

Software requirement specification document for project Lebaladna

Supervised by: Dr. Marwa Solayman

November 8, 2023

# 1 Introduction

**1.1 Purpose of this document**

the purpose of this document is to use it as a reference when need

## 1.2 Scope of this document

## this system is going to help Lebaladna organization, Lebaladna foundation Objective is to decrease community needs' gaps through creating repeatable effective youth volunteer models within educational contexts that can implement all scales of sustainable development projects. it helps volunteers so they can help the needy people easy.

## 1.3 Overview

## the system will help the volunteers and it will speed up the workflow and reduce the cost and automate some volunteer's tasks

## 1.4 Business Context

# we are doing the project for a charity organization called Lebaladna and their first activity was taken in 2006 and for now, they still using paper document and excel sheets we are building this system for them to help them.

# 2 General Description

## 2.1 Product Functions

Admin:

As an admin, I would like to do a system for Lebaladna, a system that will provide me full access to all of the departments and generate reports.

Scout:

As a scout member, I would like to do Computer-based survey data gathering, a smart database to direct data needed for each department automatically.

survey data gathering, data base to directing to each sector information must know

1: How many family members?

2: how old are the family members?

3: are they working?

4: how much is the salary for each working member?

5: the educational level?

6: what are diseases for each member?

7:how much does the medicine cost?

8: The type of home ownership is (rent or ownership) 9:if rent how much does it cost ?

10: type of home(1, room/2,flat/3,house)

11: what are the materials used for building the home (1: Bricks/2: Concrete/3:wood/4: straw(

12:how many rooms

13: The number of ceilings

14: what they need and how many

**Education:**

**As an education member, I would like to have a database for uneducated children and have a profile for each child and monitor their learning journey from certain criteria.**

**HR:**

**As an HR member, I would like to have a database containing volunteers records and profiles of each volunteer, also I would like to monitor their performance from certain criteria.**

## 2.2 Similar System Information

## Ressala, Egypt's largest volunteer-driven charity organization, engages in a range of activities, from distributing food in slums to visiting orphanages. Ressala has 4 teams which are feeding team, developing team, awareness team, support team while lebaladna have 3 teams HR, research, and education. Ressala takes donation from people but lebaldna does not. [1] A positive example of a survey of Egyptian youth that took account of resource availability and opportunity was also reported in 2010. The Education for Employment, 2010 (e4e) survey sampled 1,500 youth and 1,500 employers including public and private educators, policy makers, and civil society leaders. The aim of the survey was to assess job and educational aspirations of young people in correlation with the educational and employment conditions in which they were living. By design, this survey sought to connect behaviors and attitudes of youth with the resources available to them. There would be little value if youth said they aspired to become physicians yet they had no opportunity for medical education, or to become computer programmers if they had no access to computers. It follows that there is value in the decision to assess youth attitudes alongside available education and employment opportunities.

## 2.3 User Characteristics

## Lebaladna organization manager has the ability to access volunteers and edit them HR team has the ability to access and evaluate volunteers to choose which team to be in education team has the ability to access children and grade them and research team has the ability to count how many families need food and ceilings.

## 2.4 User Problem Statement

## The problem of lebaldna organization is that they use too much time in research’s team to write in a printed template the usage of how many food and ceiling and in team HR that they cant evaluate members to put them in which team and the problem of team education that they cant evaluate children in classes and grading

## 2.5 User Objectives

Lebaladna organization wants online template that count how many families that want food or ceiling in research team and mobile application that evaluate members to see which team is better for each volunteer and another mobile application that evaluate children attendance and exams grading.

## 2.6 General Constraints

Time constraints: the project must be finished in less than 1 month Money constraints: the project is free without money

# 3 Functional Requirements

This section lists the functional requirements in ranked order. Functional requirements describes the possible effects of a software system, in other words, what the system must accomplish. Other kinds of requirements (such as interface requirements, performance requirements, or reliability requirements) describe how the system accomplishes its functional requirements see table 1. Each functional requirement should be specified in a format similar to the following:

Table 1: Functional Requirement XYZ

|  |  |
| --- | --- |
| Function Name | accessHr |
| Description | allows the HR to access the volunteer |
| Critically | High |
| Technical issues | Wrong name, blank inputs |
| Cost and schedule | 1h |
| Risks | - |
| Dependencies with other requirements | - |
| Pre-Condition | Home page |
| Post-Condition | Volunteers page |

|  |  |
| --- | --- |
| Function Name | modifyHr |
| Description | allows the HR to modify the volunteer |
| Critically | meduim |
| Technical issues | Wrong name, |
| Cost and schedule | 1h |
| Risks | - |
| Dependencies with other requirements | - |
| Pre-Condition | Home page |
| Post-Condition | Modifying page |

|  |  |
| --- | --- |
| Function Name | addHr |
| Description | allows the HR to Add a volunteer |
| Critically | Critical |
| Technical issues | Wrong name, |
| Cost and schedule | 2h |
| Risks | - |
| Dependencies with other requirements | - |
| Pre-Condition | Home page |
| Post-Condition | Modifying page |

|  |  |
| --- | --- |
| Function Name | deleteHr |
| Description | allows the HR to delete a volunteer |
| Critically | meduim |
| Technical issues | Wrong name, |
| Cost and schedule | 2h |
| Risks | - |
| Dependencies with other requirements | - |
| Pre-Condition | Home page |
| Post-Condition | Modifying page |

|  |  |
| --- | --- |
| Function Name | accessEducation |
| Description | Can access the student |
| Critically | High |
| Technical issues | Wrong name,blank inputs |
| Cost and schedule | 1h |
| Risks | - |
| Dependencies with other requirements | - |
| Pre-Condition | Home page |
| Post-Condition | access page |

|  |  |
| --- | --- |
| Function Name | modifyEducation |
| Description | Can modify the student |
| Critically | High |
| Technical issues | Wrong name,blank inputs |
| Cost and schedule | 1h |
| Risks | - |
| Dependencies with other requirements | - |
| Pre-Condition | Home page |
| Post-Condition | access page |

|  |  |
| --- | --- |
| Function Name | addEducation |
| Description | Can add a student |
| Critically | critcal |
| Technical issues | Wrong name,blank inputs |
| Cost and schedule | 1h |
| Risks | - |
| Dependencies with other requirements | - |
| Pre-Condition | Home page |
| Post-Condition | access page |

|  |  |
| --- | --- |
| Function Name | deleteEducation |
| Description | Can delete a student |
| Critically | Meduim |
| Technical issues | Wrong name,blank inputs |
| Cost and schedule | 1h |
| Risks | - |
| Dependencies with other requirements | - |
| Pre-Condition | Home page |
| Post-Condition | access page |

Scout: AddHousingUnits

|  |  |
| --- | --- |
| Function Name | AddHousingUnits |
| Description | Allow the scout to add a new house. |
| Critically | Critical |
| Technical issues | Blank inputs |
| Cost and schedule | 10 minutes |
| Risks | - |
| Dependencies with other requirements | - |
| Pre-Condition | Can’t add new house |
| Post-Condition | New house is added |

Scout: ModifyHousingUnites

|  |  |
| --- | --- |
| Function Name | ModifyHousingUnites |
| Description | Allow scout to modify on the House added before |
| Critically | high |
| Technical issues | - |
| Cost and schedule | 20 minutes |
| Risks | If the scout used the modify function before add an house before, an condition could be checked before allowing modify function |
| Dependencies with other requirements | AddHousingUnites function |
| Pre-Condition | The house unites was unmodifiable |
| Post-Condition | the house unites can be modified |

Scout: Access

|  |  |
| --- | --- |
| Function Name | Access |
| Description | Allow scout to Access the Houses |
| Critically | high |
| Technical issues | - |
| Cost and schedule | 10 minutes |
| Risks | If the scout used the access function before add an house before, an condition could be checked before allowing access function |
| Dependencies with other requirements | AddHousingUnites function |
| Pre-Condition | The house unites was inaccessible |
| Post-Condition | the house unites can be accessed |

Scout: Delete

|  |  |
| --- | --- |
| Function Name | Delete |
| Description | Allow scout to delete on the House added before |
| Critically | high |
| Technical issues | - |
| Cost and schedule | 20 minutes |
| Risks | If the scout used the delete function before add an house before, an condition could be checked before allowing delete function |
| Dependencies with other requirements | AddHouseUnits function |
| Pre-Condition | The house unites was undeletable |
| Post-Condition | the house unites can be deleted |

HousingUnites: FoodContainerCount

|  |  |
| --- | --- |
| Function Name | FoodContainerCount |
| Description | This function calculate the number of food container count needed according to the count of people in the house |
| Critically | Medium |
| Technical issues | Blank input if people count |
| Cost and schedule | 10 minutes |
| Risks | - |
| Dependencies with other requirements | People count attribute |
| Pre-Condition | Could not determine the right number of food containers needed |
| Post-Condition | The number of food containers |

HousingUnites: IftarMealCount

|  |  |
| --- | --- |
| Function Name | IftarMealCount |
| Description | This function calculate the number of iftar meals count needed according to the count of people in the house |
| Critically | Medium |
| Technical issues | Blank input if people count |
| Cost and schedule | 10 minutes |
| Risks | - |
| Dependencies with other requirements | People count attribute |
| Pre-Condition | Could not determine the right number of iftar meal needed |
| Post-Condition | The number of iftar meal |

Educational: EducationalSkills

|  |  |
| --- | --- |
| Function Name | EducationalSkills |
| Description | This function returns the grade of student’s overall educational skills according to the weight of each skill and the grade of each one |
| Critically | High |
| Technical issues | Blank inputs |
| Cost and schedule | 10 minutes |
| Risks | - |
| Dependencies with other requirements | All the educational skills attributes |
| Pre-Condition | No grade was calculated |
| Post-Condition | Return grade of all the educational skills attributes |

Topics : TopicsSkills

|  |  |
| --- | --- |
| Function Name | TopicsSkills |
| Description | This function returns the grade of student’s overall topics skills according to the weight of each skill and the grade of each one |
| Critically | High |
| Technical issues | Blank inputs |
| Cost and schedule | 10 minutes |
| Risks | - |
| Dependencies with other requirements | All the educational skills attributes |
| Pre-Condition | No grade was calculated |
| Post-Condition | Return grade of all the topics skills attributes |

Commitment: CommitmentGrading

|  |  |
| --- | --- |
| Function Name | CommitmentGrading |
| Description | This function returns the grade of student’s commitment |
| Critically | High |
| Technical issues | - |
| Cost and schedule | 10 minutes |
| Risks | -. |
| Dependencies with other requirements | All the commitment attributes |
| Pre-Condition | No grade was calculated |
| Post-Condition | Return grade of all the commitment attributes |

QualitativeSkills : QualitativeSkillGrade

|  |  |
| --- | --- |
| Function Name | QualitativeSkillGrade |
| Description | This function returns the grade of volunteers qualitative skills |
| Critically | high |
| Technical issues | Blank inputs of skills |
| Cost and schedule | 10 minutes |
| Risks | - |
| Dependencies with other requirements | All the qualitative skills attributes |
| Pre-Condition | No grade was calculated |
| Post-Condition | Return grade of all the qualitative skills attributes |

QuantitativeSkills : QuantitativeSkillsGrade

|  |  |
| --- | --- |
| Function Name | QuanitaiveSkillsGrade |
| Description | This function returns the grade of volunteers quantitative skills |
| Critically | High |
| Technical issues | Blank inputs of skills |
| Cost and schedule | 10 minutes |
| Risks | - |
| Dependencies with other requirements | All the quantitative skills attributes |
| Pre-Condition | No grade was calculated |
| Post-Condition | Return grade of all the qualitative skills attributes |

Lebaldana manger: accessTheHR

|  |  |
| --- | --- |
| Function Name | accessTheHR |
| Description | Allow the manager to access all the HR functions and attributes. |
| Critically | Critical |
| Technical issues | - |
| Cost and schedule | 10 minutes |
| Risks | - |
| Dependencies with other requirements | - |
| Pre-Condition | Manager can't access the HR system. |
| Post-Condition | Manager can access the HR system. |

Lebaldana manger: accessTheScout

|  |  |
| --- | --- |
| Function Name | accessTheScout |
| Description | Allow the manager to access all the Scout functions and attributes. |
| Critically | Critical |
| Technical issues | - |
| Cost and schedule | 10 minutes |
| Risks | - |
| Dependencies with other requirements | - |
| Pre-Condition | Manager can't access the Scout system. |
| Post-Condition | Manager can access the Scout system. |

Lebaldana manger: accessTheEducation

|  |  |
| --- | --- |
| Function Name | accessTheEducation |
| Description | Allow the manager to access all the Education functions and attributes. |
| Critically | Critical |
| Technical issues | - |
| Cost and schedule | 10 minutes |
| Risks | - |
| Dependencies with other requirements | - |
| Pre-Condition | Manager can't access the Education system. |
| Post-Condition | Manager can access the Education system. |

# 4 Interface Requirements

This section describes how the software interfaces with other software products or users for input or output. Examples of such interfaces include library routines, token streams, shared memory, data streams, and so forth.

## 4.1 User Interfaces

Use some software for primitive plan of your project. Describes how this product interfaces with the user.

**4.1.1 GUI**

Diagram

Description automatically generated

**4.1.2 CLI**

Describes the command-line interface if present. For each command, a description of all arguments and example values and invocations should be provided.

**4.1.3 API**

Describes the application programming interface, if present. For each public interface function, the name, arguments, return values, examples of invocation, and interactions with other functions should be provided.

**4.1.4 Diagnostics or ROM**

Describes how to obtain debugging information or other diagnostic data.

**4.2 Hardware Interfaces**

Mobile device / pc/ laptop

**4.3 Communications Interfaces**

Internet

**4.4 Software Interfaces**

Describes any remaining software interfaces not included above.

**5 Performance Requirements**

For pc Minimum 4gb RAM and i5 CPU

# For mobile minimum 4gb RAM and latest version of its software

# 6 Design Constraints

Specifies any constraints for the design team using this document.

**6.1 Standards Compliance**

User limt:250-750

**6.2 Hardware Limitations**

if one of the requirements is not met, it would definitely lead to a huge overload which would cause the system to stutter and lag, it would also lower it's speed which leads to unpleasant results.

**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 Reliability**

The system can be relied upon as long as the number of users does not exceed 750

**7.2 Portability**

the system is Portable, can be used for mobile app and web app

**7.3 scalability**

If the number of users exceeds more than 750, it will cause the system to fail, in which case the system will need to be upgraded

# 8 Preliminary Object-Oriented Domain Analysis

## Diagram Description automatically generated8.1 Inheritance Relationships

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

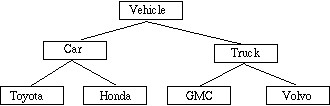
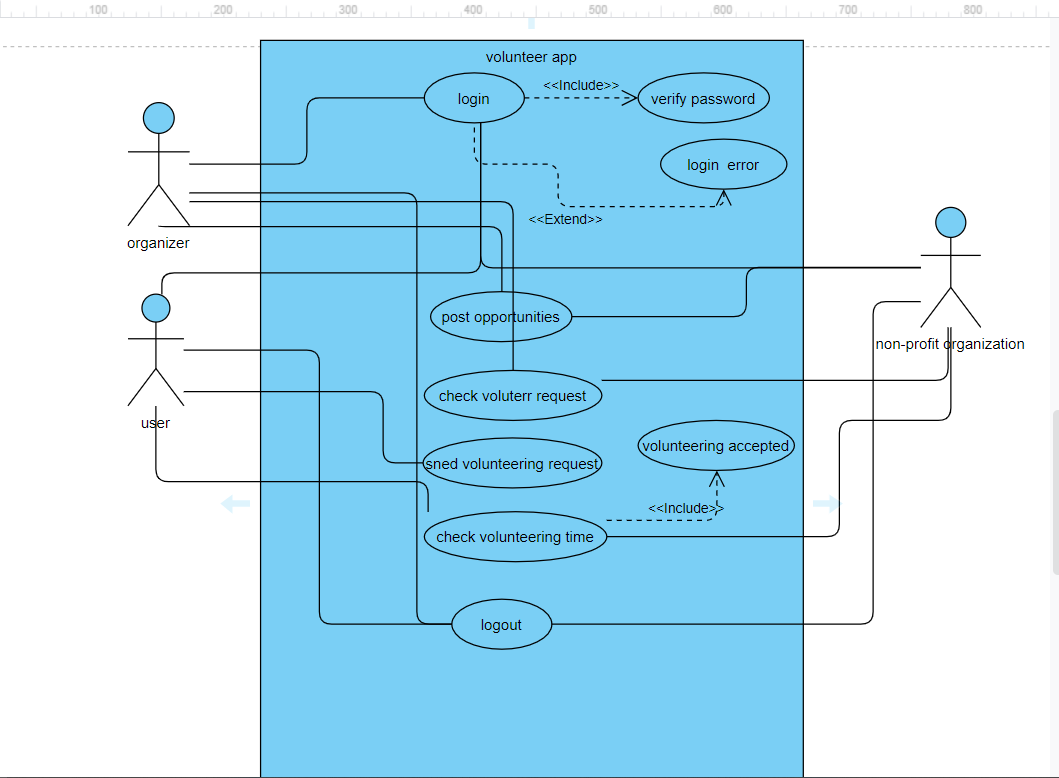
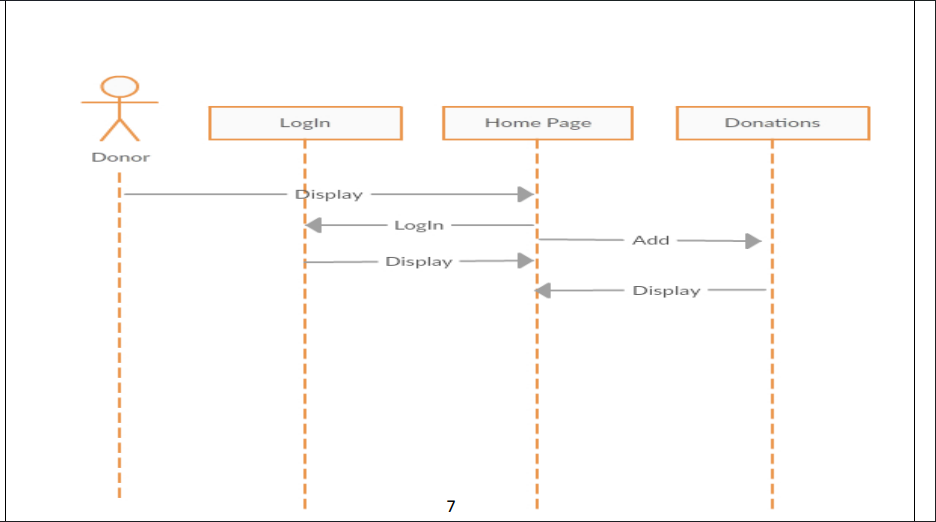
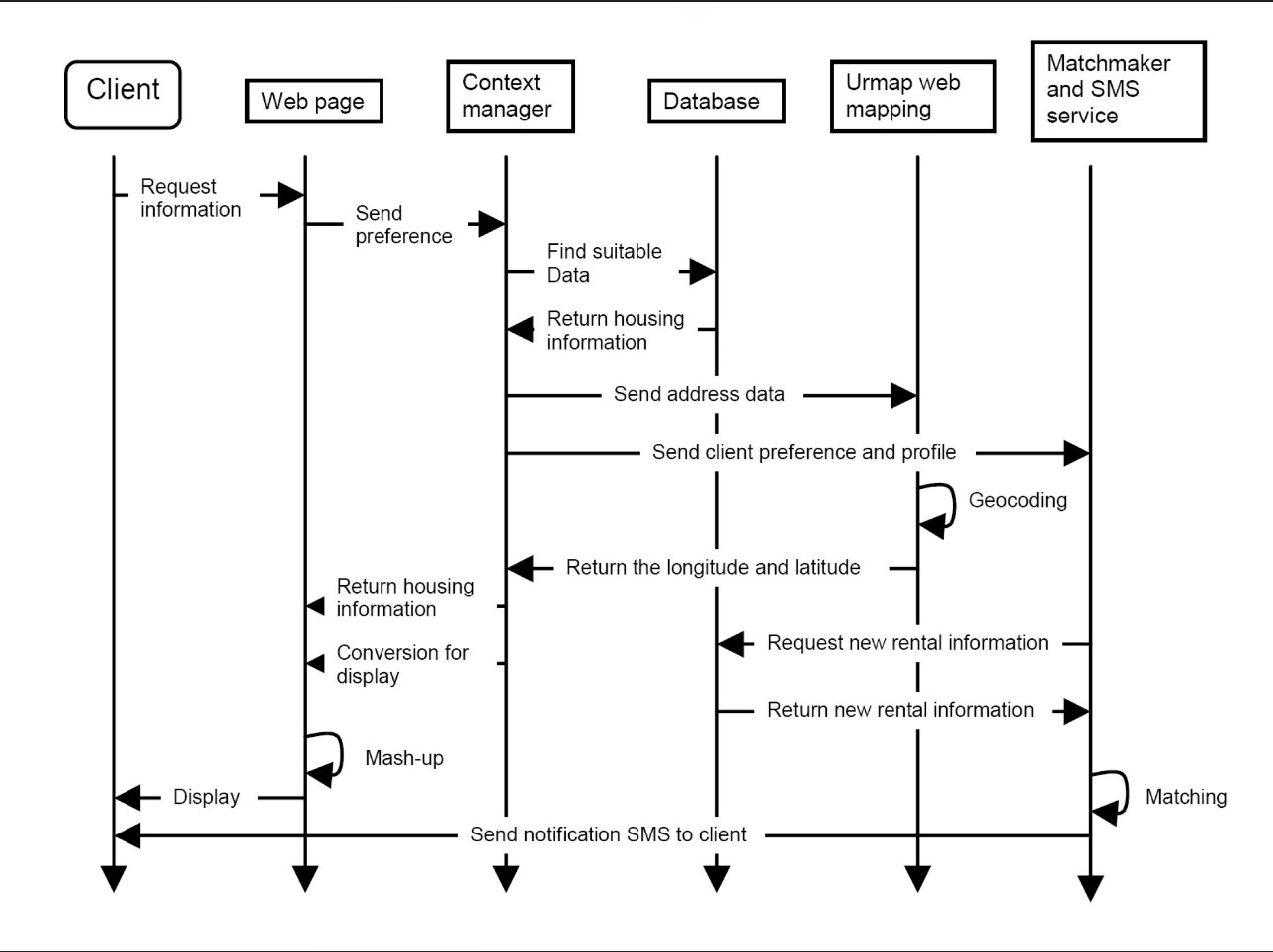


Figure 1: Inheritance Relations







## 8.2 Class descriptions

This class is a master class that manages all the systems and has access to all the data that belong to other main classes such as HR, Scout and Education.

**8.2.1 Class name**

lebaladna manger

**8.2.2 List of Superclasses:**

.

**8.2.3 List of Subclasses:**

HR , Scout and Education.

**8.2.4 Purpose:**

Give the manager access to all main classes.

**8.2.5 Collaborations:**

**8.2.6 Attributes:**

Object called x of HR class, object called y of Scout class, object called z of Education class.

**8.2.7 Operations**

access HR: void function that does not return anything but gives the manager full access to the HR department.

access Scout: void function that does not return anything but gives the manager full access to the Scout department.

access Education: void function that does not return anything but gives the manager full access to the Education department.

**8.2.8 Constraints:**

## 8.2 Class descriptions

This class has all the volunteer attributes.

**8.2.1 Class name**

volunteer.

**8.2.2 List of Superclasses:**

Names all immediate superclasses.

**8.2.3 List of Subclasses:**

Names all immediate subclasses.

**8.2.4 Purpose:**

States the basic purpose of the class.

**8.2.5 Collaborations:**

Names each class with which this class must interact in order to accomplish its purpose, and how.

**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).

**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.

**8.2.8 Constraints:**

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

## 8.2 Class descriptions

This class manages the HR department.

**8.2.1 Class name**

HR.

**8.2.2 List of Superclasses:**

lebaladna manger

**8.2.3 List of Subclasses:**

volunteer

**8.2.4 Purpose:**

Manage and monitor all the volunteers.

**8.2.5 Collaborations:**

Names each class with which this class must interact in order to accomplish its purpose, and how.

**8.2.6 Attributes:**

X is an object of volunteer class.

Volunteers\_list is a list of all the volunteers.

**8.2.7 Operations**

AccessHR: this function takes the x which is the object of the volunteer class and takes the list of volunteers, to access the volunteer's data.

modifyHR: this function takes the x which is the object of the volunteer class and takes the list of volunteers, to modify the volunteer’s data.

addHR: this function takes the x which is the object of the volunteer class and takes the list of volunteers, to create and add a new volunteer to the list.

deleteHR: this function takes the x which is the object of the volunteer class and takes the list of volunteers, to delete a volunteer from the list.

**8.2.8 Constraints:**

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

## 8.2 Class descriptions

This class manages the education department.

**8.2.1 Class name**

education

**8.2.2 List of Superclasses:**

lebaladna manger

**8.2.3 List of Subclasses:**

uneducated child

**8.2.4 Purpose:**

Manage and monitor all the uneducated children.

**8.2.5 Collaborations:**

Names each class with which this class must interact in order to accomplish its purpose, and how.

**8.2.6 Attributes:**

X is an object of an uneducated child class.

students\_List is a list of all the students.

**8.2.7 Operations**

accessEducation: this function takes the x which is the object of the uneducated child class and takes the list of students\_List, to access the student’s data.

modifyEducation: this function takes the x which is the object of the uneducated child class and takes the list of students\_List, to modify the student’s data.

addEducation: this function takes the x which is the object of the uneducated child class and takes the list of students\_List, to create and add a new student to the students\_list.

**8.2.8 Constraints:**

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

## 8.2 Class descriptions

This class manages the scout department.

**8.2.1 Class name**

scout

**8.2.2 List of Superclasses:**

lebaladna manger

**8.2.3 List of Subclasses:**

housing units.

**8.2.4 Purpose:**

Manage all the housing units.

**8.2.5 Collaborations:**

Names each class with which this class must interact in order to accomplish its purpose, and how.

**8.2.6 Attributes:**

string address, int flatt number, string house build material, int room count, string home ownership, int ceiling count in need, int persone count, int blankets needed,string water connection

**8.2.7 Operations**

foodContainersCount: this function takes the int persone count, to calculate the food container count needed.

iftarMealCount: this function takes the int persone count, to calculate the iftar meal count needed.

**8.2.8 Constraints:**

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

## 

## 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**

volunteer's qualitative skills

**8.2.2 List of Superclasses:**

volunteer

**8.2.3 List of Subclasses:**

-

**8.2.4 Purpose:**

It collects the volunteer's qualitative skills

**8.2.5 Collaborations:**

-

**8.2.6 Attributes:**

**int leadership-int teamwork-int initiative-int planning-int recruiting-int training-int mangeing-int evaluating-int presentaion**

**8.2.7 Operations**

+qualitativeSkills (int leadership, int teamwork, int initiative, int planning, int recruting, int training, int mangeing, int evaluting, int presentaion)

**8.2.8 Constraints:**

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

## 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**

topics

**8.2.2 List of Superclasses:**

uneduacted child

**8.2.3 List of Subclasses:**

-

**8.2.4 Purpose:**

**Collect the exam grading and homework and activities participation**

**8.2.5 Collaborations:**

-

**8.2.6 Attributes:**

**char exams grading-int homework submission-int activities participation**

**8.2.7 Operations**

+topicsSkills(char exams grading, int homework submission, int activities participation)

**8.2.8 Constraints:**

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

## 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**

educational

**8.2.2 List of Superclasses:**

uneduacted child

**8.2.3 List of Subclasses:**

-

**8.2.4 Purpose:**

**Check if the uneducated child has some skills**

**8.2.5 Collaborations:**

-

**8.2.6 Attributes:**

**int presentation-int communication-int leadership-int initiative**

**int teamwork-int involvement-int sharing**

**8.2.7 Operations**

+educationalSkills(int presentaion, int communication, int leadership, int initiative, int teamwork, int involvement, int sharing)

**8.2.8 Constraints:**

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

## 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**

houssing units

**8.2.2 List of Superclasses:**

scout

**8.2.3 List of Subclasses:**

-persone in need

**8.2.4 Purpose:**

**Collects the data from scout**

**8.2.5 Collaborations:**

-

**8.2.6 Attributes:**

**string address-int falt number-string house bulid material-int room count**

**string home ownership-int cealing count in need-int persone count**

**int blankets needed-string water connection**

**8.2.7 Operations**

+foodContainersCount(int persone count)

+iftarMealCount(int persone count)

**8.2.8 Constraints:**

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

## 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**

volunteer's quantitative skills

**8.2.2 List of Superclasses:**

**Volunteer**

**8.2.3 List of Subclasses:**

**8.2.4 Purpose:**

**Collects the data from scout**

**8.2.5 Collaborations:**

-

**8.2.6 Attributes:**

**int onground attendece-int meetings attendece**

**8.2.7 Operations**

+quantitativeSkills(int onground attendece, int meetings attendece)

**8.2.8 Constraints:**

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

# 9 Operational Scenarios

# the volunteer use the app or the website to log in after login he/she has to select the team if he/she chooses HR a new web page opens with his/her information and a button (edit) that allows the HR team member to edit the volunteer's information and button (add new) that allows the HR member to add new volunteer if he/she chooses Education a new web page opens with the information of the uneducated child and a button (edit) that allows the education team member to edit the information and button (add new) that allows the education member to add new uneducated child if he/she chooses the scout a new page opens and the page contains questions that the scout team member have to ask it for the in-need persons

# 10 Preliminary Schedule Adjusted

Timeline

Description automatically generated

**11 Preliminary Budget Adjusted**

**Spreadsheet: United Consolidated Metropolitan Health Agency (UCMHA)**

|  |  |
| --- | --- |
| **Salaries for developers** | 16,000 |
| **Supplies** | 300 |
| **Equipment** | 1,500 |
| **Phone** | 400 |
| **Insurance** | 800 |
| **Rent** | 4,000 |
| **Totals** | 23,000 |

# 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**

HR ->human resources

CPU-> central processing unit

RAM-> Random Access Memory

**12.2 Collected material**

Table

Description automatically generated with low confidence

Application, table

Description automatically generated

Table

Description automatically generated

**13 References**

# References

[1] JOHN D. GERHART CENTER. (2012). Takaful 2012 Second Annual Conference on Arab Philanthropy and Civic Engagement. *Resala - a Message about Giving ’ : Youth, Voluntarism and an Emerging Imaginary of Egypt*. Published. https://d1wqtxts1xzle7.cloudfront.net/30884272/Takaful\_2012\_Volume-with-cover-page.pdf?Expires=1623196067&Signature=IZYk7jP1Jd-SopWivrTO5udfHiboW7TV7NFwzOGc8U-oCs3GxSSQ9qkSmarpKf-I2xns2frLM1tUrbzD4fKkh1evXpdCIeqVoSECWNHr2hEGyr~mjXi4cVAv-r4bLjPIea6wDH8ZeQ-7GK5KKRxQY31Xf3jbVfz-KVBxez-kyOEhsc0C9mi1mh59JA8y0Qr5wRBLkEJzgftE2eliiShzttAs4MVW~nFqChTIWHGhoM5wKfpusakG~kxWtka9ik6YjSfazlUTncXlXNQMwtioxZ3MVOVESWUQL-Nf08~UiwZr0sP7-B4Wfz-HfUtRBvt5wqBW97uojz5PiHqKbZPJqg\_\_&Key-Pair-Id=APKAJLOHF5GGSLRBV4ZA#page=114

**Database**

**Diagram

Description automatically generated**