
Software Test Plan

for

Online Volunteer Management System

Version 1.0 approved.

Prepared by X

American International University-Bangladesh (AIUB)

2023.12.24

Table of Contents

Revision History	2
1. TEST PLAN IDENTIFIER: VMS-TP01.0	3
2. REFERENCES.....	3
3. INTRODUCTION.....	3
Background to the Problem.....	3
Solution to the Problem.....	3
4. REQUIREMENT SPECIFICATION	4
4.1 System Features	4
4.2 System Quality Attributes.....	8
4.3 UI/UX	9
4.4 System Interface.....	13
4.5 Project Requirements	15
5. FEATURES NOT TO BE TESTED.....	17
6. TESTING APPROACH	17
6.1 Testing Levels.....	18
6.2 Test Tools.....	19
6.3 Meetings.....	20
7. TEST CASES/TEST ITEMS	20
8. ITEM PASS/FAIL CRITERIA	26
9. TEST DELIVERABLES	27
10. STAFFING AND TRAINING NEEDS.....	27
11. RESPONSIBILITIES.....	28
12. TESTING SCHEDULE.....	28
13. PLANNING RISKS AND CONTINGENCIES	29
14. APROVALS.....	29

Revision History

Revision	Date	Updated by	Update Comments
0.1	2023.12.02	Khandoker Mostakim Kabir	First Draft
0.2	2023.12.10	Nafisa Maliyat	Updated Test Cases, Schedule
0.3	2023.12.15	An Nazmus Sakib	Added Meeting plans, Testing tools
0.4	2023.12.22	Limia Sadina Sathi	Updated system quality attributes

1. TEST PLAN IDENTIFIER: **VMS-TP01.0**

2. REFERENCES

- UI/UX Designing tools <https://www.figma.com/>
- Case Drawing tools <https://www.umlet.com/>

3. INTRODUCTION

Background to the Problem

- Bangladesh is a developing country. Every day we face copious problems in our communities. It is not possible for the government to solve all the problems. Such as waterlogging, traffic jams, animal danger, an unhygienic environment, lack of education, etc. These problems should be considered important to make a community safe, educated, and clean because they can help improve the community and the quality of life.

Solution to the Problem

- Our solution is to provide a sophisticated system for students and local people to identify a problem and work together to solve the problem by assigning volunteers in the community. Using an app, after sufficient requests have been made, we will check whether any volunteers are free or nearby to be able to help. Depending on the problem, available volunteers with specific skillsets will reach the problem location and try to solve it. These solutions can be clearing water blockages, managing traffic, helping animals, cleaning parks etc. Our government is busy with bigger projects rather than these trivial community problems. By using our manpower, we can solve the problems efficiently as they are working willfully with their own chosen skillsets in their own areas. This is a social welfare system for people to help their community. The basic functionality of our proposed solution includes identifying the problem and tapping the request option for that particular group of problems categorized in the app. Which will then help assign volunteers using AI. After local people or students place request, the location and the problem are made visible in the app with the help of GPS technology along an indicator displaying the severity of the problem. The severity of the problem depends on the number of people identifying and placing requests in the app for the same problem in a short period of time. The volunteers nearby the problem location are notified with the help of AI and are requested to help. The volunteers with the needed skillset are notified first. If they are available, they will press the responding button in the app which will notify the people that the volunteers are on their way. After reaching the location, the volunteer updates his or her response to “Reached”. The volunteer will try to fix the problem and if they succeed

the status will be updated to completed. If he/she needs additional help, they can request that via the app and other volunteers can join. As the problems will be queued by their severity level, the volunteers will be responding first to the most severe problem at hand. After that they can move onto the next task in queue. By solving the problems through this app, we can make a good impact in our society. This app is specifically targeted for the people of an area or a community. By using the app, everyone can come together to fix their communal problems fast rather than waiting for government to interfere. Also, willing young people and students can volunteer themselves for the good of others by signing up as a volunteer. They can see how much time they have given in doing good for others and compare stats with other volunteers. This can grow into a friendly competition which in turn would benefit their community.

- There is currently no software solution for this problem in our country, there are many volunteering organizations but they don't have a proper software system to manage their volunteer.

4. REQUIREMENT SPECIFICATION

4.1 System Features

1. Registration

- 1.1 Open the app.
- 1.2 The volunteer will fill in the registration form with his/her full name and NID along with username and a 6-digit password.
- 1.3 Volunteers email address must be provided in the form.
- 1.4 After submitting the form, the app will send a confirmation mail to the volunteer's email.
- 1.5 After opening the confirmation link, a new profile is created for the volunteer. The volunteer can now login.

Priority Level: Low

Precondition: App must be installed, and location permission should be granted.

Cross-reference: 2.1, 2.2

2. Volunteer Profile

- 2.1 The software shall allow volunteers to access their profile after logging in.
- 2.2 Volunteers can change their username, user ID, password, skillset, preferred work by accessing the edit information tab.

- 2.3 If a change is made, the database information is updated.
- 2.4 There will be a badge section in which volunteers can show the badges they earned through reaching milestones in their voluntary works.
- 2.5 Volunteers can upload pictures of a problem they've solved which in turn can be viewed by other volunteers by accessing their profile.
- 2.6 There will be a Delete and a Deactivate tab for volunteers to remove their profile or to take a break from volunteering.

Priority Level: Low

Precondition: User must be registered and logged in.

Cross-reference: 1.1, 1.5, 2.1, 2.3

3. Problem Locating & Requesting

- 3.1 The software shall allow people other than volunteers to enter without logging in to locate problems and request help.
- 3.2 In the login section, people can enter in the 'Request help' tab to request help for a problem they've located.
- 3.3 The 'Request help' tab shall have a section where people can put the location, problem category and locating time for a problem.
- 3.4 The information about problems is stored in the database and artificial intelligence determines the severity of the problem based on the number of people requesting help for one particular problem.

Priority Level: High

Precondition: App must be installed, and location permission should be granted.

Cross-reference: 5.1, 6.1, 6.3, 7.1, 7.2, 8.1

4. Show Problems

- 4.1 The software shall allow volunteers to view problems from the database in their profile after logging in.
- 4.2 The volunteer can even select a location to see problems of that particular area.
- 4.3 If volunteer is available for the current problem, he can confirm.
- 4.4 A volunteer can suggest another volunteer.
- 4.5 A volunteer can share ideas based on the problem.

Priority Level: High

Precondition: The volunteer should be logged in.

Cross-reference: 4.1, 4.4

5. Notify Volunteers

5.1 The software shall use GPS to search the map for volunteers within a certain radius of the identified problem location.

5.2 The system will identify the volunteers present in the given radius.

5.3 The system will then check the problem requirements with the skills and experience of the volunteers from the volunteer database.

5.4 Artificial Intelligence will be used to find the volunteer most suitable to deal with the problem.

5.5 The software then notifies the best qualified volunteer with a notification request.

5.6 If the qualified volunteer accepts the request, he or she will be shown the way.

5.7 If the qualified volunteer rejects the request, the software then passes the notification request to the second-best qualified volunteer.

5.8 If no volunteer is found in the assigned radius from the problem, the system will increase the radius to broaden the search area for volunteers.

Priority Level: High

Precondition: Volunteers have to log in and their location must be turned on.

Cross-references: 7.1, 8.1

6. Show location & estimated time

6.1 The volunteer should be able to see the precise geographical position on a map using GPS.

6.2 The volunteer should be able to see nature or type of road and its status (closed/open/partially open etc.).

6.3 The volunteer should be able to see the duration of the work in progress.

6.4 Information on how long the problem may be solved should be available with the help of GPS and artificial intelligence.

Priority Level: High

Precondition: Volunteers have to log in and their location must be turned on.

Cross-reference: 4.1, 4.4, 6.6

7. Problem Status

- 7.1 The nature of the problem that is being solved should be displayed.
- 7.2 Problem status will change to “In Progress” when a volunteer reaches the problem location.
- 7.3 It will also show the duration that the problem-solving process has been in progress.
- 7.4 The system should show how many volunteers are working on the issue.
- 7.5 It will display the severity of the problem and if any additional help has been requested.
- 7.6 The experience level of those working on the problem will also be displayed.

Priority Level: Medium

Precondition: App must be installed and opened.

Cross-reference: 4.4, 4.4, 7.3, 3.4, 9.2

8. Request Volunteers

- 8.1 The deployed volunteer can place requests for additional volunteers.
- 8.2 The volunteer can even add additional requirements for the required additional volunteer.
- 8.3 The volunteer can choose the option to leave after the additional volunteer arrives. (optional).
- 8.4 The system checks the severity of the problem to check for additional volunteers.
- 8.5 Depending on the number of problems in the area, GPS tracks the free volunteers in the area.
- 8.6 The free volunteers are notified regarding the task along with the problem status.

Priority Level: Medium

Precondition: Volunteers have to log in and their location must be turned on.

Cross-references: 6.3, 6.4, 6.5, 6.6, 6.7, 6.8, 8.1, 8.2, 8.3

9. Notify Local Administrator

- 9.1 An automated generated e-mail about the problem including time schedule, volunteer profile and other necessary information will be sent to the respective ward Counsellor from the database with the help of artificial intelligence.
- 9.2 A confirmatory reply with administrative support will be provided (if necessary) regarding the problem.

Priority Level: Low

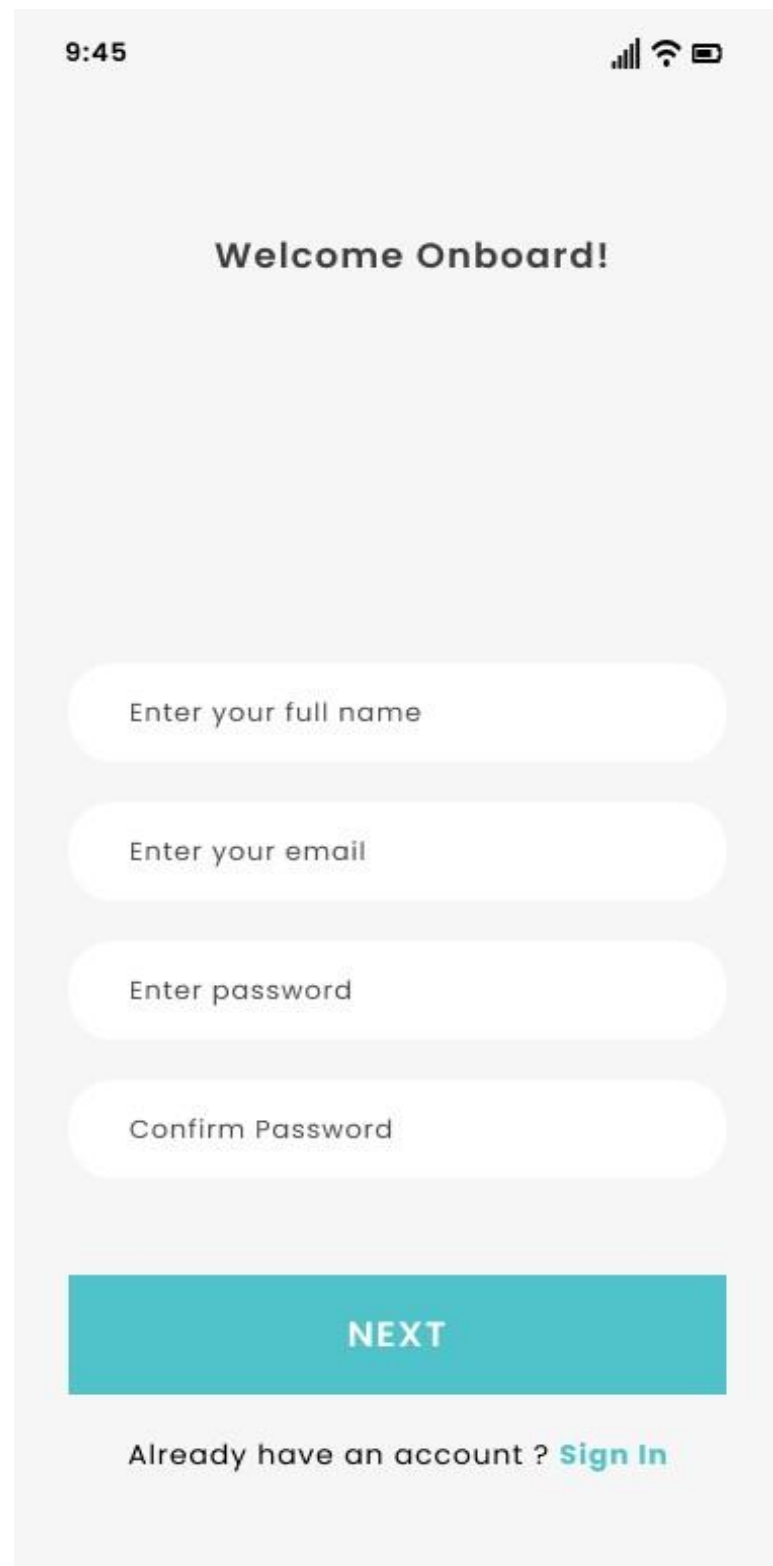
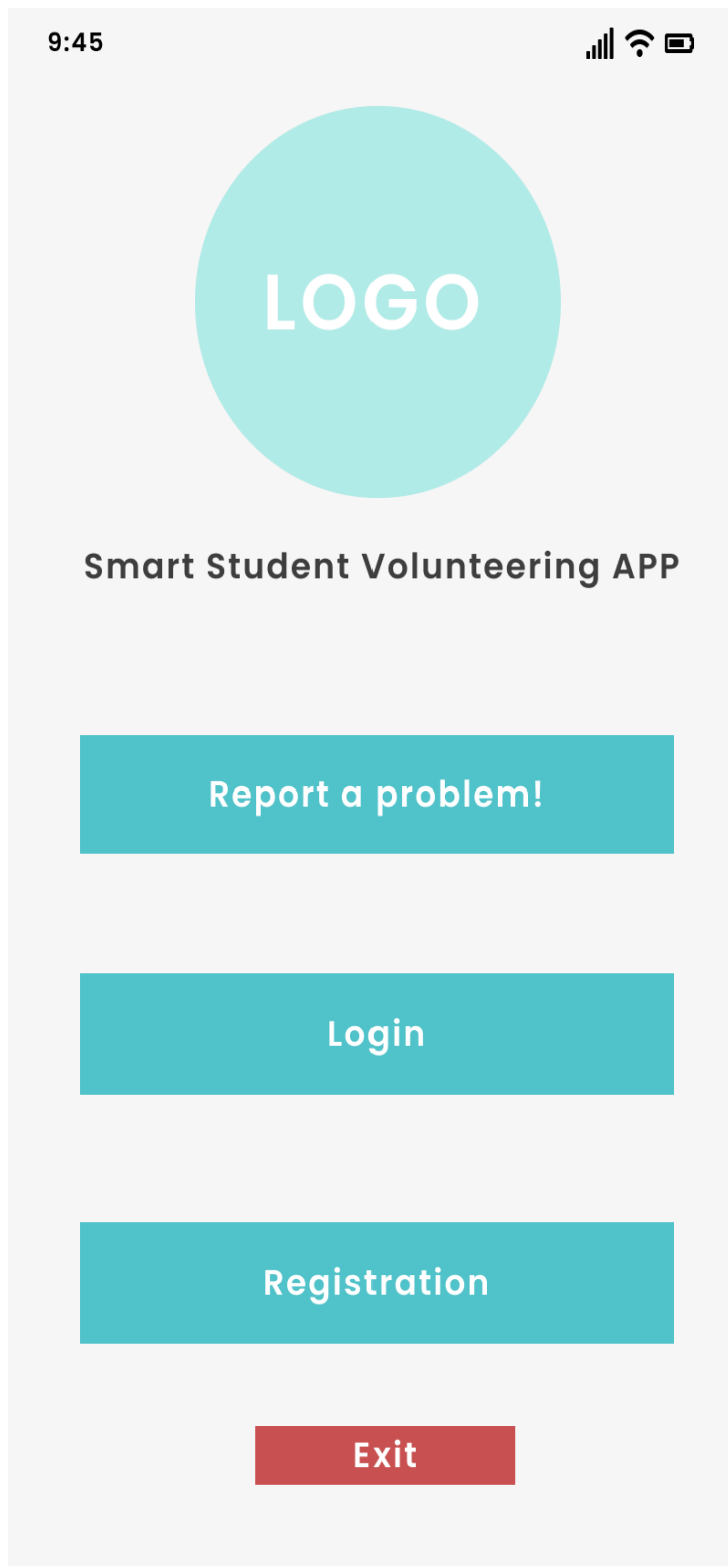
Precondition: Help must be requested for problem and volunteer must reach the problem location.

Cross-reference: 4.1, 8.2, 8.3

4.2 System Quality Attributes

- **Security:** Security shall be moderate as no crucial information is shared by user or volunteer.
- **Portability and Compatibility:** The system shall run on windows, browsers and android based mobile phones.
- **Performance:** The system shall support 1,000 users per hour by responding within 2-3 seconds depending on platform.
- **Scalability:** The system shall be able to support 3,000 users maximum at the same time while maintaining optimal performance.

4.3 UI/UX




9:45



Hello! VOLUNTEER NAME

Requested Problems:

img

Description

img

Description

SEE MAP

CHOOSE PROBLEM

REJECT PROBLEM

9:45



Login

LOGO
HERE

Enter your email

Enter password

Login

Exit

9:45



Name

Email

NID

DOB

Institute

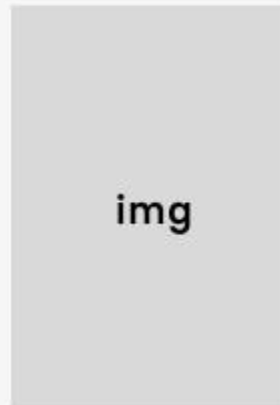
Reference

Free Time

Change INFO

RETURN

9:45



Enter your full name

Enter your DOB

Enter Institute

Email

New password

Confirm password

Upload new
Photo

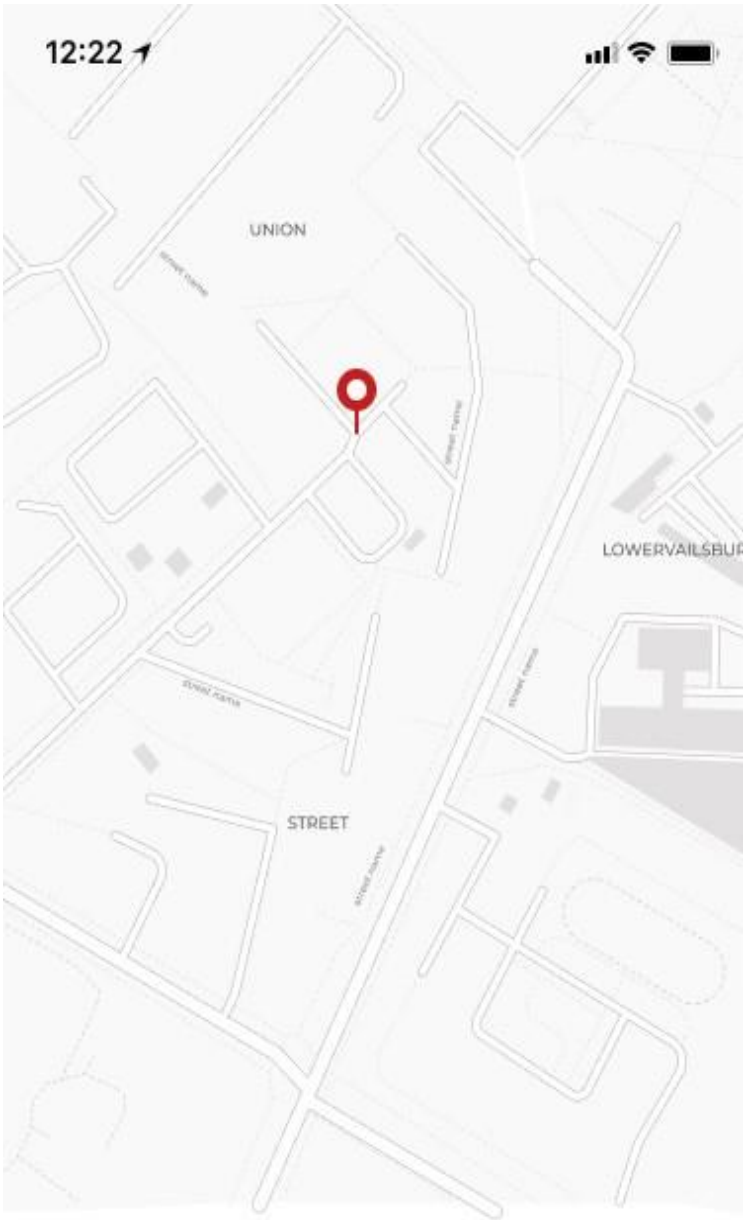
Delete
Account

Free time

preferred work

Save Edit

Exit without saving



Request

Add Photo

9:45

Problem Status

Problem ID:

Category:

View Photos

Problem Severity:

No. of volunteers present:

Estimated time of completion:

Problem feedback by volunteers

Feedback

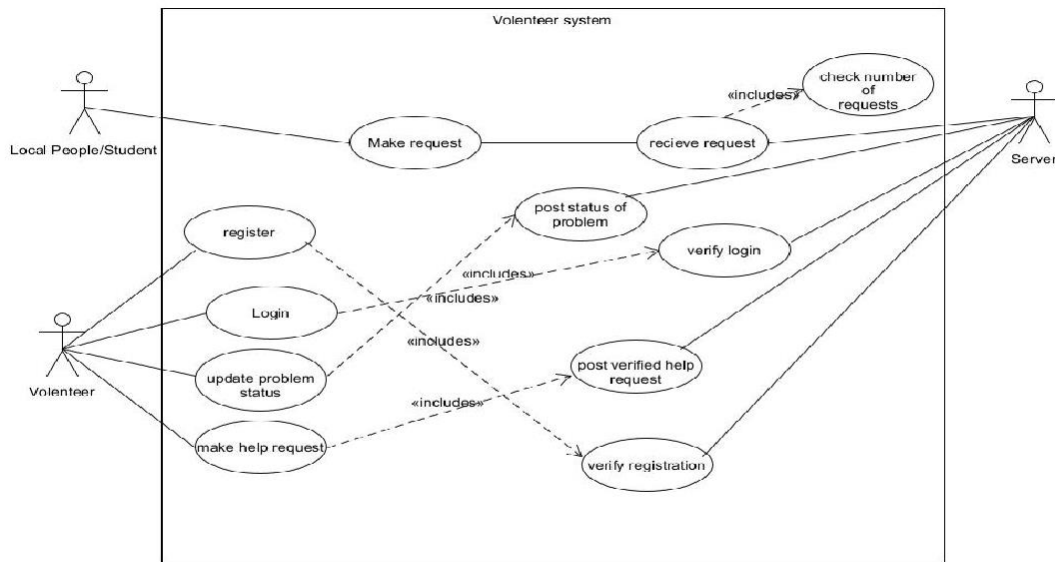
Progress bar:

55%

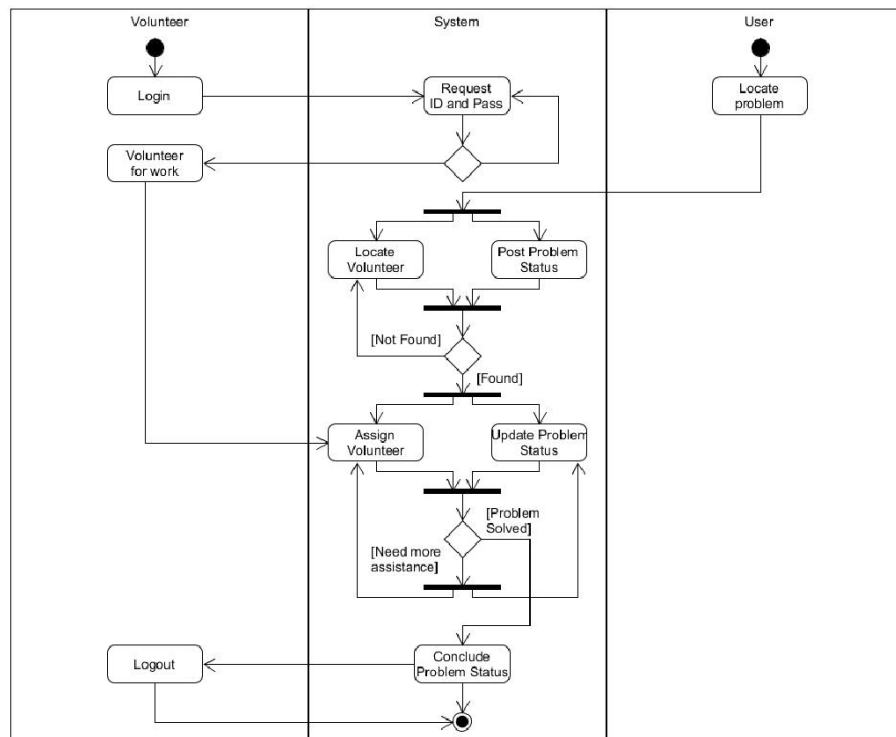
View Volunteer List

4.4 System Interface

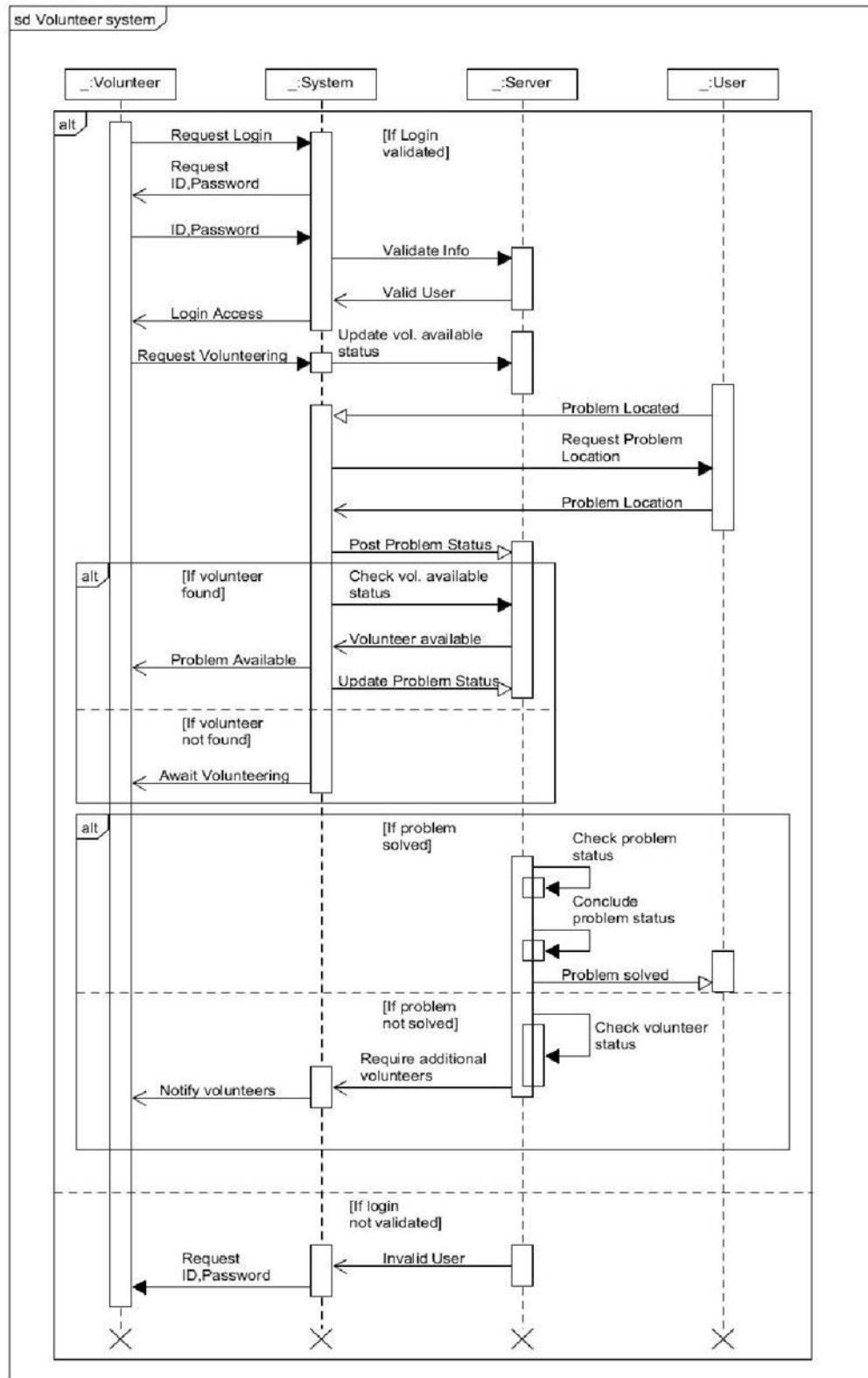
1. Use case Diagram



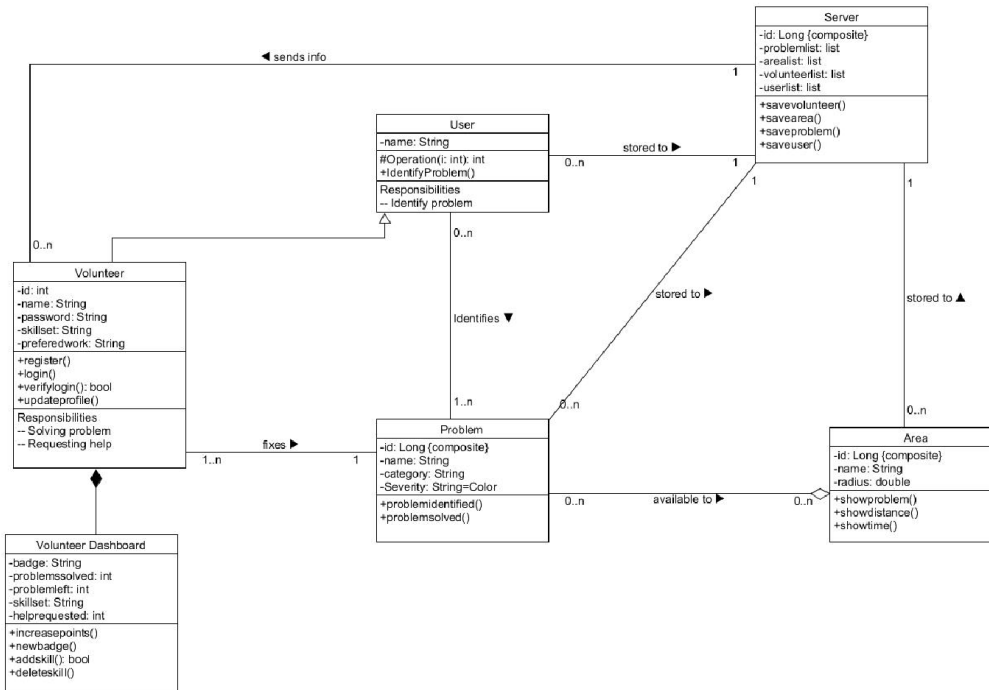
2. Activity Diagram



3. Sequence Diagram



4. Class Diagram



4.5 Project Requirements

An algorithmic software cost estimating methodology is the Constructive Cost Model (COCOMO).

Constructive Cost Model

We are assuming that the SLOC (Source Lines of Code) that we require here after analyzing all the components.

SLOC = 10,000

Now we need to figure out the effort, development time, and required number of people.

Suppose that, our software project type is organic, the values of the Coefficient<Effort Factor> = 2.4

P = project complexity = 1.05

SLOC = 10,000

T= SLOC-dependent coefficient = 0.38

Now,

Effort = PM = Coefficient*(SLOC/1000)^P

$$PM = 2.4*(10000/1000)^{1.05}$$

$$= 26.93$$

Development Time= DM= 2.5*(PM)^T

$$= 2.5*(26.93)^{0.38}$$

$$= 8.74$$

$$= 9 \text{ [In months]}$$

Required Number of People = ST

$$= PM/DM$$

$$= 26.93/8.74$$

$$= 3.08 \sim 3$$

Budget Estimation

Duration in weeks = 9*4 = 36 weeks Office days

Per week working days = 5 days

Working hours = 7 Hours

So, per week working hours is = (5*7) hours = 35 hours

So Total Working hours is = 35*36= 1260 hours.

Developer salary is = 500 Taka/ Hour

Total developers Salary = (500* 1260) = 6,30,000 Taka

Tester salary is= 300 Taka/Hour

Total tester Salary = (300* 1260) = 3,78,000 Taka

Expense	Amount	Total Amount
Developer salary		6,30,000
Tester salary		3,78,000
Requirement analysis Cost (1 month)	22*8*400	70,400
Transportation Cost		10,000
Office rent (9 months)	9*25000	2,25,000
Hardware Expense		80,000
Maintenance cost (4 months)	16*9*1200	1,72,800
Training Cost		10,000
Project Manager salary (9 months)	25,000*9	2,25,000
Total Estimation		18,00,400

5. FEATURES NOT TO BE TESTED

The following is a list of the areas that will not be specifically addressed.

- The “Notify Volunteers” feature uses GPS for volunteers to track the problem in the app, the GPS will be added from google. So, this feature will not be tested.
- Maintenance of the system
- Network facilities

6. TESTING APPROACH

Usually, a project requires all kind of testing properly for its success. But in this scenario, we are going to focus on **Unit testing** and **Integration testing**.

6.1 Testing Levels

- **Unit Testing:** Unit testing will be performed initially as we construct our system. During this testing, we will examine each software module to determine whether or not it has errors. The software developers and QA team use this testing approach. The aim of this testing is to confirm that each piece of software code functions as planned. We will use the "White Box Testing" method in this phase.

First, we will perform static unit testing in which, we will review the code of each unit. Our author will check the readiness of a unit according to its completeness, readability, complexity etc. Then the reviewer will develop the CRs and assign them with priorities, author and also a deadline. After that, the author will work on the addressed CRs to improve the unit and fix unwanted issues. Lastly, all CRs are validated using regression testing and A summary report is generated for stakeholders about the changes.

Then Dynamic unit testing will begin. Dynamic unit testing requires the unit to be executed in isolation. We will use Selenium IDE to check every statement in the units. Emulators and test drivers will be used to test every different unit. After testing and identifying errors, we will use Backtracking to debug the errors. As our program is relatively small in size, Backtracking would be the best technique to use in debugging.

We have 10 units in our application. We will give 1 working day for each unit to test and another 2-3 days to resolving CRs and Debugging.
- **Integration Testing:** We will next do the Integration in the second section. We will ensure that all the software components are logically connected, tested collectively, and are functional during this testing. Finding problems in how separate software modules interact when they are integrated is the aim of this level of testing. We'll use the "Bottom-up Integration" method in this phase.

Bottom-up integration method will first integrate our low-level modules with one another and then move up to integrating higher ones. We have 10 modules that we will test individually in unit testing. Then we will integrate the lower modules that doesn't invoke other modules. After their integration, we will find next lower modules and continue doing the same. While doing the process, we will use test drivers to simulate higher modules. After integration is done, we will replace the module with actual model and integrate it with the previous integrated ones. We will integrate all the modules in this way.
- **System Testing:** After that, we'll test the system. We will test a fully integrated, full-featured system through system testing. Then, we'll make sure it complies with all the requirements. Testing that uses a black box falls under this category. Therefore, we will use the "Black Box Testing" method at this level.
- **Acceptance Testing:** Acceptance Testing is the final stage of our testing. We'll do this testing to see if our product is acceptable. This test will be carried out to see whether any flaws were overlooked during the functional testing stage. We will use the "Black Box Testing" methodology at this stage.

6.2 Test Tools

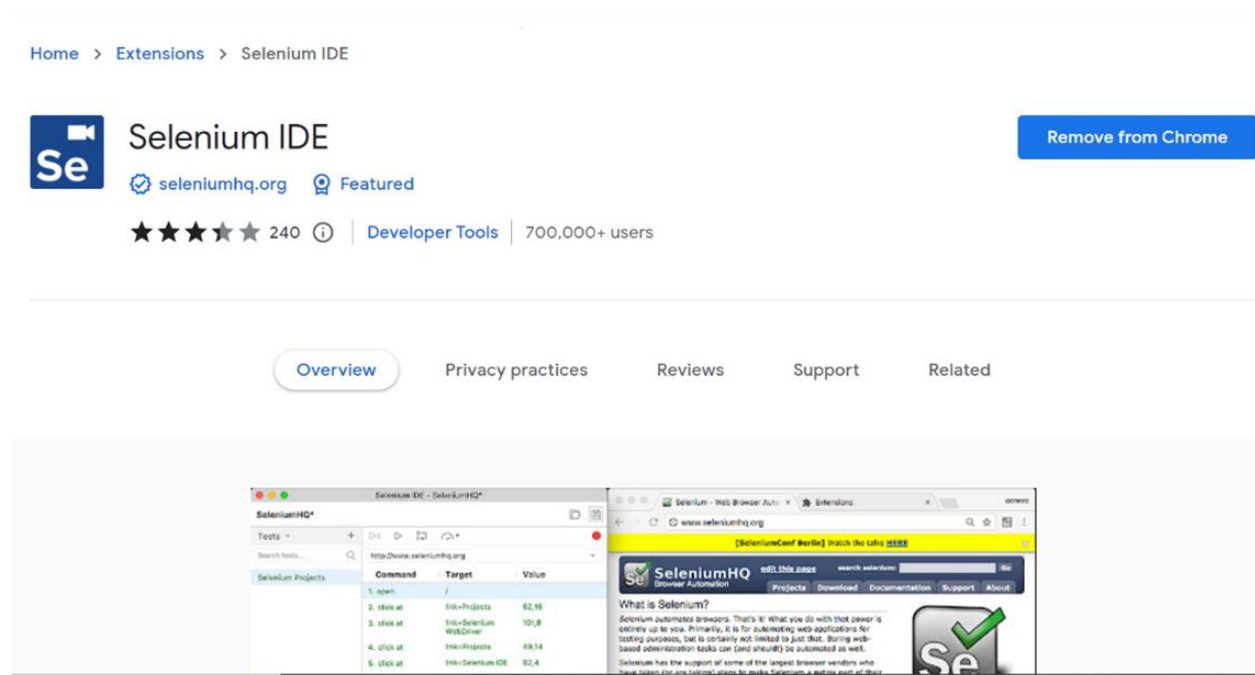
For the testing task, we will use the Selenium IDE for automating our website to test.

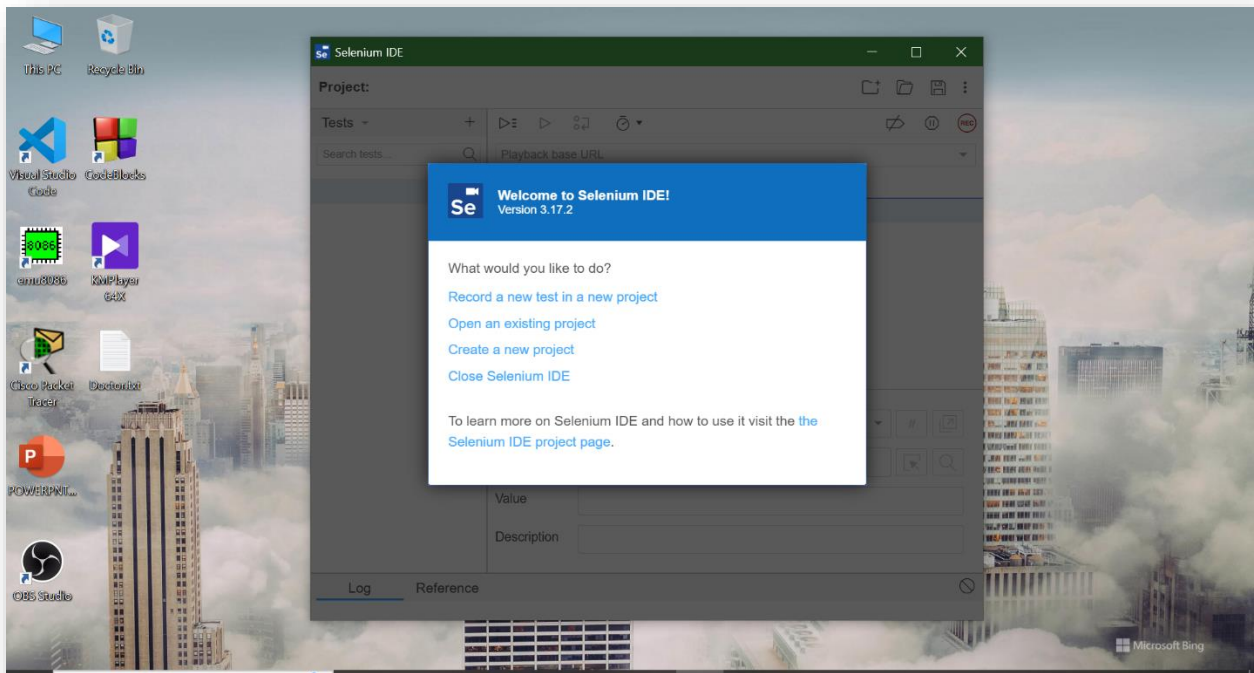
The Selenium IDE is a browser extension tool used for creating and executing automated test cases for web applications. It is a simple, yet powerful tool that allows testers to record, edit, and replay tests without writing any code. Users can easily and quickly create tests using Selenium IDE's user-friendly interface. The tool keeps track of how the user interacts with the browser, including button clicks, text inputs, and page navigation. The generated test scripts are then based on the interactions that were recorded. Selenium IDE also includes features such as assertions, which allow testers to verify that specific elements or values are present on a page. It also provides debugging capabilities, enabling users to step through their test cases and identify any errors.

In addition, Selenium IDE supports test automation on multiple browsers, including Chrome, Firefox, and Edge, and is compatible with both Windows and Mac operating systems.

Having all these advantages, we will use this tool for our automation testing purpose.

Below are the screenshots of the installation process of the selenium IDE.





6.3 Meetings

One of the most crucial elements of developing a successful testing team is assigning the correct work to the appropriate person. Therefore, it is essential to have regular team meetings to build an efficient project. The test team will consequently convene once a week to assess each member's development and decide whether or not they have finished their assigned responsibilities and finding out if there are any problems with testing will be easier. If there is a small bug, the entire team will try to fix it and try to find a solution as quickly as possible. The development and project manager will meet with the testing team head every two weeks. There will be CR assessment meetings in Unit Testing. Also, an emergency meeting can be scheduled if one is necessary.

7. TEST CASES/TEST ITEMS

1. Registration:

Project Name: Online Volunteer Management System	Test Designed by: Khandoker Mostakim Kabir
Test Case ID: FR_1	Test Designed date: 14/12/23
Test Priority (Low, Medium, High): High	Test Executed by:
Module Name: Registration	Test Execution date:

Test Title: Verify Registration process				
Description: Test Volunteers account creation process				
Precondition (If any): none				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Open the app 2. Go to Registration 3. Enter First Name, Last Name, DOB, Institute, NID/BRN, NID image, Email, Password, Self-image, Preferences, Free time as needed. 4. Tap Complete Registration	Md. Rahim 26.6.199 9 AIUB 239465780923 *Image files* Rahim@gmail.com I2jKl Rescuing Pets 8am-12pm	Volunteer should be able to create an account		
Post Condition: Volunteer will be registered, and volunteer info will be updated into the database				

2. Volunteer Profile:

Project Name: Online Volunteer Management System		Test Designed by: Khandoker Mostakim Kabir		
Test Case ID: FR_2		Test Designed date: 14/12/23		
Test Priority (Low, Medium, High): Medium		Test Executed by:		
Module Name: Volunteer Profile		Test Execution date:		
Test Title: Verify profile edit function				
Description: Test volunteer profile’s editing function				
Precondition (If any): Volunteer must be logged in				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
5. Go to profile info 6. Go to change info 7. Enter Name, Pictures,	Md.Rahim 26.6.1999 AIUB Rahim@gmail.com	Volunteer should be able to change their info		

DOB, Institute, Email, Password, Preferences, Free time as needed 8. Enter current password 9. Tap save edit	I2jKl Rescuing Pets 8am-12pm Image file R@h1m			
Post Condition: Volunteers profile info will be updated into the database.				

3. Problem Locating and Requesting:

Project Name: Online Volunteer Management System		Test Designed by: Khandoker Mostakim Kabir			
Test Case ID: FR_3		Test Designed date: 14/12/23			
Test Priority (Low, Medium, High): High		Test Executed by:			
Module Name: Problem locating and requesting		Test Execution date:			
Test Title: Verify users Problem Request process					
Description: Test User’s Request function					
Precondition (If any): None					
Test Steps		Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Go to Report a problem 2. Go to map 3. Select problem location 4. Go to Category 5. Select problem category 6. Select Add Photo(optional) 7. Add a Photo(optional) 8. Tap Request		Bashundhara Block-F, Road -118 Water Blockage An image file(optional)	User should be able to request for a problem		
Post Condition: A Job is added in the Requested Problems tab for volunteers and the information is uploaded into the database					

4. Show Problem:

Project Name: Online Volunteer Management System	Test Designed by: Limia Sadina Sathi
Test Case ID: FR_4	Test Designed date: 15/12/23
Test Priority (Low, Medium, High): High	Test Executed by:

Module Name: Show problem			Test Execution date:	
Test Title: Requested Problem shown				
Description: Test problem shown				
Precondition (If any): The volunteer should be logged in				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Open profile 2. Go to problem list	1. Water blockage in Bashundhara 2. Unhygienic environment in Kuratoli	User should be able to view problems		

5. Notify Volunteer:

Project Name: Online Volunteer Management System		Test Designed by: Limia Sadina Sathi		
Test Case ID: FR_5		Test Designed date: 15/12/23		
Test Priority (Low, Medium, High): High		Test Executed by:		
Module Name: Notify Volunteer		Test Execution date:		
Test Title: Receiving Volunteering Notification				
Description: Test about notification for volunteer request is being received by the volunteers.				
Precondition (If any): Volunteer Request has been placed in volunteer’s location. Location must be turned on.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. App searches for volunteers nearby for dummy problem. 2. App identifies potential volunteers. 3. Problem information from the database is sent along with the option for accepting or rejecting request.	Problem ID:22 Category: Water Clog Location: Kuratoli Bazar, Dhaka Severity: High	Phone receives notification along with the pop-up containing problem information and option to accept or reject volunteering.		

Post Condition: Problem information can be viewed along with the map location.
--

6. Show Location and Estimated Time:

Project Name: Online Volunteer Management System		Test Designed by: Nafisa Maliyat		
Test Case ID: FR_6		Test Designed date: 17/12/23		
Test Priority (Low, Medium, High): High.		Test Executed by:		
Module Name: Show ETA & location		Test Execution date:		
Test Title: Verify if location is shown accurately				
Description: Test if the location of requested problem is shown				
Precondition (If any): Location request received.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Enter dummy problem 2. See if location procced accurately	Problem Id: XX Traffic congestion Location: Kuril	Location should be shown		
Post Condition: Location is updated				

7. Problem Status:

Project Name: Online Volunteer Management System			Test Designed by: Nafisa Maliyat	
Test Case ID: FR8_1			Test Designed date: 17/12/23	
Test Priority (Low, Medium, High): High			Test Executed by:	
Module Name: Problem Status			Test Execution date:	
Test Title: Problem Request Processing				
Description: Process a requested problem				
Precondition (If any): Request verified as real				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)

1. Request Help For problem	Problem Id: XX Animal rescue Multiple animals	Problem severity increased		
Post condition: Request is updated accordingly				

8. Request Volunteer:

Project Name: Online Volunteer Management System		Test Designed by: An Nazmus Sakib		
Test Case ID: FR9		Test Designed date: 18/12/23		
Test Priority (Low, Medium, High):		Test Executed by:		
Module Name: Request Volunteers		Test Execution date:		
Test Title: Validate the Request Volunteer Feature				
Description: Test to validate the request additional volunteer Feature for volunteers while volunteering				
Precondition (If any): The volunteer must be solving a problem that he accepted.				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Volunteer presses the request volunteer option along with additional needs. 2. The progress of the problem is updated in the database. 3. App then notifies other volunteers nearby based on the requirement of the problem and along with the problem progress information as well as additional needs.	Problem ID:22 Category: Water Clog Location: Kuratoli Bazar, Dhaka Severity: High Progress: 75%	Volunteer request option will send notification to other volunteers nearby based on the requirements of the problem and needs of the working volunteer.		
Post Condition: The progress is hampered; the fixing of the problem is delayed.				

9. Notify Local Administrator:

Project Name: Online Volunteer Management System		Test Designed by: An Nazmus Sakib		
Test Case ID: FR10_1		Test Designed date: 18/12/23		
Test Priority (Low, Medium, High): Medium		Test Executed by:		
Module Name: Notify local administrator		Test Execution date:		
Test Title: Automated Email Sending				
Description: Test e-mail sent to local administrator				
Precondition (If any): Help must be requested for problem and volunteer must choose the problem				
Test Steps	Test Data	Expected Results	Actual Results	Status (Pass/Fail)
1. Show problem and its location 2. Show necessary helps needed 3. Tap sent e-mail	1. Traffic jam in Kuratoli	E-mail must be sent to the local administrator		

8. ITEM PASS/FAIL CRITERIA

Testing team will be a group of -

- Unit Test Manager
- Test Analyst
- Developer

The team will be doing the following tasks-

- There will be error free codes and in time compilation
- The test will pass only if there is 100% passing rate.
- If any error found or passing rate remains below 100% it will be resolved in time.
- All the possible tests will be documented

Here we have implemented 24 test cases. 75% of the test cases were passed successfully when we applied the test case and 25% were fail. Because of some query related issues on the database, the test cases were failed. When the test case was applied after solving query related problem, all the test cases are successfully passed.

9. TEST DELIVERABLES

- **Acceptance test plan**

The user acceptability tests all turned out to be successful. The user interface was easy to use but still efficient. The UAT thus went off without a hitch.

- **System/Integration test plan**

Every aspect of system integration was done successfully. As a result, the database was operating correctly and all of its features were responsive.

- **Unit test plans/turnover documentation**

The unit testing was done, and all were working without any bug.

- **Report mock-ups**

There were no mock-ups created for the report we are currently reviewing because it is the project report.

- **Defect/Incident reports and summaries**

It functions well in circumstances where the project requirements are not fully understood. The process is iterative and based on trial and error between the client and developer.

10. STAFFING AND TRAINING NEEDS

For the project's system/integration and acceptance testing phases, it is desirable that at least one (1) full-time tester be allocated to the task. In order to participate in evaluations, meetings, etc., a person will need to be assigned part-time at the start of the project. After around six months, they will be assigned full-time. The project manager/test manager will take on this responsibility if a separate test person is not readily accessible. A full-time professional communicator should be required. sufficient numbers of operation staff should be needed. The following training needs to be handled in order for this project to be effective.

- The operation employees will need to receive communication and language training from a professional communicator.
- Instruction on the fundamental functions of the EDI interface will be required for the developers and testers for the project's final acceptance.

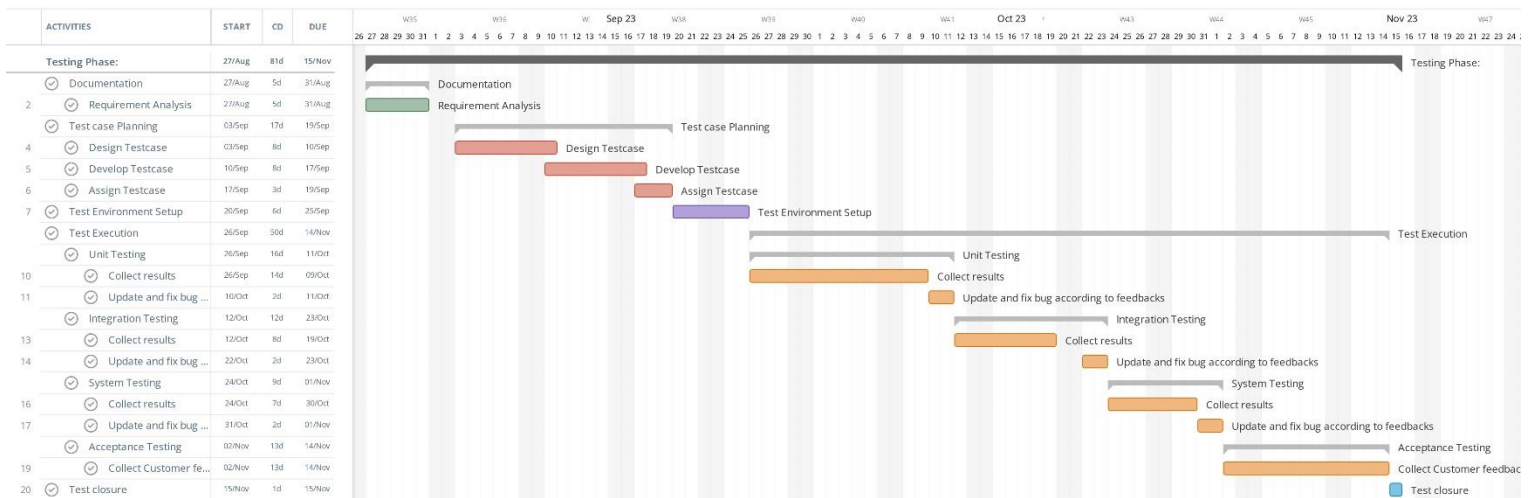
11. RESPONSIBILITIES

	TM	PM	Dev Team	Test Team	Client
Acceptance test Documentation & Execution	*	*		*	*
System/Integration test Documentation & Execution	*		*	*	
Unit test documentations & Execution	*		*	*	
System Design Reviews	*	*	*	*	*
Detail Design Reviews	*	*	*	*	
Test Procedures and rules	*	*	*	*	
Change control and regression testing	*	*	*	*	*

12. TESTING SCHEDULE

Online volunteer management
Read-only view, generated on 26 Aug 2023

Instantantt



13. PLANNING RISKS AND CONTINGENCIES

S/N	Risk Description	Probability	Impact	Mitigation Plan
1	If the system gets unpopular	60%	Delay project 3 weeks	Taking feedback from users then updating the system
2	If Customer changes some requirements	50%	Delay project 2 weeks	Fulfilling customer requirements
3	If some of staffs are unfamiliar with the system	40%	Delay project 2 weeks	Training the staffs
4	We might require more staffs	40%	Delay project 1 weeks	Hiring new staffs
5	There might be few bugs in the system	40%	Delay project 1 weeks	Reduce bugs
6	The users find the system hard to use	60%	Delay project 4 weeks	Arrange customer service to help users

14. APPROVALS

Project Sponsor	PASS
Development Management	PASS
EDI Project Manager	PASS
RS Test Manager	PASS
RS Development Team Manager	PASS
Reassigned Sales	PASS
Order Entry EDI Test manager	PASS