**<Community Service Volunteer Tracking System>**

Software Requirements Specification

Version <1.0>

**MATHEBULA, NN (47489189)**

**SHEZI, P (45225524)**

**KONDILE, SESETHU (41590503) MZOBE, BAYA (44027516)**

**NYEMBEZI, T (40183955)**

**SEDZE, TK (52827097)**

**KUMANE, QN (45097208)**

**Submitted in partial fulfillment of the requirements of**

**CMPG 224, Software Engineering**

**22/08/2025**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <22/08/25> | <1.0> | SRS 1.0 | Group 6 |
| <xx/xx/xx> | <2.0> | SRS 2.0 |  |

Table of Contents

Introduction 1.0 5

Purpose 1.1 5

Scope 1.2 5

Definitions and Acronyms 1.3 5

References 1.4 5

System OVERVIEW 2.0 6

System Description 2.1 6

System Features 2.2 6

System Actors and roles 3.0 7

Actors 3.1 7

Roles 3.2 8

Use case diagram 3.3 8

FUNCTIONAL Requirements 4.0 9

F & FR 3.1 9

F & FR 3.2 9

F & FR 3.3 9

F & FR 3.4 10

F & FR 3.5 10

F & FR 3.6 #

F & FR 3.7 #

Non-functional Requirements 5.0 11

Usability 5.1 11

Performance 5.2 11

Security 5.3 11

Reliability 5.4 11

etc.

Usecases/scenarios 6.0 12

Use case Name 4.1 12

Use case Name 4.2 12

Use case Name 4.3 13

Use case Name 4.4 13

Use case Name 4.5 14

Use case Name 4.6 #

etc

Assumptions and Constraints 7.0 15

Assumptions 7.1 15

Constraints 7.2 15/16

aPPENDIX 8.0 17

**Introduction**

* 1. **Purpose**

The Community Service Volunteer Tracking System (CSVTS) will make it easier for organizations, coordinators, and volunteers to manage and keep track of community service activities. The system provides a central platform where volunteers can register, log their hours, and stay updated on opportunities, while administrators can oversee participation, verify contributions, and generate useful reports.

* 1. **Scope**

This system focuses on handling volunteer registration, tracking service hours, monitoring project involvement, and sending timely reminders or notifications. The CSVTS is not intended to manage unrelated organizational tasks outside community service tracking. Its primary goal is to streamline volunteer management, ensuring transparency, accountability, and efficiency in community service programs.

**1.3 Definitions and Acronyms**

Concept Definition

CSVTS Community Service Volunteer Tracking System

UI User Interface

**1.4 References**

Volunteer Matters – Volunteer Management Software  
 <https://www.volunteermatters.com>

Galaxy Digital – How to Choose Volunteer Management Software <https://www.galaxydigital.com/blog/how-to-choose-a-volunteer-management-software>

Track It Forward – Volunteer Time Tracking

<https://www.trackitforward.com>

Forecast – What is Volunteer Time Tracking?

<https://www.forecast.app/learn/what-is-volunteer-time-tracking>

Kindful – Volunteer Management Software for Nonprofits

<https://kindful.com/nonprofit-glossary/volunteer-management-software-for-nonprofits>

Wikipedia – Non-profit Technology  
 <https://en.wikipedia.org/wiki/Non-profit_technology>

**System Overview**

**2.1 Product description**

The Community Service Volunteer Tracking System (CSVTS) is a centralized, web-based platform designed for NGOs and government agencies to manage volunteer information and participation. The system reduces the need for manual record-keeping by providing a digital hub where volunteer profiles, tasks, service hours, and achievements can be stored and retrieved directly and efficiently. Instead of using papers or spreadsheets, everything is stored in the system. It simplifies the coordination between organizations and volunteers by allowing users to register, log hours, track performance, and generate reports. On the other side, administrators can assign tasks, monitor engagement, and analyse contributions, and thus improving decision-making and accountability in community service programs. The goal is to make volunteer management easier, faster, and more reliable for both volunteers and organizations.

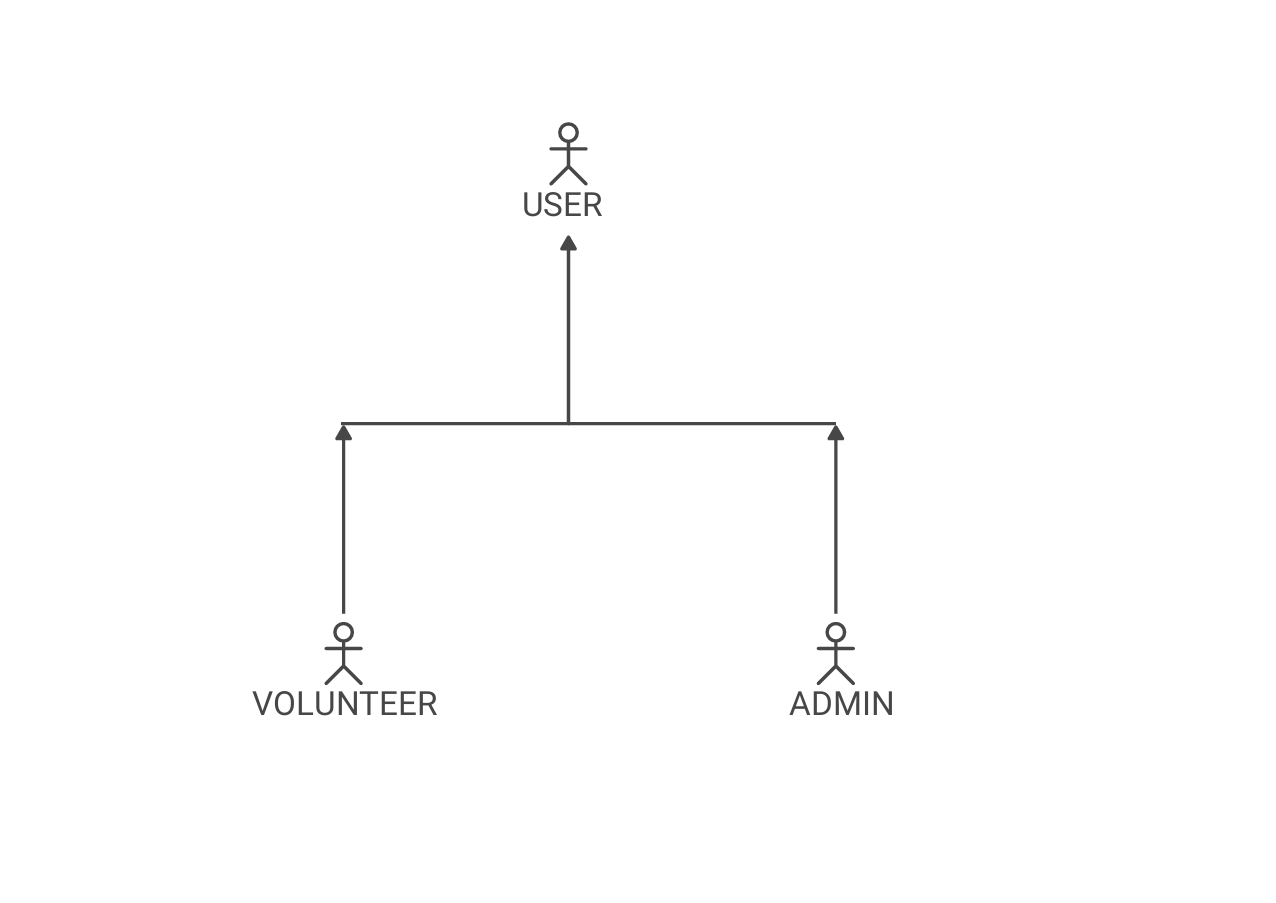
**2.2 System features**

* **Volunteer Registration and Profiles** – Volunteers can sign up and create accounts with personal details, skills, and their areas of interest.
* **Task Assignment and Tracking** – Administrators can assign tasks to volunteers and be able to track their progress up to the completion.
* **Automated Volunteer Hour Logging** – Service hours are automatically recorded and stored for accurate reporting.
* **Performance Monitoring** – Track the overall volunteer participation, contributions, and achievements over time, how active each volunteer has been.
* **Volunteer List Display** – Provide an overview of all registered volunteers for easy access and management.
* **Secure Access Control** – Different levels of access for volunteers, administrators, and managers to ensure data protection and integrity.
* **Reporting and Insights** – Generate detailed reports on volunteer activity and program effectiveness

**System Actors and Roles**

**3.1. Actors**

|  |  |  |
| --- | --- | --- |
| **Actor** | **Role/Description** | **Interaction with System** |
| Administrator | Manages the entire volunteer and task database. | Logs in, adds/edits/deletes volunteers and tasks, views all system data. |
| Volunteer | A person who signs up to be a volunteer for Non – profit organization | Logs in, views and updates their personal profile, and views their assigned tasks. |



**3.2 Roles**

|  |  |
| --- | --- |
| **Volunteer** | **Admin** |
| * Login to the system * Create and update personal profile * View assigned tasks * Update task status (e.g., mark as complete) * Search for available tasks | * Login to the system * Add, edit, and delete volunteer accounts * Create, update, and delete tasks * Assign tasks to volunteers * View all system data and reports |

**3.3 Use Case Diagram**

**Volunteer Management System**



ADMIN



VOLUNTEER

**Functional Requirements**

**3.1 Feature 1#**

**Description: Volunteer Profile Management**

**FR:**

* The system shall allow a volunteer to create a new profile with a unique username and a password.
* The system shall allow a volunteer to update their contact information, such as their phone number and email address.
* The system shall allow a volunteer to specify their skills and availability.
* The system shall display a volunteer's profile to them upon successful login.

**3.2 Feature 2#**

**Description: Task Management**

**FR:**

* The system shall allow an administrator to create a new task with a title, description, and deadline.
* The system shall allow an administrator to assign a task to one or more volunteers.
* The system shall allow an administrator to update the status of an assigned task for example “In Progress," "Completed" or “Not Completed”.
* The system shall allow an administrator to delete a task.
* The system shall provide a search and filter function for administrators to view tasks.

**3.3 Feature 3#**

**Description: Smart Search and Filtering**

**FR:**

* The system shall allow administrators to search for volunteers by specific skills for e.g. "First Aid," "Marketing," "Public Speaking".
* The system shall allow administrators to filter the list of volunteers based on their stated availability e.g. "Weekends," "Evenings".
* The system shall allow both administrators and volunteers to search for tasks by keywords present in the task title or description.
* The system shall allow administrators to filter tasks by their status e.g., "Assigned," "Pending," "Completed," "In Progress".
* The search functionality shall return results within 3 seconds, regardless of the number of records.

**3.4 Feature 4#**

**Description** **Time Logging and Reporting**

**FR:**

* The system shall allow a volunteer to submit the number of hours they have spent on a specific task.
* The system shall allow an administrator to approve or reject submitted hours.
* The system shall generate a report that summarizes total hours worked by each volunteer.
* The system shall track and display the total number of hours a volunteer has completed.

**3.5 Feature 5#**

**Description: Role-Based Access Control**

**FR:**

* The system shall require user authentication (login) before granting access to any feature.
* The system shall restrict all administrative functions (e.g., managing volunteers, managing tasks) to the Administrator role only.
* The system shall ensure a Volunteer can only view and edit their own profile and view their own assigned tasks.

**Non-Functional Requirements**

**5.1 Usability**

* Admin Usability - The system will have a simple and easy to use interface that allows administrators to manage volunteers and assign tasks.
* Volunteer Usability - The volunteer’s portal will have a clear and easy to use layout that allows volunteers to view their assigned tasks and update their profile details.

**5.2 Performance**

* The system will load pages and complete search queries within 3 seconds
* Database operations such as adding, updating, or deleting a record will be completed and confirmed within 1 second.

**5.3 Security**

* The system will provide a secure sign-up process where each volunteer can create their own unique account with a username (or email) and a password.
* The system will require username and password authentication for all administrative functions.
* It will protect sensitive volunteer data such as email addresses, personal details etc from unauthorized access.
* The system will validate all users
* The system will implement role-based access to ensure that a regular volunteer can only access their own profile and assigned tasks. Volunteers should not have access to administrative functions or other volunteers’ data.

**5.4 Reliability**

* The system will be available and operational 99.9% of the time.
* The system will handle invalid requests and unexpected errors gracefully, without crashing or corrupting data.

**Use cases/Scenarios**

Table x: (Login)

|  |  |
| --- | --- |
| **Use case name** | Login |
| **Actors** | Administrator, Volunteer |
| **Preconditions** | The user has a valid username and password. |
| **Flow of events** | 1. The user navigates to the login page 2. The user enters their username and password 3. The user clicks the "Login" button 4. The system validates the credentials 5. The system grants access and redirects the user to their respective dashboard |
| **Post Conditions** | The user is successfully authenticated and logged in |

Table x: (Manage Volunteers)

|  |  |
| --- | --- |
| **Use case name** | Manage Volunteers |
| **Actors** | Administrator |
| **Preconditions** | The Administrator is logged into the system |
| **Flow of events** | 1. The Administrator navigates to the "Manage Volunteers" section 2. The system displays a list of all volunteers 3. The Administrator selects an action (add, edit, or delete) 4. The Administrator enters or modifies the volunteer's details 5. The system performs the requested action and confirms the change |
| **Post Conditions** | The volunteer database is updated, and the list of volunteers reflects the changes. |

Table x: (Manage Tasks)

|  |  |
| --- | --- |
| **Use case name** | Manage Tasks |
| **Actors** | Administrator |
| **Preconditions** | The Administrator is logged into the system |
| **Flow of events** | 1. The Administrator navigates to the "Manage Tasks" section 2. The system displays a list of all tasks 3. The Administrator selects an action (create, edit, or delete) 4. The Administrator enters or modifies the task details and assigns it to a volunteer 5. The system performs the action and confirms the update |
| **Post Conditions** | The tasks database is updated, and the task list reflects the changes |

Table x: (View Assigned Tasks)

|  |  |
| --- | --- |
| **Use case name** | View Assigned Tasks |
| **Actors** | Volunteer |
| **Preconditions** | The Volunteer is logged into the system |
| **Flow of events** | 1. The Volunteer navigates to their personal dashboard 2. The system retrieves and displays a list of tasks assigned specifically to that volunteer 3. The Volunteer can view the details for each task (description, deadline) |
| **Post Conditions** | The Volunteer has successfully viewed their assigned tasks |

Table x: (Manage Profile)

|  |  |
| --- | --- |
| **Use case name** | Manage Profile |
| **Actors** | Volunteer |
| **Preconditions** | The Volunteer is logged into the system |
| **Flow of events** | 1. The Volunteer navigates to their profile page 2. The system displays the volunteer's current information 3. The Volunteer can edit their contact information and skills 4. The Volunteer clicks the "Save" button to submit changes 5. The system validates the input and saves the updated information to the database |
| **Post Conditions** | The Volunteer's profile information is successfully updated |

**Assumptions and Constraints**

* 1. **Assumptions**

1. User Competency

* It is assumed that administrators and volunteers possess basic computer literacy to interact with the system interfaces effectively.
* Volunteers and administrators will have access to devices (PCs, tablets, or smartphones) with supported browsers or applications.

1. Infrastructure Availability

* Reliable internet connectivity is assumed for both administrators and volunteers to access the system without interruption.
* Hosting infrastructure (servers and databases) will be maintained at a level that supports the specified performance and availability targets.

1. Authentication and User Management

* It is assumed that volunteers will register their own accounts using valid email addresses or unique usernames.
* The system will enforce unique user credentials for each volunteer account.

1. Development and Maintenance

* The system will be developed and maintained using industry – standard security practises
* Role-based access control policies will be predefined and enforced consistently.
  1. **Constraints**

1. Usability Constraints

* The system user interfaces must remain simple and intuitive for both administrators and volunteers, limiting complex functionality to maintain ease of use.
* Volunteer portals will restrict user access to only their assigned tasks and profile data, with no cross-access allowed.

1. Performance Constraints

* Page load times and search query responses must be completed within 3 seconds under normal operating conditions.
* Database operations (add, update, delete) must confirm success within 1 second.

1. Security Constraints

* Secure sign-up and login processes are mandatory and must include validation of user credentials.
* Role-based access control must strictly enforce separation of administrator and volunteer capabilities.
* Sensitive personal data of volunteers must be protected against unauthorized access at all times.
* The system must authenticate all users before granting access to their respective sections.

1. Reliability Constraints

* The system must achieve an uptime/service availability of 99.9%, including scheduled maintenance windows.
* The system must effectively handle invalid user requests and unexpected errors without crashing or data corruption.

**Appendix**