer work

	and string.
Output	Returns true if the number of volunteers is updated in certain volunteer work
Description	The function will search in the file (AllvolunteerWorks.txt) for the ID of volunteer work (received id) then update the number of volunteers.
Priority	6/10
Expected risks	The id of volunteer work is null or does not exist.
Preconditions	The number of volunteers not updated in the file (AllvolunteerWorks.txt)
Postconditions	The number of volunteers is updated in the file(AllvolunteerWorks.txt)
Dependences	Depends on the function (Register in volunteer work)
Cost and schedule	1 hour
Technical Issues	

15-

<i>Code</i>	<i>p.15</i>
<i>Name</i>	Report monthly items donated by the organization
<i>Type</i>	Functional requirement
<i>Criticality</i>	medium

<i>Input</i>	A file (itemdonated.txt) containing item Id and donation date and another file (itemwarehouse.txt) And a required month
<i>Output</i>	none
<i>Description</i>	The function will search for item id in (itemwarehouse.txt) for all items in (itemdonated.txt) with the same month as the required and display the item name and donation date.
<i>Priority</i>	4/10
<i>Expected risks</i>	The user may enter a future date that wasn't recorded yet.
<i>Preconditions</i>	Login
<i>Postconditions</i>	Items' names and date of the donation will be displayed on the screen.
<i>Dependences</i>	The required month
Cost and schedule	2 hours
Technical issues	

16-

Code	p.16
Name	Calculate the total income for each year.
Type	Functional requirement
Criticality	high
Input	A file (cashDonationsIncome.txt) containing

	a date, description, amount of income and an integer representing a specific year.
Output	Returns the sum of cash income during the month.
Description in	The function will search in the file for a date that contains the same month that the function received and will display each integer containing cash income with its corresponding description and date during the specified month. This function will also add all the values of the cash income of the specified month and display them.
Priority	3/10
Expected risks	Some of the cash income may be null values. Some of the date values may be null values. Some of the descriptions may be null values.
Preconditions	login
Postconditions	A string containing all the calculation of the income that got donated during a specific month
Dependences	Cash donation data The required month
Cost and schedule	2 hours
Technical issues	

17-

Code	p.17
Name	Prepare monthly report for cash donation
Type	Functional requirement/
Criticality	high

Input	A file (donation.txt) A required month
Output	none
Description	The function will display the donation amount and donation date, and donation details for all donations during the required month
Priority	5/10
Expected risks	The user may enter a future date that wasn't recorded yet.
Preconditions	login
Postconditions	Donation amount and date of the donation will be displayed on the screen.
Dependences	Donation data
Cost and schedule	1 hour
Technical issues	

18-

Code	p.18
Name	Check if the donations delivered to recipient

Type	Functional requirement
Criticality	high
Input	An Integer carries Id of the donor who wants to check his donation, String carries donation status(empty), A file (donations.txt) containing id of donation, id of donor, donation date , boolean representing if the donation status “delivered or not delivered”.
Output	Returns TRUE if the donation status is “delivered” and return FALSE if donation status is “not delivered”
Description	The function will search in the (donations.txt) file for the received id then return donation status.
Priority	3/10
Expected risks	the received id is null or not registered in the file.
Preconditions	The returned value is empty
Postconditions	A string will contain donation status “delivered or not”
Dependences	Donation data
Cost and schedule	1 hour
Technical issues	

19-

Code	p.19
------	------

Name	Update the current status of items
Type	Functional requirement
Criticality	Medium
Input	String carries return value , integer carries Item id and file(itemwarehouse.txt) containing item ID, description, quantity,and location.
Output	Returns true if the location of the item is updated.
Description	The function will search in the file (itemwarehouse.txt) for the ID item (received id) then update the location of the item.
Priority	5/10
Expected risks	the received id is null or not registered in the file.
Preconditions	login item status is not updated in the file (itemwarehouse.txt)
Postconditions	item status is updated in the file (itemwarehouse.txt)
Dependences	Item data
Cost and schedule	45 minutes
Technical issues	

20-

Code	p.20
Name	Make Donation

Type	Functional Requirement
Criticality	High
Input	Donor id A file(donation_detailst.txt) containing order id,donor id, payment method id, type of donation,amount of donation, if donation is delivered,date of delivery,time of delivery. Another file (donation_order.txt) containing order id,product id, donor id,date of donation
Output	A new record of the donation will be made in both text files
Description	The function will create a new record of the donation in the files (donation_order.txt) and (donation_detailst.txt).
Priority	10/10
Expected risks	That an id of a specific donation(product) is null
Preconditions	The record of the new donation is not made
Postconditions	A new record of the donation is created in the files
Dependences	Depends on the files (donations.txt) which contains id of type of donations
Cost and schedule	1 hour
Technical issues	

21-

Code	p.21
Name	Update cash in ledger

Type	Functional Requirement
Criticality	High
Input	cash A file(ledger.txt) containing donor id , date of transaction,cash.
Output	A new record of the donation will be added in the file(ledger.txt)
Description	The function will create a new record of the transaction containing date of the transaction,cash,donor id in the file (ledge.txt)
Priority	10/10
Expected risks	That the cash received from the function (make donation) is null
Preconditions	The file doesn't contain the record of the new donation is not made
Postconditions	A new record of the donation is created in the files
Dependences	Depends on the function (make donation)
Cost and schedule	1 hour
Technical issues	

22-

Code	p.22
Name	Check training attendance.
Type	Functional requirement
Criticality	Medium
Input	An integer carries Id of the volunteer who wants to take the attendance, A file (volunteerTrainingSchedual.txt) containing volunteer's id, the number of training sessions that attended for a specific volunteer in the same week,the volunteer attendance, and an integer for a specific week.

Output	Returns true in case the ID has been found and attendance was taken.
Description	The function will search in the file (volunteerTrainingSchedual.txt) for the ID of the volunteer who wants to take the attendance (received id) then it is recorded in the file that he attended.
Priority	5/10
Expected risks	the received id is null or not registered in the file.
Preconditions	The attendance will not be updated from “absent” to “attend” in the file.
Postconditions	The attendance will be updated from “absent” to “attend” in the file.
Dependences	Volunteer id
Cost and schedule	1 hour
Technical issues	

4 Interface Requirements

4.1 User Interfaces

The interface will contain a field for entering the email and password, and through the email it will be determined whether the user is a donor or one of the employees of the charity. If he was a donor, the ids of cases that need donation will appear to donor with empty box. A box appears for donor to write inside it the id of the case that have been chosen. When the donor click the bottom under the box then another box will appear for the donor to choose the type of donation whether it is money or an item. If it is money, he must specify the method of payment, either cash or visa.

4.1.1 GUI

Describes the graphical user interface if present. This section should include a set of screen dumps or mock-ups to illustrate user interface features. If the system is menu-driven, a description of all menus and their components should be provided.

<https://app.moqups.com/Fa6MYFrQHIOO6gAVd68wLcm3mrdZOH7S/view/page/ae7d7a91c>



Search

Home

About

Shop

Blog

Your Cart

BASMA W KHATWA



Donate Now

Featured Products



CASE 1



CASE 2

DONATE NOW

Learn more about the situation, see our stories and giving with NADA
quantum, and help us make a difference in the world.

Email Address

Subscribe



HELP POOR FAMILIES
\$500



KIDNEY DIALYSIS
\$200



BUILDING CEILING
\$100



EDUCATION
\$100 ★★★★★

4.1.2 CLI

Not applicable

4.1.3 API

Not applicable

4.1.4 Diagnostics or ROM

Not applicable

4.2 Hardware Interfaces

Not applicable

4.3 Communications Interfaces

Not applicable

4.4 Software Interfaces

Describes any remaining software interfaces not included above.

5 Performance Requirements

The memory and speed of the system should be compatible with the amount of data received by the system.

6 Design Constraints

The system should be simply and safe

6.1 Standards Compliance

Not applicable

6.2 Hardware Limitations

Not applicable

6.3 others as appropriate

7 Other non-functional attributes

Specifies any other particular non-functional attributes required by the system. Examples are provided below.

7.1 Security

The password is made secure by using the hashing technique

7.2 Binary Compatibility

Not applicable

7.3 Reliability

The system should be reliable and therefore perform correctly in the specified time without the need of repair.

7.4 Maintainability

In case of failure in any part of the system, this part should be repaired or restored

7.5 Portability

Not applicable

7.6 Extensibility

Not applicable

7.7 Re-usability

Not applicable

7.8 Application Affinity/Compatibility

Not applicable

7.9 Resource Utilization

Not applicable

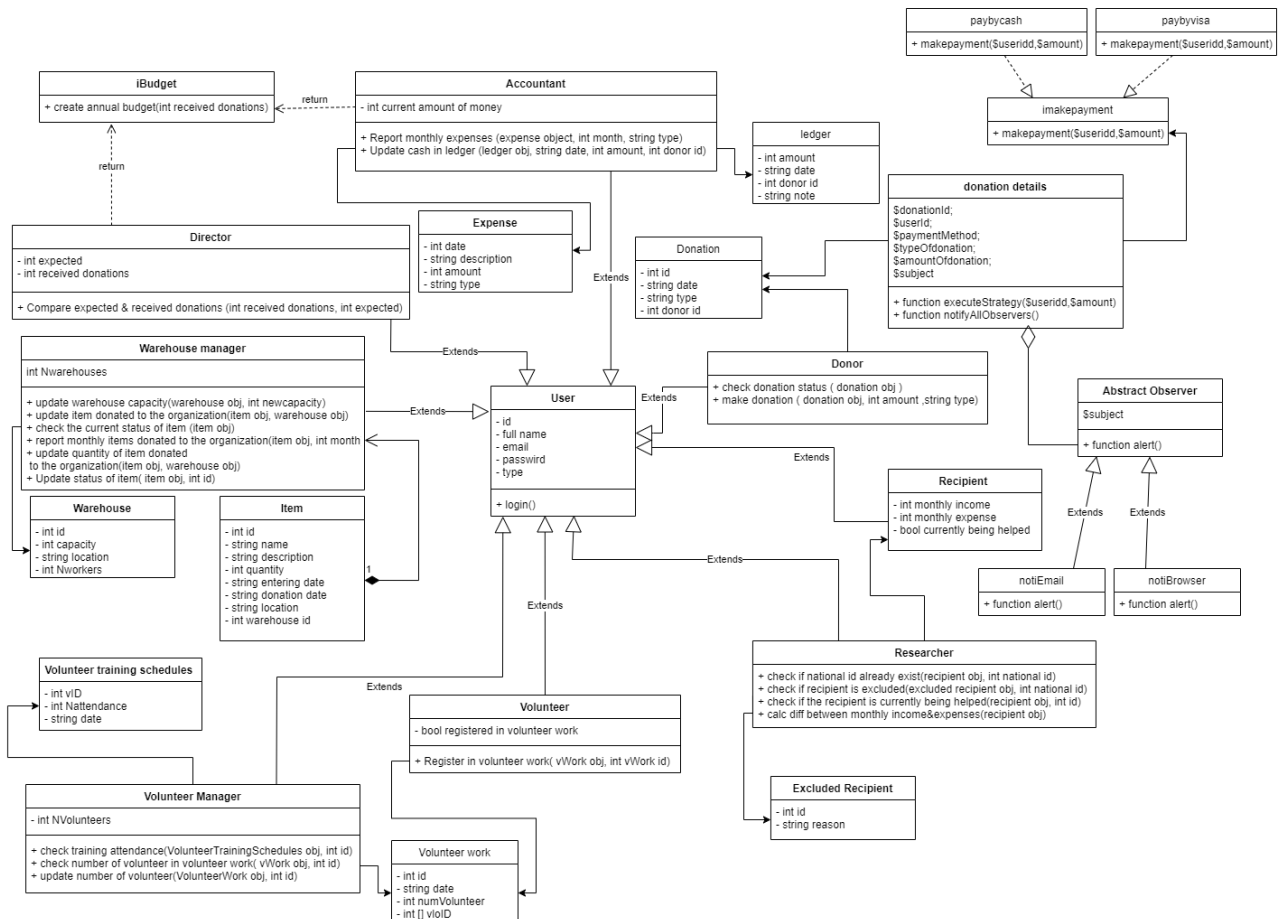
7.10 Serviceability

Not applicable

7.11 others as appropriate

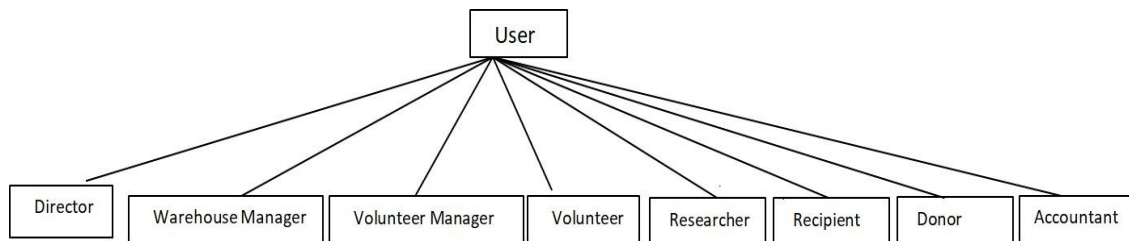
8 Preliminary Object-Oriented Domain Analysis

This section presents a list of the fundamental objects that must be modeled within the system to satisfy its requirements. The purpose is to provide an alternative, "structural" view on the requirements stated above and how they might be satisfied in the system. A primitive class diagram to be delivered.



8.1 Inheritance Relationships

This section should contain a set of graphs that illustrate the primary inheritance hierarchy (is-kind-of) for the system. For example:



8.2 Class descriptions

This section presents a more detailed description of each class identified during the OO Domain Analysis. For more details on the process giving rise to these descriptions, see Lecture 5.3: OO Domain Analysis and/or texts on object-oriented software development. Each class description should conform to the following structure:

8.2.1 Class name

Abstract or Concrete: Indicates whether this class is abstract or concrete.

user: abstract

donation: concrete.

warehouse manager :concrete

item : concrete

Budget: concrete

8.2.2 List of Superclasses:

Names all immediate superclasses.

8.2.3 List of Subclasses:

Names all immediate subclasses.

8.2.4 Purpose:

The purpose of the class user is to reduce repetition of attributes

The purpose of interface classes is to apply polymorphism and implement functions in different ways

8.2.5 Collaborations:

- warehouse manager uses the class warehouse and is aggregating the class item
- Volunteer Manager uses the class volunteer training schedules
- Class volunteer manager and volunteer use the class volunteer work
- Class researcher uses the class Excluded recipient
- Research class use Recipient Class
- Class idonate uses donation
- Donor uses class donation
- Class Researcher implements the class icalc diff
- Class Donor implements iDonate
- Class Director implements iBudget
- Class accountant implements iBudget
- Class Accountant uses class Expenses and class ledger

8.2.6 Attributes:

Lists each attribute (state variable) associated with each instance of this class, and indicates examples of possible values (or a range).

Budget:

Attributes:

- int received donations.

Accountant:

Attributes:

- int expected.

Warehouse manager:

Attributes:

- int Nwarehouses.

Director:

Attributes:

- field: type.

Expense:

Attributes:

- int date.
- string description.
- int amount.
- string type.

User:

Attributes:

- id.
- full name.
- email.
- password.
- type.

Warehouse:

Attributes:

- int id.
- int capacity.
- string location.
- int Nworkers.

Item:

Attributes:

- int id.
- string name.
- string description.
- int quantity.
- string entering date.
- string donation date.
- string location.
- int warehouse id..

Recipient:

Attributes:

- int monthly income.
- int monthly expense.
- bool currently being helped.

Volunteer training schedules:

Attributes:

- int vID.
- int Nattendance.
- string date.

Researcher:

Attributes:

- field: type.

Excluded Recipient:

Attributes:

- int id.
- string reason.

No operations.

Volunteer Manager:

Attributes:

- int NVolunteers.

8.2.7 Operations:

Lists each operation that can be invoked upon instances of this class. For each operation, the arguments (and their type), the return value (and its type), and any side effects of the operation should be specified.

Budget:

Operations:

- + Create annual budget (int received donations).

Accountant:

Operations:

- + Report monthly expenses (expense object, int month, string type).

- + Compare expected & received donations (int received donations, int expected).

Warehouse manager:

Operations:

- + Update warehouse capacity (warehouse obj, int newcapacity).

- + Update item donated to the organization (item obj, warehouse obj).

- + Check the current location of item (item obj).

- + Report monthly items donated to the organization (item obj, int month).

- + Update quantity of item donated to the organization (item obj, warehouse obj).

Director:

Operations:

+ method(type): type.

Expense:

No operations.

User:.

Operations:

+ login().

Warehouse:

No operations.

Item:

No operations.

Recipient:

Operations:

+ method(type): type.

Volunteer training schedules:

No operations.

Researcher:

Operations:

- + Calc diff between monthly income&expenses(recipient obj).

- + Check if national id already exists (recipient obj, int national id).

- + Check if recipient is excluded (excluded recipient obj, int national id).

- + Check if the recipient is currently being helped (recipient obj, int id).

Excluded Recipient:

- string reason.

No operations.

Volunteer Manager:

Operations:

- + Check training attendance (volunteer training schedules, int id

8.2.8 Constraints:

Lists any restrictions upon the general state or behavior of instances of this class

9 Operational use case scenario



10 Preliminary Schedule Adjusted

We started building the system at the end of february.The system will be finished after 3 months.

11 Preliminary Budget Adjusted

There is no budget

12 Appendices

Specifies other useful information for understanding the requirements. All SRS documents should include at least the following two appendices:

12.1 Definitions, Acronyms, Abbreviations

Provides definitions of unfamiliar definitions, terms, and acronyms.

12.2 Collected material

13 References

References

- [1] M. Rehm, N. Bee, and E. Andre, "Wave like an egyptian: accelerometer based gesture recognition for culture specific interactions," in *Proceedings of the 22nd British HCI Group Annual Conference on People and Computers: Culture, Creativity, Interaction - Volume 1*, ser. BCS-HCI '08. Swinton, UK, UK: British Computer Society, 2008, pp. 13–22.