



Palestine Polytechnic University  
College of Information Technology and Computer Engineering  
Information Technology Department

## Graduation Project: Charities and Volunteers System

Project Team:

Razan M Taha  
Sally Y Sharawneh  
Wala' A Shawar

Supervised By:

Dr. Mousa Erfa'yah

This project is to complement the graduate attributes of bachelor degree for information technology field of study

May 2016

## Dedication

To our parent who Spent nights and days doing their best To give us the best ...

To all student and who Wish to look for The future...

To who love the knowledge and Looking for the new In this world ...

To our beloved country Palestine ...

To all of our friends ...

To our loves ...

# **Acknowledgement**

First of all, we would like to thank our God for the help to complete this work successfully, and we would like to acknowledge and thank Palestine polytechnic university for the effort they had done in order to facilitate our work.

We would like to thank our supervisor Dr. Mousa Erfa'yah for this help and continuous encouragement to the team work.

Our project is a result of hard work and participation among us “project team” and our supervisors who gave us all there energy and experience in order to pass this project.

Finally, we would like to thank our families because of their support and encouragement through our study time, then we want to thank our friends and anyone help us to finish this work.

## **Abstract**

Charities play an important role in building and improving society, as its aimed to provide help for a certain category In the society, but it face many problems and difficulties through achieving its objectives. one of these problems is having a difficulties in communication with the other associations and volunteers to share its events and activities. Also there is an important role for the volunteers in building and improving the society through their services and volunteer works which is considered supportive for charities, but volunteer's efforts are scattered and there is no attention for it. Therefore, developers found that it is necessary to build a network which is gather charities and volunteers in one place, so that charities would be able to share its activates with other charities and volunteers, the volunteer also will be able to know about the charities activities and events in the suitable time to involve to it.

The Developers built a website and application for smart phones, which charities can create an account and add its objectives and information and begin communicating, sharing events and activities with other charities and volunteers. Moreover, volunteer can create an account that enables him to join charities and convoy latest activities and events by getting notification about charities activities and events shared on the website. Finally, volunteer can rate and donate for the charities which support it financially.

## المخلص

تلعب الجمعيات الخيرية دوراً كبيراً في بناء المجتمع وتطوره، حيث ان هذه الجمعيات تهدف الى تقديم المساعدة لفئة معينة من المجتمع مثل الايتام وسيدات المجتمع وغيرها، لكن هذه الجمعيات تواجه بعض المشكلات والصعوبات تحول دون تحقيق الهدف الذي تطمح اليه. ومن هذه المشكلات التي تقف عقبة أمامها، صعوبة التواصل مع الجمعيات الاخرى والمتطوعين، لمشاركة الاحداث والنشاطات التي تقوم بها.

كما أن للمتطوعين دور كبير في بناء المجتمع وتطوره، من خلال ما يقدمونه من خدمات واعمال تطوعية التي تعتبر داعمة ومساندة للجمعيات الخيرية، لكن جهود هؤلاء المتطوعين مبعثرة نوعا ما ولا يوجد اي تركيز عليها.

لذلك وجد ان من الضروري بناء شبكة تجمع بين الجمعيات الخيرية والمتطوعين في مكان واحد لتتمكن هذه الجمعيات من مشاركة الاحداث والنشاطات التي تقوم بها مع غيرها من الجمعيات الخيرية والمتطوعين بسهولة، وكذلك يتمكن المتطوع من معرفة نشاطات واحداث الجمعيات الخيرية بسهولة وفي الوقت المناسب ليتمكن من المشاركة وتقديم الاعمال التطوعية.

قام فريق العمل ببناء موقع و تطبيق للهواتف الذكية ، تستطيع من خلاله الجمعيات الخيرية انشاء حساب في الموقع واطافة معلوماتها واهدافها، والبدء من خلال الحساب بالتواصل مع غيرها من الجمعيات الخيرية بالاطافة الى الاعلان عن النشاطات التي تقوم بها والاحداث، لكي يتمكن المتطوعون من المشاركة بها والتفاعل معها ، كما أن المتطوع يستطيع من خلال الموقع انشاء حساب يمكنه من الانضمام للجمعيات الخيرية، للبقاء على معرفة باخر النشاطات والاحداث التي تقوم بها الجمعيات التي انضم اليها ، والمشاركة بهذه الاحداث من خلال الحصول على اشعارات عن احداث الجمعيات التي تقوم هذه الاخيرة بمشاركتها في الموقع، كذلك يستطيع المتطوع تقييم الجمعية الخيرية والتبرع للجمعيات الخيرية من خلال الموقع والذي يساهم في دعم تمويل هذه الجمعيات.

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Introduction. . . . .	5
1.2	Problem Statement. . . . .	5
1.2.1	Problems related to volunteers. . . . .	5
1.2.2	Problems Related to Charities. . . . .	5
1.3	Goals and Objectives. . . . .	6
1.4	Project Scope. . . . .	6
1.5	Project Methodology. . . . .	6
1.5.1	MVC Design Pattern. . . . .	6
1.5.2	Why MVC ? . . . . .	6
1.5.3	The Directory Structure. . . . .	7
1.5.4	MVC Sequence Diagram. . . . .	8
1.6	Tasks. . . . .	9
1.7	Times tables. . . . .	10
1.7.1	First Semester. . . . .	10
1.7.2	Second Semester. . . . .	10
<b>2</b>	<b>Planning</b>	<b>11</b>
2.1	Introduction. . . . .	11
2.2	System Actors. . . . .	11
2.3	System Requirements. . . . .	11
2.3.1	Functional Requirements. . . . .	11
2.3.2	Non-functional Requirements. . . . .	13
2.4	Constrains. . . . .	13
2.5	Risks Analysis and Solutions. . . . .	13
<b>3</b>	<b>Analysis and Design</b>	<b>14</b>
3.1	Introduction. . . . .	14
3.2	Requirements Analysis Tables. . . . .	14
3.2.1	Administrator Requirements. . . . .	14
3.2.2	Charities Requirements. . . . .	15
3.2.3	Volunteers Requirements. . . . .	17
3.2.4	Charities and Volunteers Requirements. . . . .	18
3.3	Use Case Diagram. . . . .	22
3.4	System Database. . . . .	23
3.5	Database schema. . . . .	28
<b>4</b>	<b>Software Implementation.</b>	<b>29</b>
4.1	Introduction. . . . .	29
4.2	Implementation tools. . . . .	29
4.2.1	WAMP Server. . . . .	29

4.2.2	NetBeans. . . . .	29
4.2.3	Android Studio. . . . .	29
4.2.4	Bootstrap Framework. . . . .	29
4.3	Website Interfaces. . . . .	30
4.3.1	Administrator Control Panel. . . . .	30
4.3.2	Index Page. . . . .	30
4.3.3	User Homepage. . . . .	31
4.3.4	Charity Profile. . . . .	32
4.3.5	Volunteer Profile. . . . .	32
4.3.6	Mobile Friendly Interfaces. . . . .	33
4.4	Mobile Application Interfaces. . . . .	35
4.4.1	Login Screen. . . . .	35
4.4.2	Home Screen. . . . .	35
4.4.3	New Event Screen. . . . .	36
4.4.4	New Post Screen. . . . .	36
<b>5</b>	<b>Testing</b>	<b>37</b>
5.1	Introduction. . . . .	37
5.2	Website Unit Testing. . . . .	37
5.3	Mobile Application Unit Testing. . . . .	39
5.4	Interfaces for Units Testing. . . . .	40
5.5	Integration Testing. . . . .	45
<b>6</b>	<b>Results and Recommendation</b>	<b>46</b>
6.1	Introduction. . . . .	46
6.2	Project Results. . . . .	46
6.3	Future Recommendation. . . . .	46

# List of Tables

3.1	Search for user. . . . .	14
3.2	Block volunteers or charities. . . . .	14
3.3	Approve charities. . . . .	15
3.4	Create account for charities. . . . .	15
3.6	Add new posts. . . . .	15
3.5	Add branch . . . . .	16
3.7	Delete posts. . . . .	16
3.8	Create events. . . . .	16
3.9	Cancel events. . . . .	17
3.10	Create new account for volunteer. . . . .	17
3.11	Review charities . . . . .	18
3.12	Donate to charities. . . . .	18
3.13	Search for volunteers and charities accounts. . . . .	18
3.14	Get notification about charity activity. . . . .	19
3.15	Follow volunteers or charities . . . . .	19
3.16	Add comment on charities posts. . . . .	19
3.17	Delete comment. . . . .	20
3.18	Report volunteers or charities. . . . .	20
3.19	Join events. . . . .	20
3.20	Search for events. . . . .	21
3.21	Administrator table. . . . .	23
3.22	Users table. . . . .	23
3.23	Charity table. . . . .	24
3.24	Events tables. . . . .	24
3.25	Posts table. . . . .	24
3.26	Activities table. . . . .	25
3.27	Comments table. . . . .	25
3.28	Notifications table. . . . .	25
3.29	Ratings table. . . . .	25
3.30	Joined events table. . . . .	26
3.31	Followers table. . . . .	26
3.32	Donation table. . . . .	26
3.33	Reports table. . . . .	26
3.34	Phones table. . . . .	27
3.35	Photos table. . . . .	27
5.1	Website components report . . . . .	37
5.2	Mobile application components report . . . . .	39



# List of Figures

1.1	Model, view , controller transaction . . . . .	7
1.2	MVC sequence diagram . . . . .	8
1.3	First semester time table. . . . .	10
1.4	Second semester time table. . . . .	10
3.1	Use case diagram . . . . .	22
3.2	Database Schema . . . . .	28
4.1	Administrator control panel . . . . .	30
4.2	Index page . . . . .	30
4.3	Arabic index page . . . . .	31
4.4	User Homepage. . . . .	31
4.5	Charity profile. . . . .	32
4.6	Volunteer profile. . . . .	32
4.7	Index page . . . . .	33
4.8	Home page . . . . .	33
4.9	Profile page . . . . .	34
4.10	Signin screen. . . . .	35
4.11	Home screen. . . . .	35
4.12	New event screen. . . . .	36
4.13	New Post screen. . . . .	36
5.1	Username or password in correct . . . . .	40
5.2	Blocked user . . . . .	40
5.3	Successful message after creating account . . . . .	41
5.4	Valid information . . . . .	41
5.5	Required fields error . . . . .	42
5.6	Password characters error . . . . .	42
5.7	Value in confirm password field error . . . . .	42
5.8	Search for charity. . . . .	43
5.9	Search results. . . . .	43
5.10	Charity profile. . . . .	44
5.11	New Event. . . . .	44
5.12	Success message. . . . .	45
5.13	Get notification about charity event. . . . .	45

# Chapter 1

## Introduction

### 1.1 Introduction.

Charities and volunteers play an important role in serving a specific group of people in the society. Many charities have their own website or mobile application to share their activities, also there are many websites and mobile applications for volunteers. Because they're scattered around, it's hard for people to reach them and get the best out of these charities, it's also hard for charities and volunteers to communicate with each other. As developers, we decided to make one network that connects all these charities and volunteers in one place. This way, it will be easy for them to communicate with each other, furthermore, it will be easier for people to get the most benefits from these charities.

This document is divided into six chapters, chapter one talks about problem statement, project goals and objectives, project scope, project methodology and tasks. Chapter two talks about the planning of this project that explains in system actor, system requirements, constraints and risk analysis and solutions. Chapter three talks about analysis and design that explains in requirements analysis, use case diagram and system database. Chapter four talks about software implementation that explains in implementation tools, website interfaces and mobile application interfaces. Chapter five talks about system testing that explains in unit testing and integration testing. Chapter six concludes the document in results and recommendations for the system.

### 1.2 Problem Statement.

#### 1.2.1 Problems related to volunteers.

Because every charity has their own website or mobile application. Firstly, the volunteers face troubles to search for multiple websites and download multiple mobile applications. Secondly, difficult to find new charities. Thirdly, they face a big problem to keep knowing their news daily and what's the last updates. Finally, the differences in websites and applications designs create a problem to the volunteers, so our website displays simple interface and has all required information in the same place.

#### 1.2.2 Problems Related to Charities.

Charities face problems with websites and application managements and this is costly, also the difficulty of communication between charities itself.

## **1.3 Goals and Objectives.**

The main goal of this project is connect all charities and volunteers in the same place. The project targets the following specific objectives:

- Facilitate the communication between the charities and volunteers.
- Volunteers find new charities easily.
- Support the communication between charities itself.
- Support the communication between volunteers from different places.
- Easy way to know new events.

## **1.4 Project Scope.**

This project targets all charities and volunteers all over the world, also people who care about charities and support them physical or moral.

## **1.5 Project Methodology.**

Project methodology mainly depends on SDLC Model which describes a process for planning, creating, testing, and deploying an information system. This project is built on set of open source software which primarily used in programming PHP websites based on MVC framework and supported with central database. In addition, to set of other services using smart-phones software and Techniques that operating on Android platform .

### **1.5.1 MVC Design Pattern.**

The model-view-controller (MVC) design pattern, which is widely adopted in Web programming. MVC aims to separate business logic from user interface considerations, so that developers can more easily change each part without affecting the other. In MVC, the model represents the information (the data) and the business rules; the view contains elements of the user interface such as text, form inputs; and the controller manages the communication between the model and the view, as shown in Figure below.[1]

### **1.5.2 Why MVC ?**

Basically, a web application or piece of software that follows the MVC structure separates the three main types of functionality into three types of files: models, views, and controllers. This allows for each portion to be designed, implemented, and tested independently from any other one. Keeping the code organized means being able to find what is needed quickly, test features, correct or alter them quicker, and add new functionality with ease. It also means more efficient code, and a better way to re-use code for faster applications. Probably one of the greatest benefits however is that many developers understand and use the MVC structure for creating web applications. If developers use any of the popular web development frameworks, then they understand and use the MVC structure as well. Because of this consistency, managing a project between several developers can be easier as well.[2]

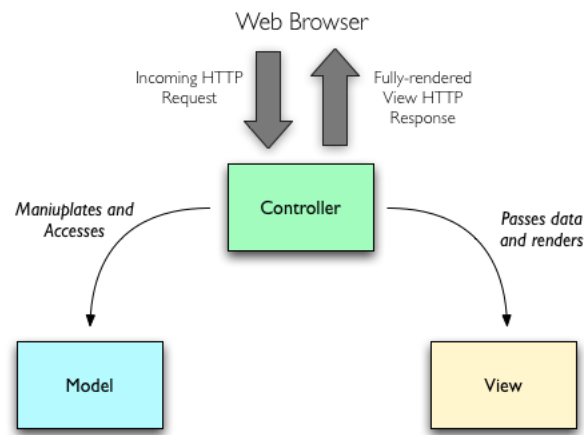


Figure 1.1: Model, view , controller transaction

### 1.5.3 The Directory Structure.

The file structure for using MVC in the standard way is quite simple — there are simply folders for views, models, and controllers, and they all link to each other via one directory up. Of course, with any web application the developer will also have other folders and files, such as an index file, a folder for images, library folder, and etc. The following description shows the directory structured of this project, ordered by communication levels from low (system core) to high (GUI):

- **Libs folder:** Application specific libraries. Basically, any kind of custom code that doesn't belong under controllers, models, or views. This folder is in the load path.
  - **Bootstrap.php:** class to controls loading of controllers and calling Actions.
  - **Controller.php:** class with single function to controls loading of models.
  - Database.php: PDO class for database connecting and access.
  - **Model.php:** class to represent, control, and track the transaction to and from the database by related queries (select, insert, update, delete).
  - **Session.php:** class to manage session functions and recall these functions in controllers, models or views.
  - **View.php:** class with a single function called render() to invoke page components from intended views sub-folder.
- **Models folder:**
  - Consists of multiple files. Each is related to a single controller file.
  - Each file receives instructions from its controller, ensure the secrecy and safety of the incoming data, filter any malicious requests then deliver it to the model file in libs folder.
  - When results are ready, the model file (in libs) will hand them back to the controller.
- **Controllers folder:**

- The controllers' files considered as mid-layer that takes in data from the user via view form, pass them to the model, and then get the appropriate data or calculation back to view layer to show the response to the user.

- **Views folder.**

- This folder stores the HTML files related to the main application screens, and graphical user interface.
- It has sub-folders each store a single page components in separated files
- Each view is connected with one controller.

Other folders and files to be considered:

- **Public folder:** This folder contains all asset files such as css, js, images, icons, etc.
- **uploads folder:** a folder to store users' profile images, pdf files, ... etc.
- **.htaccess file:** .htaccess is a configuration file for Apache Web Server software to run the site in offline mode on local machine.
- **Config.php file:** general database configuration.
- **Index.php file:** this file has a `_autoload()` function to trigger nested class loading at libs folder.

### 1.5.4 MVC Sequence Diagram.

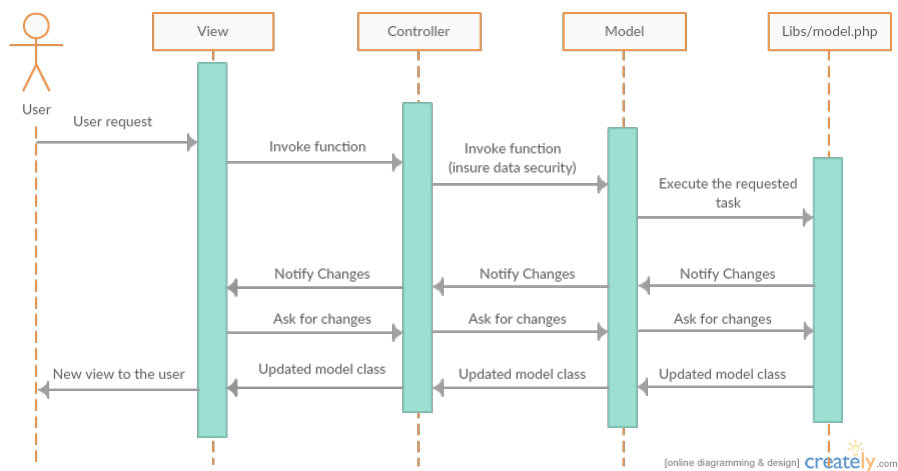


Figure 1.2: MVC sequence diagram

## 1.6 Tasks.

The work team consists of three members working together in all work stages. And this work team contribute to achieve the best ways to do the work. And this project consists of four stages:

- **First stage:** representing in meeting with project supervisor and team to collect required data about system.
- **Second stage:** represented in built the website using PHP programming language ,mobile application using Android and set of descriptive programming languages like HTML , JAVASCRIPT and CSS.
- **Third stage:** represented in running the website and mobile application in order to complete required developments and requirements before launch the website and the application.
- **Fourth stage:** documentation of all steps from the start to the end. Also documentation of maintenance operations for the problems that website and application face in running stage.

## 1.7 Times tables.

### 1.7.1 First Semester.

Task	1 <sup>st</sup> Month				2 <sup>nd</sup> Month				3 <sup>rd</sup> Month				4 <sup>th</sup> Month			
	WK1	WK2	WK3	WK4	WK1	WK2	WK3	WK4	WK1	WK2	WK3	WK4	WK1	WK2	WK3	WK4
Planning and collect information	█	█	█	█												
Collect system requirements					█	█										
Requirements description							█	█	█	█						
Suggestion a Setup Project Design											█	█	█			
Writing a First Draft Report													█	█		
Documentation	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
Presentation of First Semester																█

Figure 1.3: First semester time table.

### 1.7.2 Second Semester.

Task	1 <sup>st</sup> Month				2 <sup>nd</sup> Month				3 <sup>rd</sup> Month				4 <sup>th</sup> Month			
	WK1	WK2	WK3	WK4	WK1	WK2	WK3	WK4	WK1	WK2	WK3	WK4	WK1	WK2	WK3	WK4
correction of introduction	█	█	█													
Design of user interfaces				█	█	█										
Programming for website and mobile application ,development							█	█	█	█	█	█				
Verification of results													█	█		
Documentation	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█	█
Presentation of second Semester																█

Figure 1.4: Second semester time table.

# Chapter 2

## Planning

### 2.1 Introduction.

This chapter will show system actors, functional requirements and non-functional requirements. At the end of chapter will determine constrains and risks that project faces and the possible solutions to this risks.

### 2.2 System Actors.

- **Administrator:** Primary Actor that using the system to achieve a goal.
- **Charity:** Primary Actor that using the system to achieve a goal.
- **Volunteer:** Primary Actor that using the system to achieve a goal.
- **Database:** Secondary Actor that the system needs assistance from to achieve the primary actor's goal.
- **Bank Account:** Secondary Actor that the system needs assistance from to achieve the primary actor's goal.

### 2.3 System Requirements.

This system include functional and non-functional requirements that shown in this points:

#### 2.3.1 Functional Requirements.

- Administrator Requirements:
  1. Search for users.
  2. Approve charities.
  3. Block volunteers or charities.
- Charities Requirements:
  1. Create new account.



2. Add branch.
3. Update profile.
4. Search for volunteers and charities accounts.
5. Follow volunteers or charities.
6. Get notification about charity activity from following list.
7. Add new posts.
8. Delete posts.
9. Create events.
10. Join events.
11. search for events.
12. Add comment on charities posts.
13. Delete comments.
14. Report volunteers or charities.

- **Volunteers Requirements:**

1. Create new account on website.
2. Update profile
3. Search for volunteers and charities accounts.
4. Follow volunteers or charities.
5. Get notification about charity activity from following list.
6. Review charities.
7. Join events
8. search for events.
9. Add comment on charities posts.
10. Delete comments.
11. Donate to charities through website or application.
12. Report volunteers or charities.

### **2.3.2 Non-functional Requirements.**

- Easy to deal with the system.  
The software becomes more popular if its user interface is attractive, simple to use, clear to understand and consistent on all interfacing screens.
- Data security and integrity.  
The system must provide information protection from risks which threatened and from attack activities as users accounts hacking and access their information.
- High system efficiency and speed of response.  
The system must be responsive in short time, Response for all users' requests without impact on system speed.
- The ability to develop and maintain the system.  
The system must be designed in high degree of flexibility and that facilitate the development and periodic maintenance.
- High code quality.  
In website programming, the developers used MVC framework, so this make the code well-structured, clear and easy to edit and test.
- Backup and recovery.  
The system has a backup files, so that it is accessible in case of data deletion or corruption. And recovery in case of a loss and setting up systems that allow that data recovery due to data loss.

### **2.4 Constrains.**

1. Work with the available budget to build the system.
2. Completion of the project within a specified period of time, that is from the beginning of the semester to the end of the next semester.
3. Availability of central server, to connect system interface with each other.

### **2.5 Risks Analysis and Solutions.**

- Risks
  1. Delay of system delivery.
  2. The possibility of find new requirements during built or operate the system.
  3. Data loss.
  4. Unstable situations as the occupation may interrupt this semester, so we won't be able to deliver the project on time.
- Solutions
  1. Distribute roles between team members to deliver the system in the specified time.
  2. Take a backup of central server using another server, to save data from lost.
  3. Be accurate in determine system requirements.

# Chapter 3

## Analysis and Design

### 3.1 Introduction.

In this chapter we describe system requirements analysis. After that use case diagram to show interactions between a user (or more generally, an “actor”) and a system that enables the user to achieve a goal. At the end we display website and mobile application interfaces.

### 3.2 Requirements Analysis Tables.

#### 3.2.1 Administrator Requirements.

Table 3.1: Search for user.

Requirement name	Search for user.
Description	Administrator can search for users in users lists.
Input	user ID or username
Output	search result
Scenario	<ol style="list-style-type: none"><li>1. enter user ID or username in search bar.</li><li>2. Click on “search” button.</li><li>3. results will be displayed.</li></ol>

Table 3.2: Block volunteers or charities.

Requirement name	Block volunteers or charities.
Description	Administrator can block charity or volunteer.
Input	user ID
Output	Show message with success or fail operation.
Scenario	<ol style="list-style-type: none"><li>1. Choose charity or volunteer account from users list.</li><li>2. Click on “block” button.</li><li>3. New account added in the block list.</li></ol>

Table 3.3: Approve charities.

Requirement name	Approve charities.
Description	Administrator can approve charities accounts after make sure of the charity existence and validity of information.
Input	Charity ID.
Output	Show message with success or fail operation.
Scenario	<ol style="list-style-type: none"> <li>1. Open charities lists.</li> <li>2. Information click on “approve” button.</li> <li>3. System will Show message with success or fail operation.</li> </ol>

### 3.2.2 Charities Requirements.

Table 3.4: Create account for charities.

Requirement name	Create new account for charity in website/mobile application
Description	When the users access the website or install the application for the first time, user can create new account.
Input	Charity name, email, password, phone, POB, city, description, mission statements and location.
Output	System will Show message with success or fail operation.
Scenario	<ol style="list-style-type: none"> <li>1. Choose create new account button.</li> <li>2. Enter all information that maintained in the input field.</li> <li>3. Confirm the information and send request to admin to approve the account.</li> </ol>

Table 3.6: Add new posts.

Requirement name	Add new posts.
Description	User add new post to describe or talk about specific activity that done by charity.
Input	Post title, post text and photos.
Output	Show message with success or fail operation.
Scenario	<ol style="list-style-type: none"> <li>1. Choose new post.</li> <li>2. Enter text and photos if need.</li> <li>3. Click on “create post”</li> </ol>

Table 3.5: Add branch

Requirement name	Add branch
Description	The charity can add branches accounts.
Input	Charity ID, branch name, email, password, phone, POB, city, description, mission statements and location.
Output	Show message with success or fail process.
Scenario	<ol style="list-style-type: none"> <li>1. Display charity profile.</li> <li>2. Click on “Add branch” button.</li> <li>3. Enter all information that maintained in the input.</li> <li>4. Confirm the information and send request to approve the account.</li> </ol>

Table 3.7: Delete posts.

Requirement name	Delete posts.
Description	User can delete post if needed.
Input	Post ID.
Output	Show message with success or fail operation.
Scenario	<ol style="list-style-type: none"> <li>1. Choose post to delete.</li> <li>2. Click on delete button.</li> <li>3. Confirm the action.</li> </ol>

Table 3.8: Create events.

Requirement name	Create events.
Description	User can create event about specific activity.
Input	Event title, description, date, time, city and address
Output	Show message with success or fail operation.
Scenario	<ol style="list-style-type: none"> <li>1. Choose create event.</li> <li>2. Enter the information that maintained in input.</li> <li>3. Click on share event.</li> </ol>

Table 3.9: Cancel events.

Requirement name	Cancel events.
Description	User can cancel event if needed.
Input	Event ID.
Output	Show message with success or fail operation.
Scenario	<ol style="list-style-type: none"> <li>1. Choose event to cancel.</li> <li>2. Click on delete button.</li> <li>3. Confirm the action.</li> </ol>

### 3.2.3 Volunteers Requirements.

Table 3.10: Create new account for volunteer.

Requirement name	Create new account for volunteer.
Description	When the users access the website or install the application for the first time, the user can create new account.
Input	Enter username, Email, Phone number, Password, Country, City, POB, Interests and Website URL.
Output	Show message with success or fail process.
Scenario	<ol style="list-style-type: none"> <li>1. If "By Facebook" button clicked. <ul style="list-style-type: none"> <li>• Facebook message shown to tell user to access the profile or cancel.</li> <li>• If the user allow to access Facebook profile, account created and home page shown.</li> </ul> </li> <li>2. If "Get started" button clicked. <ul style="list-style-type: none"> <li>• Enter the information that maintained in input.</li> <li>• Show message with success or fail process.</li> </ul> </li> </ol>

Table 3.11: Review charities

Requirement name	Review charities.
Description	Review charities to give volunteers feedback about charities.
Input	Charity ID, Rate value and review.
Output	Show message with success or fail operation.
Scenario	<ol style="list-style-type: none"> <li>1. Display charity profile.</li> <li>2. Choose “Rate me”.</li> <li>3. Fill number of stars as needed from 1-5 and fill review field.</li> <li>4. Click on review button.</li> </ol>

Table 3.12: Donate to charities.

Requirement name	Donate to charities.
Description	Volunteers can donate to charity by credit card without need to visit the charity.
Input	Credit card information.
Output	Show message with success or fail process.
Scenario	<ol style="list-style-type: none"> <li>1. User enter the Credit card information.</li> <li>2. User fill the donation value field.</li> <li>3. Click on “submit donation”.</li> </ol>

### 3.2.4 Charities and Volunteers Requirements.

Table 3.13: Search for volunteers and charities accounts.

Requirement name	Search for volunteers and charities accounts.
Description	Search for charity or volunteer account by username.
Input	username
Output	-
Scenario	<ol style="list-style-type: none"> <li>1. User enter username in search box.</li> <li>2. User choose the account required.</li> <li>3. Display the account profile.</li> </ol>

Table 3.14: Get notification about charity activity.

Requirement name	Get notification about charity activity.
Description	Get notification about charity new events.
Input	Event ID
Output	Notifications.
Scenario	<ol style="list-style-type: none"> <li>1. Charity create new event.</li> <li>2. The system send notification for all users that follow this charity.</li> </ol>

Table 3.15: Follow volunteers or charities

Requirement name	Follow volunteers or charities.
Description	Follow volunteers and charities to show their posts and events in the home page.
Input	User ID
Output	-
Scenario	<ol style="list-style-type: none"> <li>1. User search for charity or volunteer account.</li> <li>2. Choose charity or volunteer account required.</li> <li>3. Click on “Follow” button.</li> </ol>

Table 3.16: Add comment on charities posts.

Requirement name	Add comment on charities posts.
Description	User can add comment on charity.
Input	Text.
Output	-
Scenario	<ol style="list-style-type: none"> <li>1. Enter text in the comment box.</li> <li>2. Click on “comment” button.</li> <li>3. New comment appear in post comments.</li> </ol>



Table 3.17: Delete comment.

Requirement name	Delete comment.
Description	User can delete comment from charity post if he need.
Input	Comment ID
Output	-
Scenario	<ol style="list-style-type: none"> <li>1. User choose his comment to delete.</li> <li>2. Click on delete button.</li> <li>3. Comment will be deleted.</li> </ol>

Table 3.18: Report volunteers or charities.

Requirement name	Report volunteers or charities.
Description	User can send report to the administrator about another user
Input	User ID, Report text and screen shot
Output	Show message with success or fail process
Scenario	<ol style="list-style-type: none"> <li>1. Fill report box.</li> <li>2. Click on "send" button.</li> <li>3.</li> </ol>

Table 3.19: Join events.

Requirement name	Join events.
Description	charities and volunteers can join events by website or mobile application.
Input	User ID and Event ID
Output	Show message with success or fail process.
Scenario	<ol style="list-style-type: none"> <li>1. choose the event to join from upcoming events or from charity profile.</li> <li>2. click on "join event" button.</li> <li>3. button changes to "leave event", if user clicked in the button another time, user will leave this event.</li> </ol>

Table 3.20: Search for events.

Requirement name	Search for events.
Description	volunteers and charities can search for event between two specific dates.
Input	first and second date.
Output	events between these dates or no result
Scenario	<ol style="list-style-type: none"> <li>1. user choose the dates to search for events.</li> <li>2. user fill "From" and "To" fields</li> <li>3. events between these dates will be displayed.</li> </ol>

### 3.3 Use Case Diagram.

A use case is a methodology used in system analysis to identify, clarify, and organize system requirements as shown in Figure 3.1.

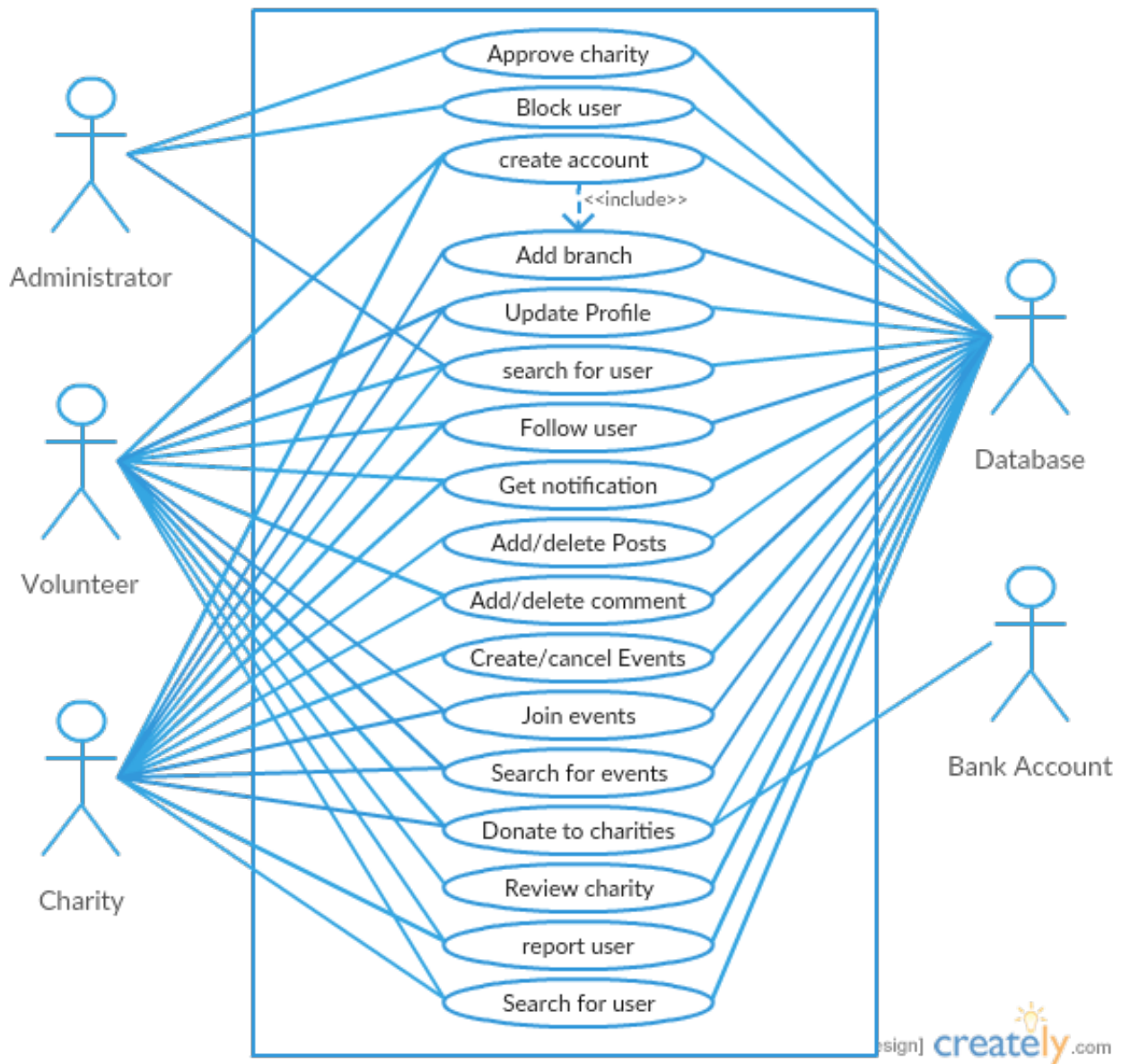


Figure 3.1: Use case diagram

### 3.4 System Database.

This section will displayed the tables of system database, describe the fields, relations between tables will be shown on database schema.<sup>1</sup>

- Administrator table.

Table 3.21: Administrator table.

Field Name	Field Type	Default	Null	Length	Description
<u>admin_id</u>	Integer	AI	No	2	Admin number
username	Varchar	-	No	32	Admin name
fname	Varchar	-	No	32	Admin first name
lname	Varchar	-	No	32	Admin last name
email	Varchar	-	No	64	Admin email
phone	Varchar	-	No	15	Admin phone number
password	Varchar	-	No	64	Admin password
address	Varchar	-	No	50	Admin address

- Users table.

Table 3.22: Users table.

Field Name	Field Type	Default	Null	Length	Description
<u>users_id</u>	bigint	AI	No	15	User number
Name	varchar	-	No	120	Username
password	varchar	-	No	32	User password
email	varchar	-	No	80	User email
profile_photo	varchar	null	Yes	25	User photo name
reg_time	timestamp	-	No	-	Registration time
gender	varchar	-	No	10	gender
birth_of_date	varchar	null	Yes	10	User birthdate
city_id	Integer	Null	No	5	City number
users_type <sup>1</sup>	tinyint	1	No	1	User type
describion	text	Null	Yes	-	Description
website_url	varchar	Null	Yes	100	Website URL
users_status <sup>2</sup>	tinyint	-	No	1	User status
POB	Integer	Null	Yes	11	User POB
intersting_id	Integer	-	No	3	Interest number
registration_method <sup>3</sup>	tinyint	1	No	1	Registration method

<sup>1</sup> 1 = 1= volunteer , 2= charity

<sup>2</sup> 1= active user , 2= blocked user

<sup>3</sup> 1 = account created by website, 2 = account created by facebook.

---

<sup>1</sup>static tables not included

- Charity table.

Table 3.23: Charity table.

Field Name	Field Type	Default	Null	Length	Description
<u>charity_id</u>	bigint	AI	No	15	Charity number
license_id	Integer	-	No	11	License number
father_id	bigint	Null	Yes	15	Father charity number
lat	vvarchar	-	No	10	latitude
long_b	vvarchar	-	No	10	longitude
mission_stat	text	-	No	-	Charity mission statement
is_approved <sup>1</sup>	tinyint	1	no	1	Approval status
bank_account	vvarchar	Null	Yes	13	Bank account

<sup>1</sup> 1 = not approved, 2 = approved.

- Events tables.

Table 3.24: Events tables.

Field Name	Field Type	Default	Null	Length	Description
<u>events_id</u>	Integer	AI	No	5	Event number
users_id	bigint	-	No	15	User number
events_title	vvarchar	-	No	100	Event title
events_description	text	-	No	-	Event description
events_date_from	date	-	No	-	Event start date
events_date_to	date	-	No	-	Event end date
events_time_from	vvarchar	-	No	10	event start time
events_time_to	vvarchar	-	No	10	Event end time
city_id	Integer	-	No	5	City number
events_address	vvarchar	-	No	25	Event address
add_time	timestamp	-	No	-	Time of added event

- Posts table.

Table 3.25: Posts table.

Field Name	Field Type	Default	Null	Length	Description
<u>post_id</u>	Integer	AI	No	5	Post number
post_text	Text	-	No	-	Post text
post_video	vvarchar	Null	yes	64	Post video name
Video_type <sup>1</sup>	vvarchar	Null	yes	15	Video type
post_time	timestamp	-	No	-	Post time added
users_id	bigint	-	No	15	User number
post_title	vvarchar	-	No	50	Post title

<sup>1</sup> video/mp4: mp4 uploaded video ,youtube: link from youtube.

- Activities table.

Table 3.26: Activities table.

Field Name	Field Type	Default	Null	Length	Description
<u>Activity_id</u>	Integer	AI	No	5	Activity number
users_id	bigint	-	No	15	User number
events_id	Integer	Null	Yes	5	Event number
post_id	Integer	Null	Yes	5	Post number
activity_type	varchar	-	No	10	Activity type

- Comments table.

Table 3.27: Comments table.

Field Name	Field Type	Default	Null	Length	Description
<u>comments_id</u>	Integer	AI	No	5	Comment number
post_id	Integer	-	No	5	Post number
users_id	bigint	-	No	15	User number
comment_text	Text	-	No	-	Comment text
comment_time	timestamp	-	No	-	Comment time added

- Notifications table.

Table 3.28: Notifications table.

Field Name	Field Type	Default	Null	Length	Description
<u>notification_id</u>	Integer	AI	No	5	Notification number
volunteers_id	bigint	-	No	15	User number
event_id	Integer	-	No	11	Event number
is_seen	tinyint	-	No	1	Seen status
time_add	timestamp	-	No	-	Time of notification added

0 = unseen , 1 = seen

- Ratings table.

Table 3.29: Ratings table.

Field Name	Field Type	Default	Null	Length	Description
<u>ratings_id</u>	Integer	AI	No	5	Rate number
review	varchar	Null	yes	150	Review text
users_id	bigint	-	No	15	User number
charity_id	bigint	-	No	15	Charity number
rating_value	double	-	No	-	Review value

- Joined events table.

Table 3.30: Joined events table.

Field Name	Field Type	Default	Null	Length	Description
<u>join_id</u>	Integer	AI	No	5	Event joined number
event_id	Integer	-	No	5	Event number
users_users_id	bigint	-	No	15	User number

- Followers table.

Table 3.31: Followers table.

Field Name	Field Type	Default	Null	Length	Description
<u>followers_id</u>	bigint	AI	No	15	Follow number
follower_id	bigint	-	No	15	Follower number
followee_id	bigint	-	No	15	Following number

- Donation table.

Table 3.32: Donation table.

Field Name	Field Type	Default	Null	Length	Description
<u>SN</u>	Integer	AI	No	5	Donation number
amount	Integer	-	No	11	Donation amount
datetime	timestamp	-	No	-	Donation date and time
charity_id	bigint	-	No	15	Charity number
users_id	bigint	-	No	15	User number

- Reports table.

Table 3.33: Reports table.

Field Name	Field Type	Default	Null	Length	Description
<u>reports_id</u>	Integer	AI	No	5	Report number
reports_text	Text	-	No	-	Report text
users_id	bigint	-	No	15	User number
user_report_id	bigint	-	No	15	User number who send the report
report_photo	varchar	null	yes	25	Report screen shot

- Phones table.

Table 3.34: Phones table.

Field Name	Field Type	Default	Null	Length	Description
<u>phones_id</u>	Integer	AI	No	5	Phones sequence number
phone	varchar	-	No	15	Phone number
users_users_id	bigint	-	No	15	User number

- Photos table.

Table 3.35: Photos table.

Field Name	Field Type	Default	Null	Length	Description
<u>photos_id</u>	Integer	AI	No	5	Photo number
post_id	Integer	-	No	5	Post number
photo_name_path	varchar	-	No	25	Photo name



### 3.5 Database schema.

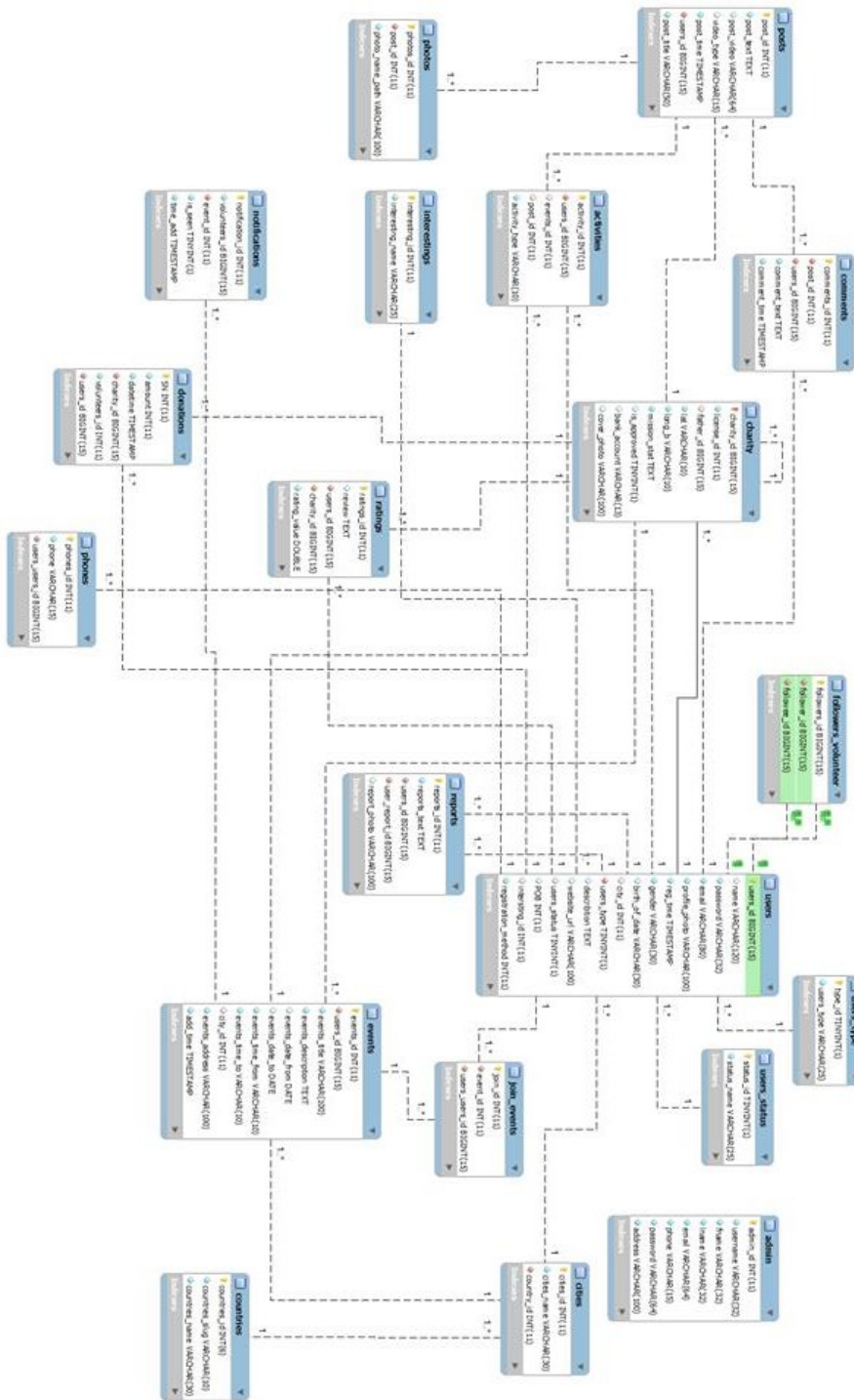


Figure 3.2: Database Schema

# Chapter 4

## Software Implementation.

### 4.1 Introduction.

This chapter will describe the developed software, display user interfaces for the website and the mobile application and implementation tools used to build the systems.

### 4.2 Implementation tools.

#### 4.2.1 WAMP Server.

Refers to a set of free (open source) applications. The WAMP stack provides developers with the four key elements of a Web server: an operating system, database, Web server and Web scripting software. The combined usage of these programs is called a server stack. In this stack, Microsoft Windows is the operating system (OS), Apache is the Web server, MySQL handles the database components, while PHP represents the dynamic scripting language.

#### 4.2.2 NetBeans.

NetBeans is a Java-based integrated development environment (IDE) for developing with Java, PHP, C++, and other programming languages.

#### 4.2.3 Android Studio.

Android studio is the official IDE for android platform development. Used to build application for smartphones based on java programming language.

#### 4.2.4 Bootstrap Framework.

Bootstrap is a free and open-source front-end library for creating websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional JavaScript extensions. It aims to ease the development of dynamic websites and web applications.[3]

## 4.3 Website Interfaces.

### 4.3.1 Administrator Control Panel.

Administrator control panel has multiple pages to control and manage the system. For example, charities lists, volunteers lists, blocked users list ...etc. as shown in Figure 4.1.

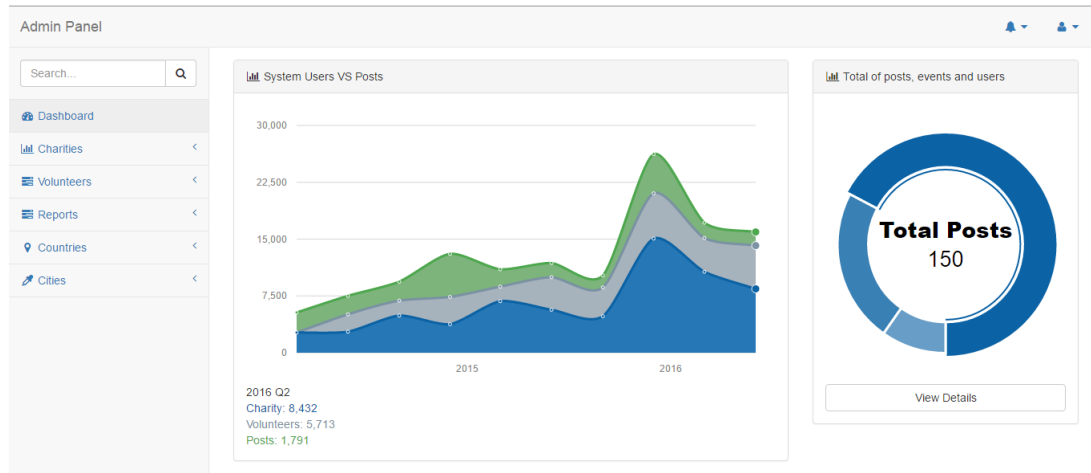


Figure 4.1: Administrator control panel

### 4.3.2 Index Page.

Index page is generally the main page a visitor navigating to a website from a search engine will see, and may also serve as a landing page to attract the attention of visitors.

- It contains the sign in form to navigation to other pages on the website. Also it contains description about the website and its services that will be shown when the user click on "about" or "service" in the header. As shown in Figure 4.2.

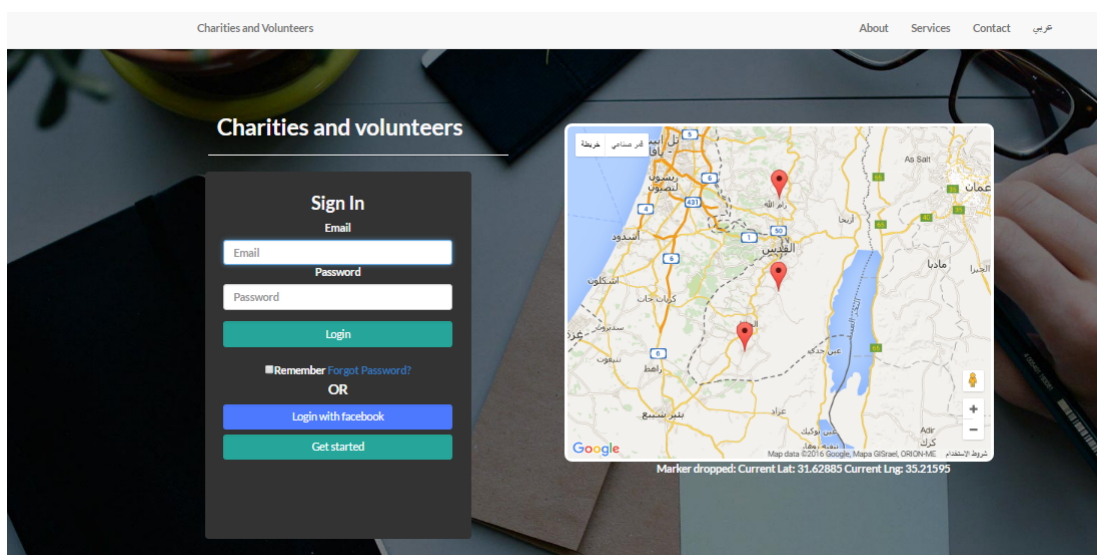


Figure 4.2: Index page

- This website support Arabic language in addition to English language. as shown in Figure 4.3.

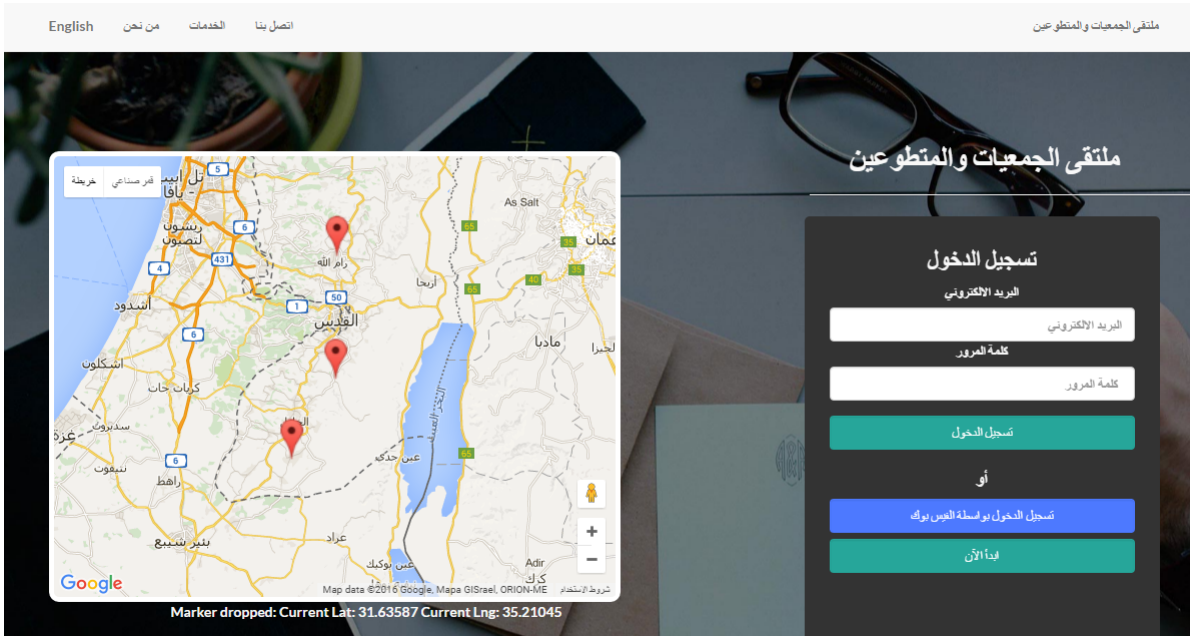


Figure 4.3: Arabic index page

### 4.3.3 User Homepage.

By default, There is a different home page for each user which display posts and upcoming events from charities in following list as shown in Figure 4.4. Also user can sorted the posts by interests or location.

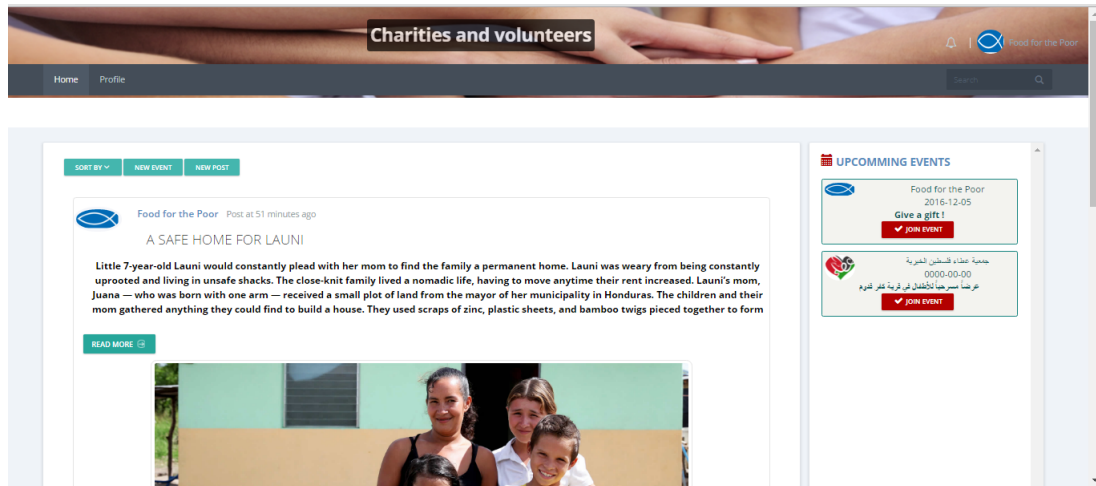


Figure 4.4: User Homepage.

### 4.3.4 Charity Profile.

Each charity has a profile to display and edit their information, posts and events and activities. As shown in Figure 4.5.



Figure 4.5: Charity profile.

### 4.3.5 Volunteer Profile.

Each charity has a profile to display and edit their information and view activities. As shown in Figure 4.6.

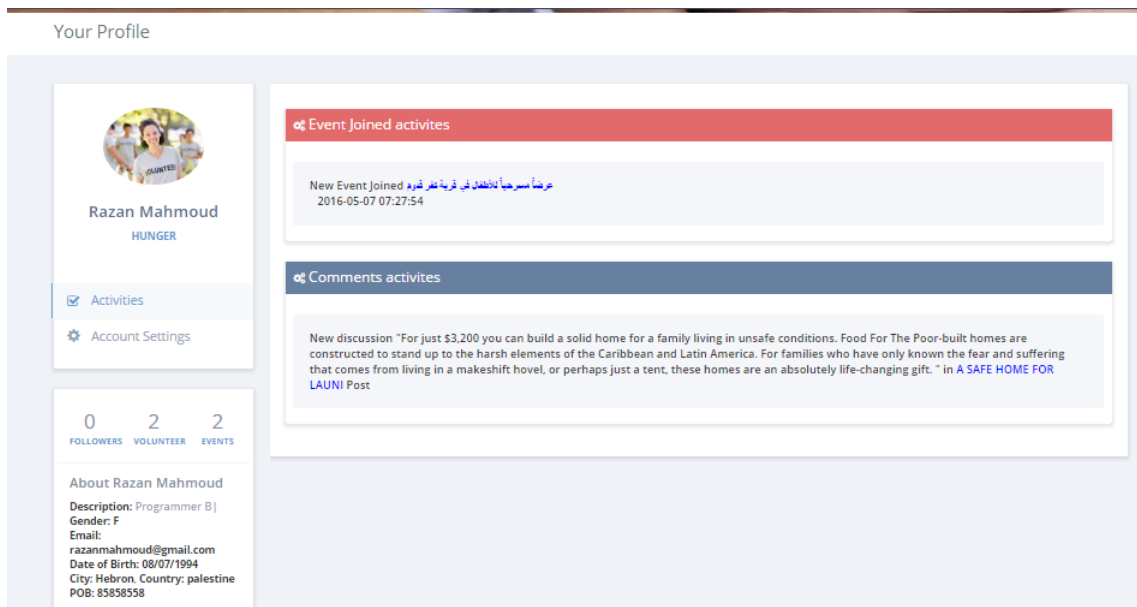


Figure 4.6: Volunteer profile.

### 4.3.6 Mobile Friendly Interfaces.

A mobile friendly website is essentially when your regular website shrinks down to be small enough to display on a mobile device, so the developers built the website with a mobile friendly design as shown in the following Figures:

- Index Page

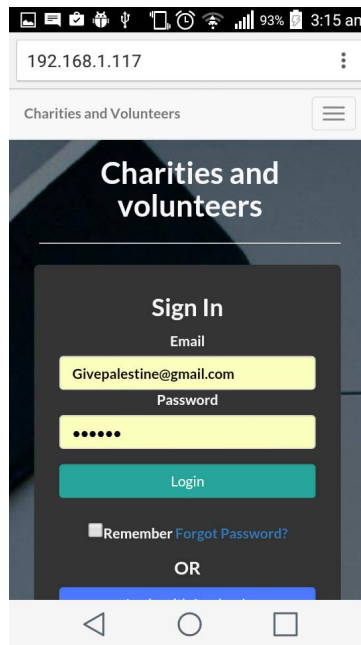


Figure 4.7: Index page

- Home Page



Figure 4.8: Home page

- Profile Account

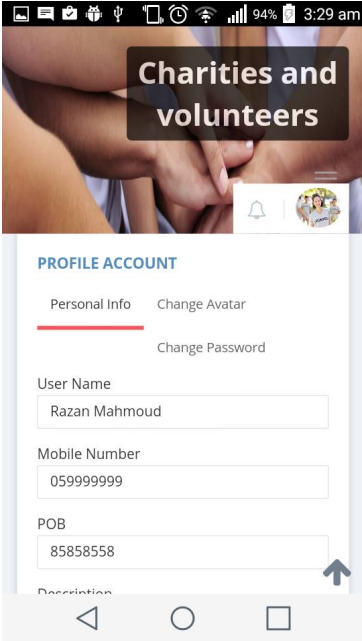


Figure 4.9: Profile page

## 4.4 Mobile Application Interfaces.

### 4.4.1 Login Screen.

It contains the signin form to navigation to other screens on the application. As shown in Figure 4.10.

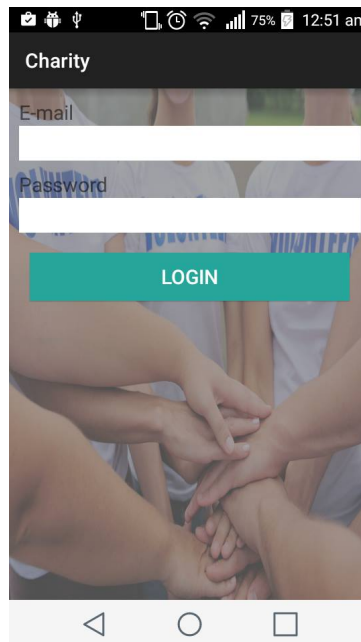


Figure 4.10: Signin screen.

### 4.4.2 Home Screen.

There is a different home page for each user which display posts and upcoming events from charities in following list as shown in Figure 4.11.



Figure 4.11: Home screen.



### 4.4.3 New Event Screen.

This screen enable the charity to create events after fill the required fields. As shown in Figure 4.12.

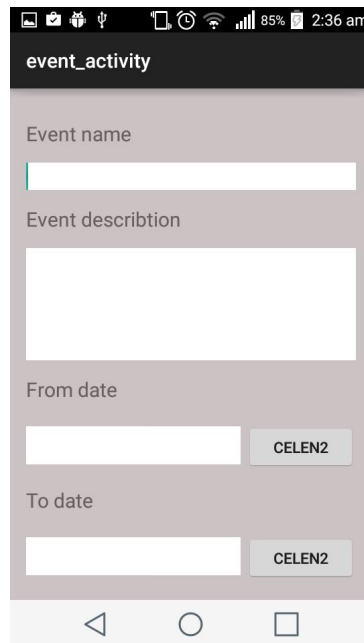


Figure 4.12: New event screen.

### 4.4.4 New Post Screen.

This screen enable the charity to add new posts after fill the required fields. As shown in Figure 4.13.

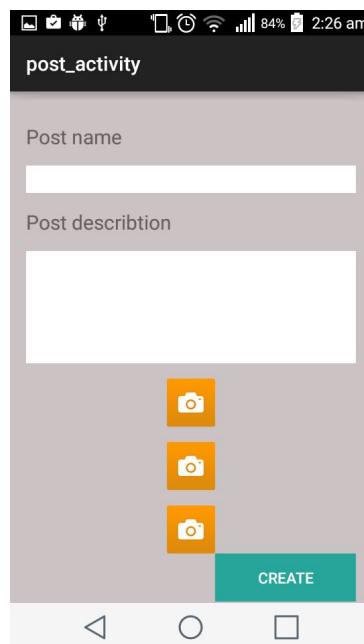


Figure 4.13: New Post screen.

# Chapter 5

## Testing

### 5.1 Introduction.

This chapter documented the system testing and insure that system work properly, manually unit testing and integration testing.

### 5.2 Website Unit Testing.

In unit testing, all parts of the system tested separately. And this table show a report of tested components and the result of the test.

Table 5.1: Website components report

#	Functionality Description	Status(pass/fail)
Index page		
1	Log in form	Pass
Header Contents		
2	Search bar in the website header	pass
3	Notification in the website header	Pass
4	User account option	pass
5	Profile button	pass
Volunteer home page		
6	Display page contents	pass
7	Posts wall	Pass
8	Events wall	pass
9	Sort posts by interests	pass
10	Sort posts by location	pass
11	Sort posts by interests and location	pass
12	Read more button in posts wall	pass
13	Join/leave event button	pass
14	See more posts and events in scroll down	pass
Charity home page		
15	Display page contents	pass
16	New post button	pass
17	New event button	pass
18	Posts wall	Pass
19	Delete post icon	pass

20	Events wall	pass
21	Delete event	pass
22	Sort posts by interests	pass
23	Sort posts by location	pass
24	Sort posts by interests and location	pass
25	Read more button in posts wall	pass
26	Join/leave event button	pass
27	See more posts and events in scroll down	pass
Volunteer profile page		
28	Display page contents	pass
29	Activities wall	pass
30	Left side bar profile	pass
31	Right side bar profile	Pass
Volunteer account settings page		
32	Display page contents	pass
33	Personal info bar	pass
34	Change avatar bar	pass
35	Change password bar	Pass
Charity Profile		
36	Display page contents	pass
37	Display charity posts	pass
38	Display charity events	pass
39	Change password bar	Pass
40	New post button	pass
41	Delete post	pass
42	New event button	pass
43	Delete event	pass
44	Add branch button	pass
45	Donate button	pass
46	Join/leave us button	pass
47	Report a problem button	pass
Charity activity page		
48	Posts activities table	pass
49	Events activities table	pass
50	Event joined activities table	pass
51	Comments activities table	Pass
Charity account settings page		
52	Display page contents	pass
53	Personal info bar	pass
54	Change avatar bar	pass
55	Change password bar	Pass
Charity about page		
56	Display page contents	pass
Charity rating page		
57	Display page contents	pass
58	Submit review	pass
search page		
59	Display page contents	pass

Notification bar		
60	Display page contents	pass
Event details page		
61	Display page contents	pass
62	Join event button	pass
Post details page		
63	Display page contents	pass
64	Add/delete comment	pass
New post page		
65	Display page contents	pass
66	Add new post	pass
New event page		
67	Display page contents	pass
68	Add new event	pass
Add branch page		
69	Display page contents	pass
70	Add new branch	pass
Sign up page		
71	Display page contents	pass
72	Add new user	pass
Report page		
73	Display page contents	pass
74	Add new report	pass
Logout page		
75	Logout from the website	pass
Donation page		
76	Display page contents	pass
77	Make a donation	fail

### 5.3 Mobile Application Unit Testing.

this table show a report of tested components and the result of the test.

Table 5.2: Mobile application components report

#	Functionality Description	Status(pass/fail)
Login screen		
1	Log in form	Pass
New post screen		
65	Display page contents	pass
66	Add new post	pass
New event screen		
67	Display page contents	pass
68	Add new event	pass
Volunteer home screen		
6	Display page contents	pass
7	Posts wall	Pass

8	Events wall	pass
14	See more posts and events in scroll down	pass
Charity home screen		
15	Display page contents	pass
16	New post button	pass
17	New event button	pass
18	Posts wall	Pass
20	Events wall	pass
27	See more posts and events in scroll down	pass

## 5.4 Interfaces for Units Testing.

- Login interfaces: if the user enter the email and password correct, the system redirect the user to the home page. But if the user enter incorrect username or password, error message appears as shown in Figure 5.1 below. Also if the user blocked, another error message appears as shown in Figure 5.2 below.

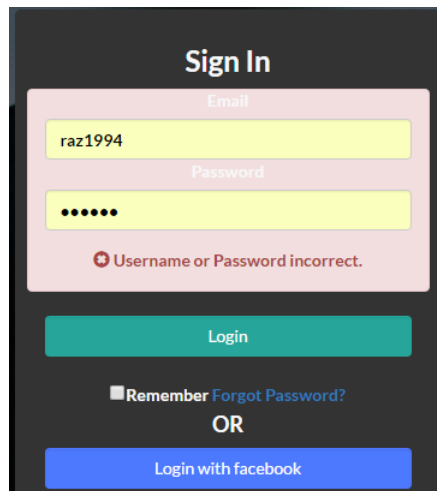


Figure 5.1: Username or password in correct

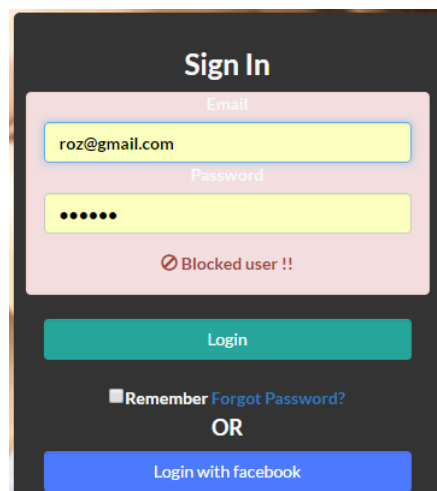


Figure 5.2: Blocked user

- Signup interfaces: if the user fill valid information in the fields, success message appears as shown in Figure (5.3) below.

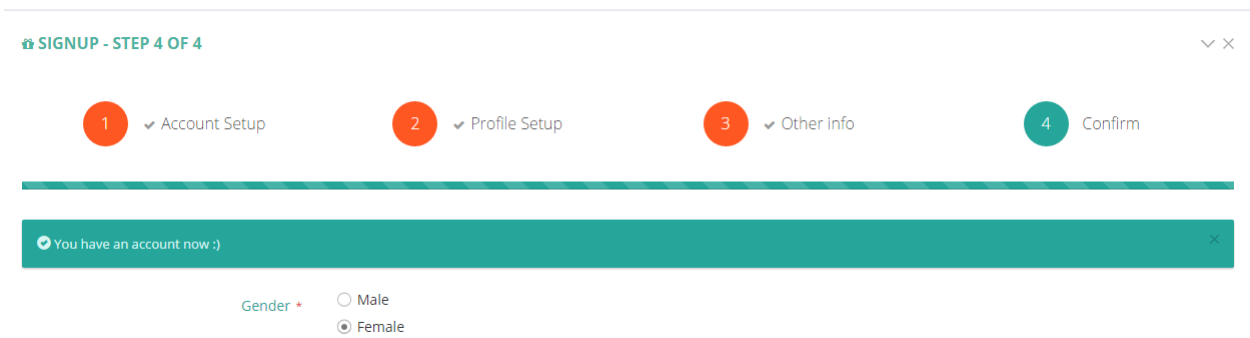


Figure 5.3: Successful message after creating account

When the user enter valid information in the fields, tcolour of the fields and labels change to green as show in Figure(5.4) below.

Provide your account details

Username \*   
Provide your username

Password \*   
Provide your password.

Confirm Password \*   
Confirm your password

Email \*   
Provide your email address

Figure 5.4: Valid information

If the user doesn't fill the fields, error messages appear as shown in Figure (5.5) below.

Provide your account details

The form contains four input fields, each with a red border and a red asterisk label. Below each field is a red error message. At the bottom is a blue 'CONTINUE' button with a right-pointing arrow.

Username \*  
This field is required.  
Provide your username

Password \*  
This field is required.  
Provide your password.

Confirm Password \*  
This field is required.  
Confirm your password

Email \*  
This field is required.  
Provide your email address

CONTINUE →

Figure 5.5: Required fields error

If the user enter a password with less than 5 characters, error messages appear as shown in Figure (5.6) below.

The form shows a single input field with a red border and a red asterisk label. The field contains three dots. Below the field is a red error message. Above the field is a teal label 'Provide your username'.

Provide your username

Password \*  
Please enter at least 5 characters.  
Provide your password.

Figure 5.6: Password characters error

If the user doesn't re-enter the same password in "Confirm password field", error messages appear as shown in Figure (5.7) below.

The form shows two input fields. The first has a teal border and a teal asterisk label. The second has a red border and a red asterisk label. Below the second field is a red error message.

Password \*  
Provide your password.

Confirm Password \*  
Please enter the same value again.  
Confirm your password

Figure 5.7: Value in confirm password field error

- create events and notification units test: in this stage of unit testing, the developer test the create event and this depends on login and sign up units. Also notification test depends on login, sign up, search for charity and join charity units. And this concluded in three steps as following:
  - **Step 1: Sign in as volunteer and search for charity to join.** The name of charity of part of name filled in the search bar as shown in Figure (5.8) below.

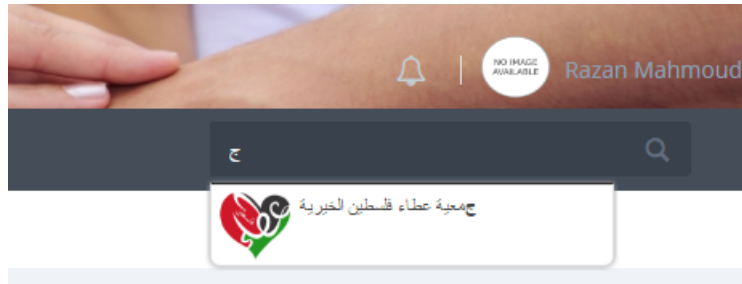


Figure 5.8: Search for charity.

The result of the search operation will appear in search result page as shown in Figure (5.9) below.

#### Search Results



Figure 5.9: Search results.



In charity that searched profile, “leave us” button in Figure (5.10) below means that the user join this charity.

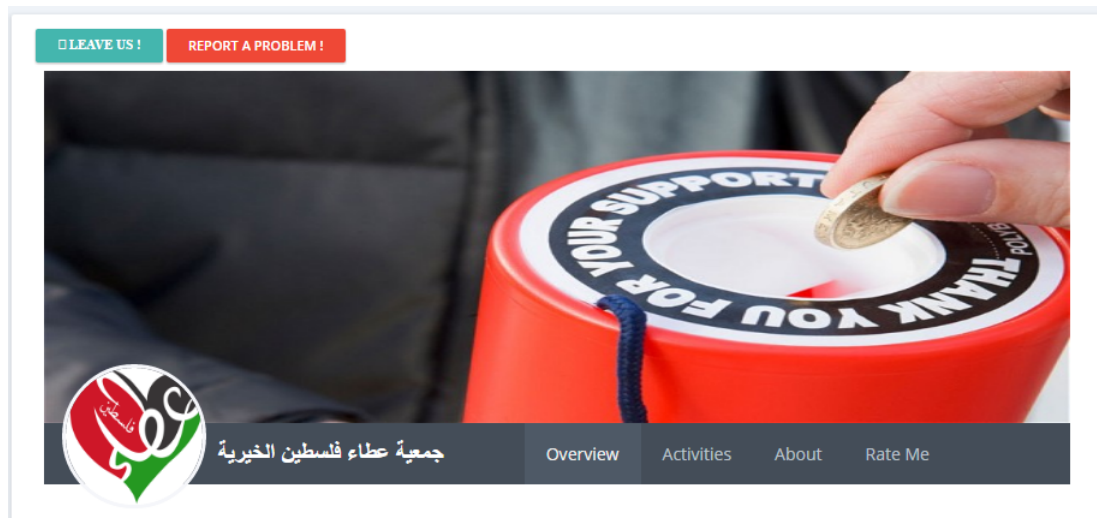


Figure 5.10: Charity profile.

– **Step 2: Sign in as charity that volunteer joined to create new event.**

In new event page, charity fill all fields related to the event to create as shown in Figure (5.11) below.



**NEW EVENT**

Event Title  
عرضاً مسرحياً للأطفال في قرية كفر قنوم

Description  
كفر قنوم-17-5-2016-تفتت جمعية عطاء فلسطين الخيرية، عروضاً ثقافية للأطفال، في مدرسة كفر قنوم الأساسية للذكور من الصف الأول ولعالية الصف الرابع الأساسي، في قرية كفر قنوم - قفيلية بعنوان  
"التشجيع على القراءة وتعليم العادات السلوكية المحسنة والدعم النفسي للأطفال"

From: 05/17/2016 Date to: 05/17/2016

Time From: 9:00 AM Time To: 4:00 PM

Country: palestine City/Town \*: Jerusalem

Address: كفر قنوم

**CREATE**

Figure 5.11: New Event.

Success message will appear after event created as shown in Figure(5.12) below.



Figure 5.12: Success message.

- **Step3: sign in as volunteer to make sure that volunteer get notification about the event created by charity.**

User get notification about the new event in notification bar as shown if Figure (5.13) below.

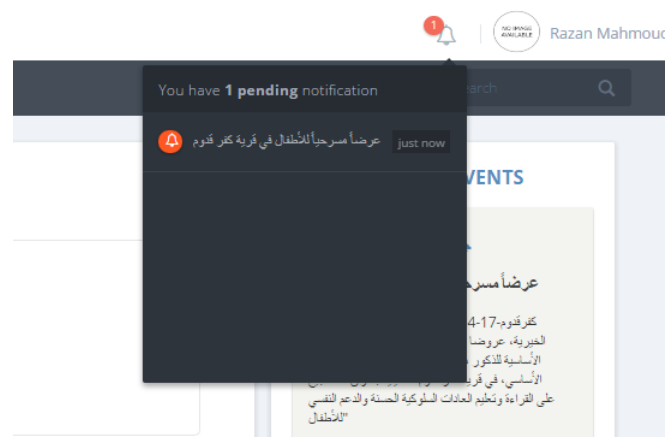


Figure 5.13: Get notification about charity event.

## 5.5 Integration Testing.

In integration testing, all parts of system tested integrally. To make sure that the interaction between all system parts work properly. The developers test the system by divide the system in three modules as following:

1. Interfaces modules: which is website or mobile application interfaces visible to the end user, where all the inputs are given.
2. Business Logic module: which has all the all the calculations and business specific methods.
3. Database.

Integration testing focus on the flow of data between the modules.

# Chapter 6

## Results and Recommendation

### 6.1 Introduction.

The final chapter of the project documentation conclude the work results and give future recommendations.

### 6.2 Project Results.

The team work succeeded in build system for charities and volunteers to connect all of them in one place to facilitate the communication between them, Based on that they have reached to the following results:

1. The system provide communication environment between charities and volunteers in a simple and clear ways.
2. The system provide a simple way for charities to share their activities and events, so everyone registered in the system can view them and interact.
3. The system provide a simple way for volunteers to volunteer with charities, interact with them by make discussions or join their events, make a donation to help them.
4. The system provide a mobile application for smart-phones to easy access and get notification about events.

### 6.3 Future Recommendation.

1. Provide a fully control panel for the administrator to edit the system easily and get fully reports, percentages and graphs.
2. Provide messages feature that give the ability for two users to communicate with each other privately.
3. Provide more than one language in the system.
4. Add new feature for mobile application to provide all features in the website.
5. Use SSL certificates that give the highest security and scalability for the system.
6. Provide email verification.

7. Hosting the website on amazon web service (AWS) to have trusted, reliable and scalable host.
8. Promote the website to a different audience in LinkedIn to make a business.

# References

- [1] Pei-jiong FU and Zhong-jun DU. Application of spring in realizing mvc framework [j]. *Computer Technology and Development*, 6:081, 2006.
- [2] Kristof Geebelen, Eryk Kulikowski, Eddy Truyen, and Wouter Joosen. A mvc framework for policy-based adaptation of workflow processes: a case study on confidentiality. In *Web Services (ICWS), 2010 IEEE International Conference on*, pages 401–408. IEEE, 2010.
- [3] Bradley Efron. *Bootstrap methods: another look at the jackknife*. Springer, 1992.