Sustainable Development Volunteer Service Website Project Design Document _Group 9

Instructor: Dr. Aidan Murphy

Honghao Liu 23219672

Haoming Xi 23219607

Yanxiang Ao 23219615

1. Overview

This project aims to build a volunteer service platform centered on "Sustainability", promoting collaboration among students, social organizations, and administrators to advance the sustainable development of public welfare activities. The website provides different functions and permissions for three types of users: Participants (Students), Organizations, Admins.
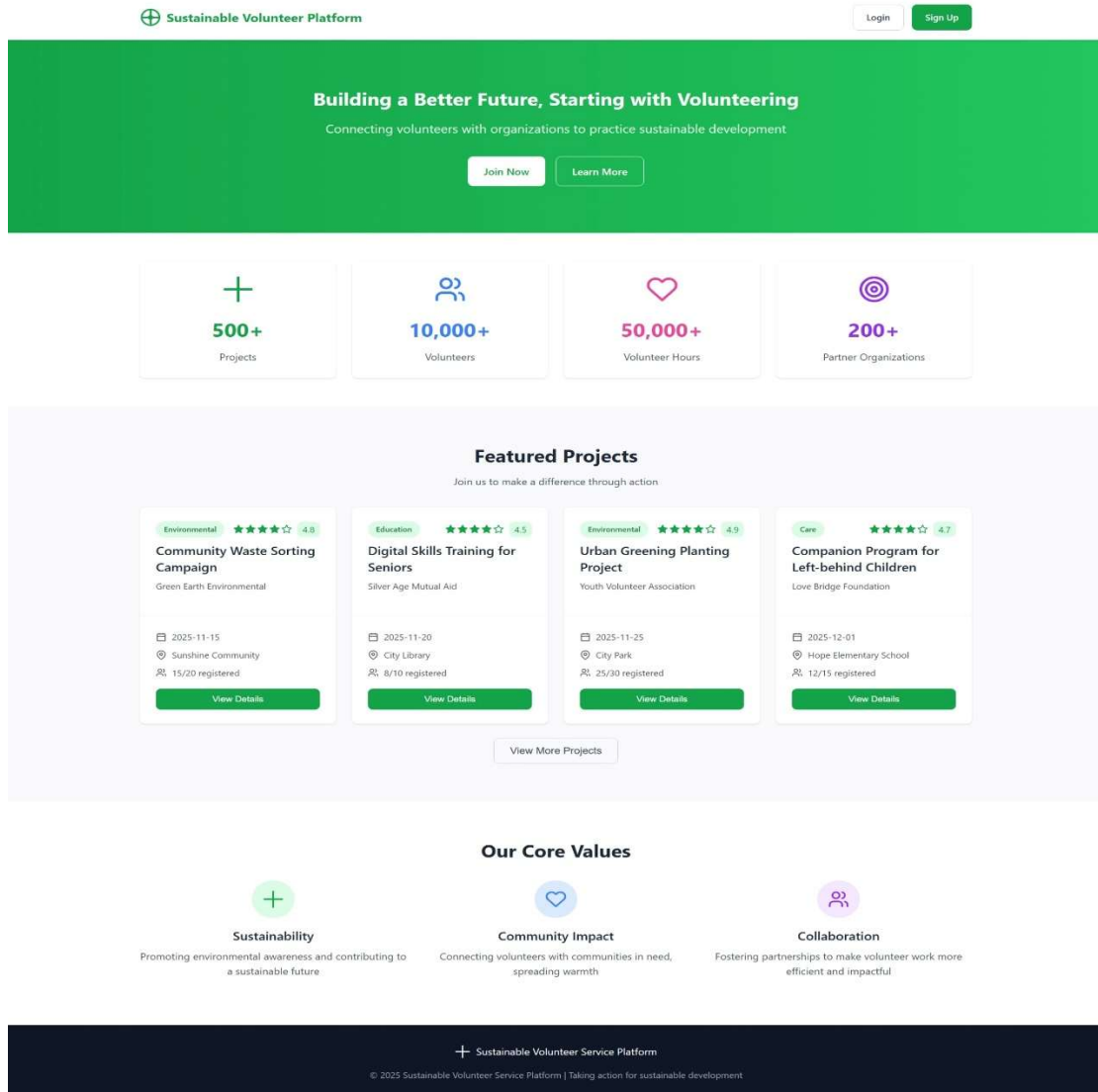
Core Objectives：

i. Encourage students to participate in sustainability-focused volunteer activities.

ii. Provide organizations with a platform to publish and manage projects.

iii. Ensure authenticity and integrity of projects through an admin review system.

iv. Digitally record and certify the volunteer service hours.

2. Objectives

i. Provide a digital volunteer service platform focused on sustainability.

ii. Implement a multi-user login system with different access permissions.

iii. Support project publishing, registration, and management functions.

iv. Record and certify students' volunteer service hours.

v. Ensure data security and access control.

vi. Deliver a clean, modern, and user-friendly web interface.

3. Functional Design & UI Prototype
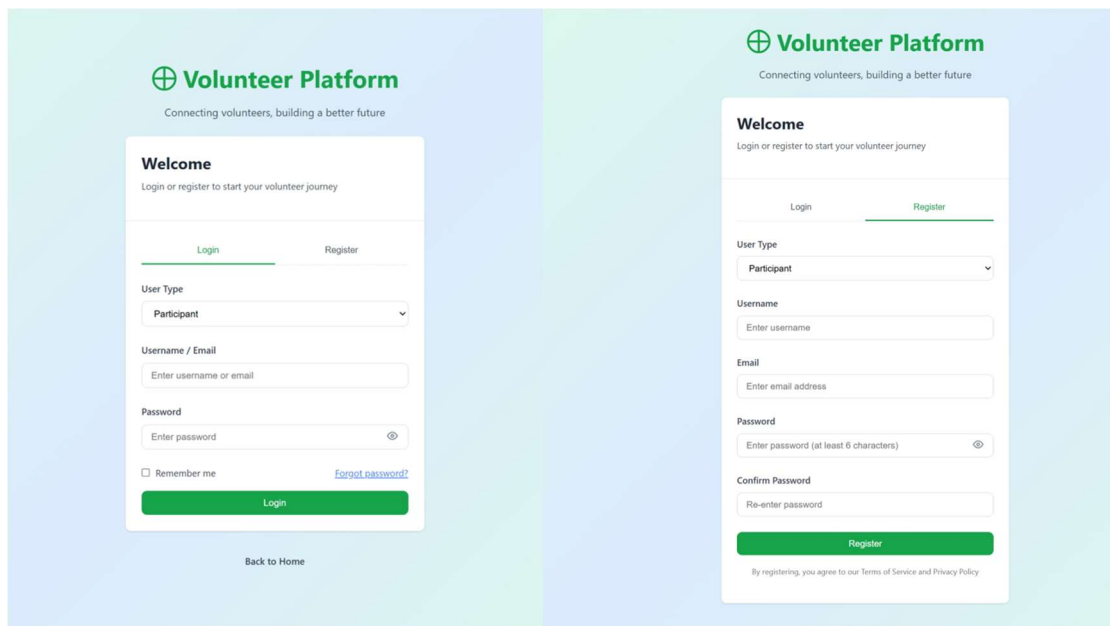
3.1 Home Page

3.1.1 Home Page Functional Design

i. Browse project summaries but cannot register or comment.

ii. Encouraged to sign up to gain full access.

3.1.2 Home Page Introduction

A sustainability-themed homepage (green tone), showcasing the mission, registration options, and project statistics. Includes "Join Now" and "Learn More" buttons, plus featured projects encouraging volunteer engagement.
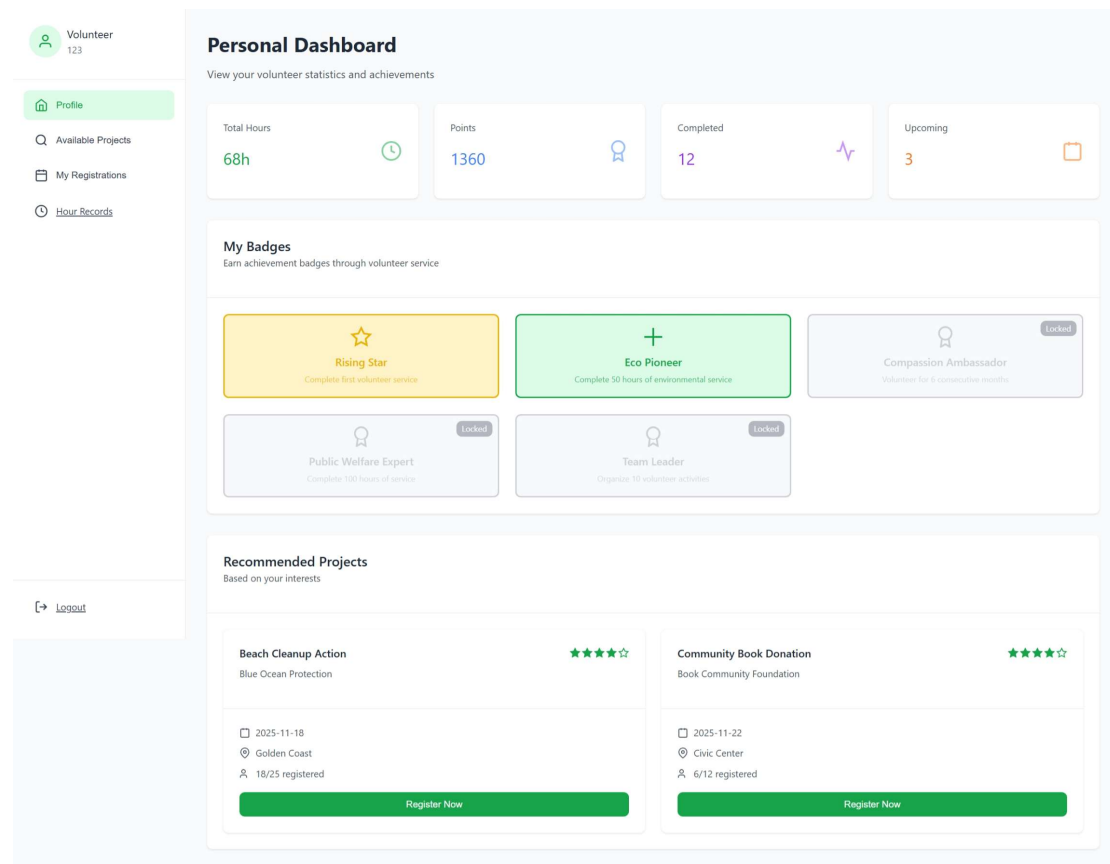
3.2 Login / Register Pages

3.2.1 Login / Register Pages Functional Design

i. Three user types: Participant / Organization / Admin.

ii. All users can log in, edit personal information, or delete their accounts.

iii. Passwords are encrypted (e.g., using BCrypt).

3.2.2 Login / Register Pages Introduction

i. Login Page: Users choose their role (Participant / Organization / Admin) and

log in via username/email and password.

ii. Register Page: New users select a role and enter username, email, and

password to create an account. The interface is clean and intuitive.
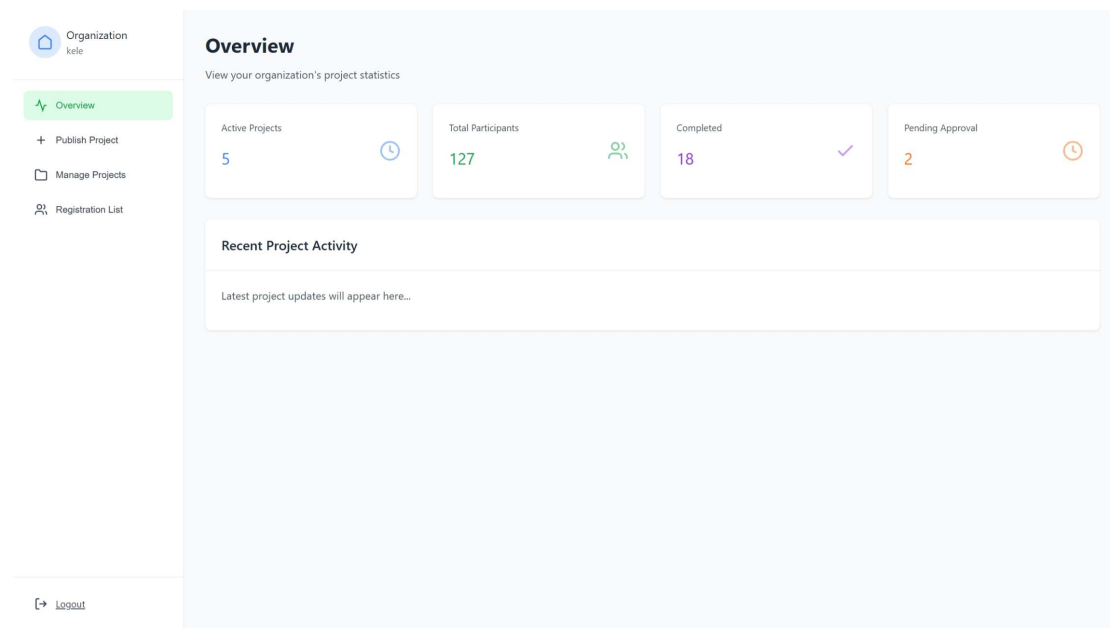
3.3 Participant Dashboard

3.3.1 Participant Dashboard Functional Design

i. Browse and register for volunteer projects.

ii. View project details (content, time, requirements).

iii. Check personal volunteer hours.

iv. Withdraw from unstarted projects.

3.3.2 Participant Dashboard Introduction

Displays volunteer data and achievements: total hours, points, completed and upcoming projects, and earned badges (e.g., Rising Star, Eco Pioneer). Sidebar provides quick access to available projects, registration history, and service records.
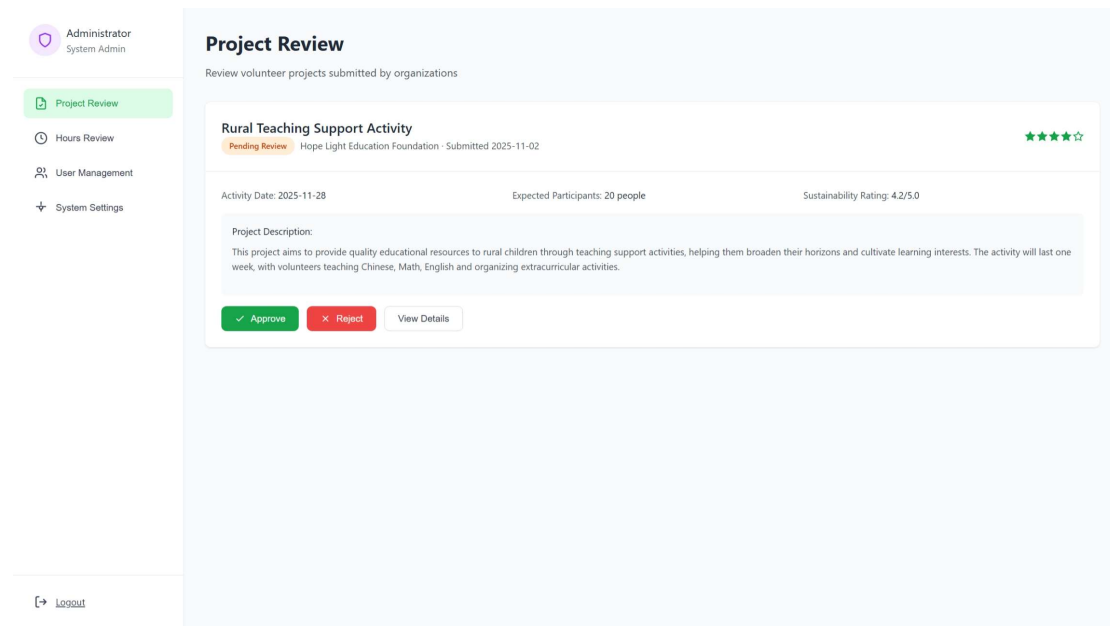
3.4 Organization Dashboard

3.4.1 Organization Dashboard Functional Design

i. Create and publish volunteer projects.

ii. Set project theme, content, time, and minimum participants.

iii. Manage registration lists.

iv. Track project progress and participant engagement.

3.4.2 Organization Dashboard Introduction

Shows organizational project statistics: active projects, participants, completed and pending projects. Allows publishing new projects, managing existing ones, and viewing registration lists.

3.5 Admin Panel

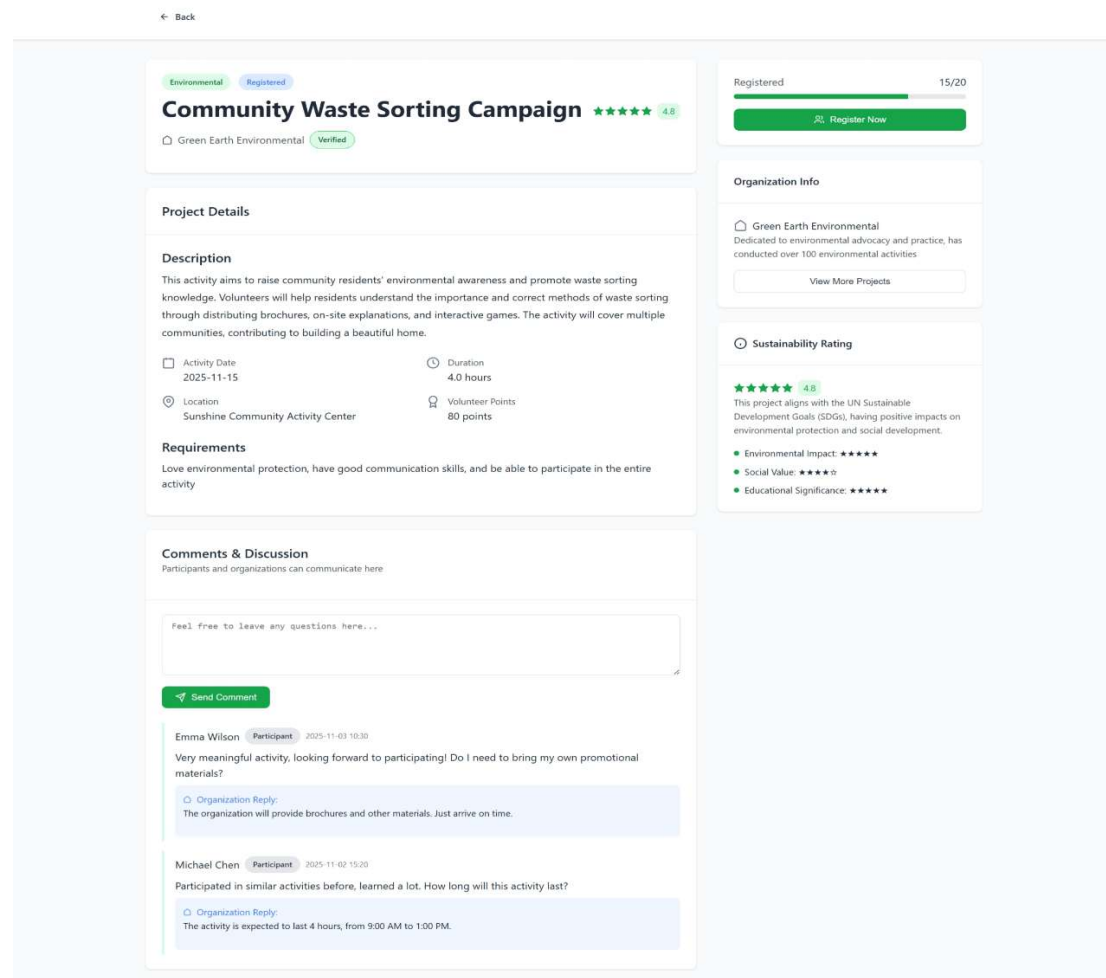## 3.5.1 Admin Panel Functional Design

i. Review and approve organization projects for sustainability alignment

ii. Verify attendance and project completion records

iii. Award service hours and points

iv. Manage all user accounts and database security

3.5.2 Admin Panel Introduction

The Project Review Page lets admins review organization-submitted projects. It displays project title, organization, date, participants, sustainability rating, and description, with options to approve, reject, or view details.

3.6 Project Details Page

3.6.1 Project Details Page Functional Design

i. Organizations can publish projects and students can register to participate.

ii. Supports project management and communication.

iii. Participants and organizations can comment and exchange feedback.

3.6.2 Project Details Page Introduction

The page shows project details (description, time, location, points, participants, and organization info), along with a sustainability rating, registration button, and discussion area for communication.

4. Technical Details

Module	Technology Stack / Tools
Frontend	HTML5 + CSS3 + JavaScript
Backend	Python Flask
Database	SQLite
Security	Password encryption + User access control
Deployment	Local development + GitHub repository
Version Control	GitLab / GitHub
Prototype Design	Figma for UI + Color scheme design

5. Project Structure

Project Structure

```
/
├── app.py                      # App factory, registers blueprints, seeds admin
├── config.py                   # Configuration (secret key, DB URI, etc.)
├── models.py                   # SQLAlchemy models and db instance
├── routes.py                   # All Flask routes (Blueprint 'main')
├── forms.py                    # Minimal form parsers for login/register
├── requirements.txt            # Python dependencies
├── README.md                   # Project documentation
├── instance/                   # Instance folder (Flask convention)
│   └── volunteer.db            # SQLite database (auto-created)
├── templates/                  # HTML templates
│   ├── index.html              # Home page
│   ├── login.html              # Login/Register page
│   ├── participant_dashboard.html
│   ├── organization_dashboard.html
│   ├── admin_panel.html
│   ├── project_detail.html
│   └── volunteer_record.html
└── static/                     # Static files
    ├── css/
    │   └── style.css           # Main stylesheet
    └── js/
        ├── home.js
        ├── auth.js
        └── participant.js
```

This is a typical Flask Web Application structured using the MVC (Model-View-Controller) pattern, including:

i. Backend logic (app.py, routes.py, models.py, forms.py).

ii. Frontend templates (templates/).

iii. Static assets (static/).

iv. Database (instance/volunteer.db).

Core functions include:

i. User registration/login for multiple roles.

ii. Project creation and registration.

iii. Volunteer record management.

iv. Admin dashboard.

v. Clear frontend-backend separation for maintainability.
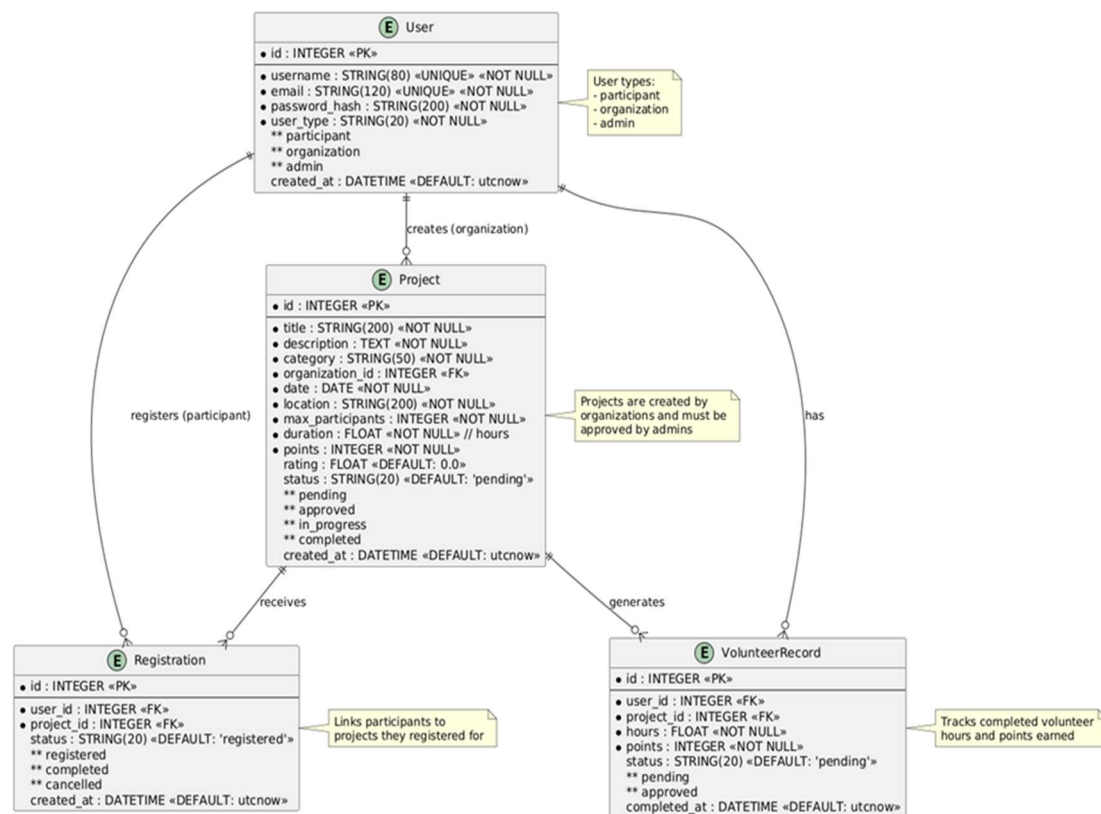
6. Database Models

User: Stores user accounts (participants, organizations, admins)

Project: Volunteer project information

Registration: Project registrations by participants

VolunteerRecord: Certified volunteer hours and points

6.1 Database Schema Diagram

Four main tables: **User**, **Project**, **Registration**, and **VolunteerRecord**.
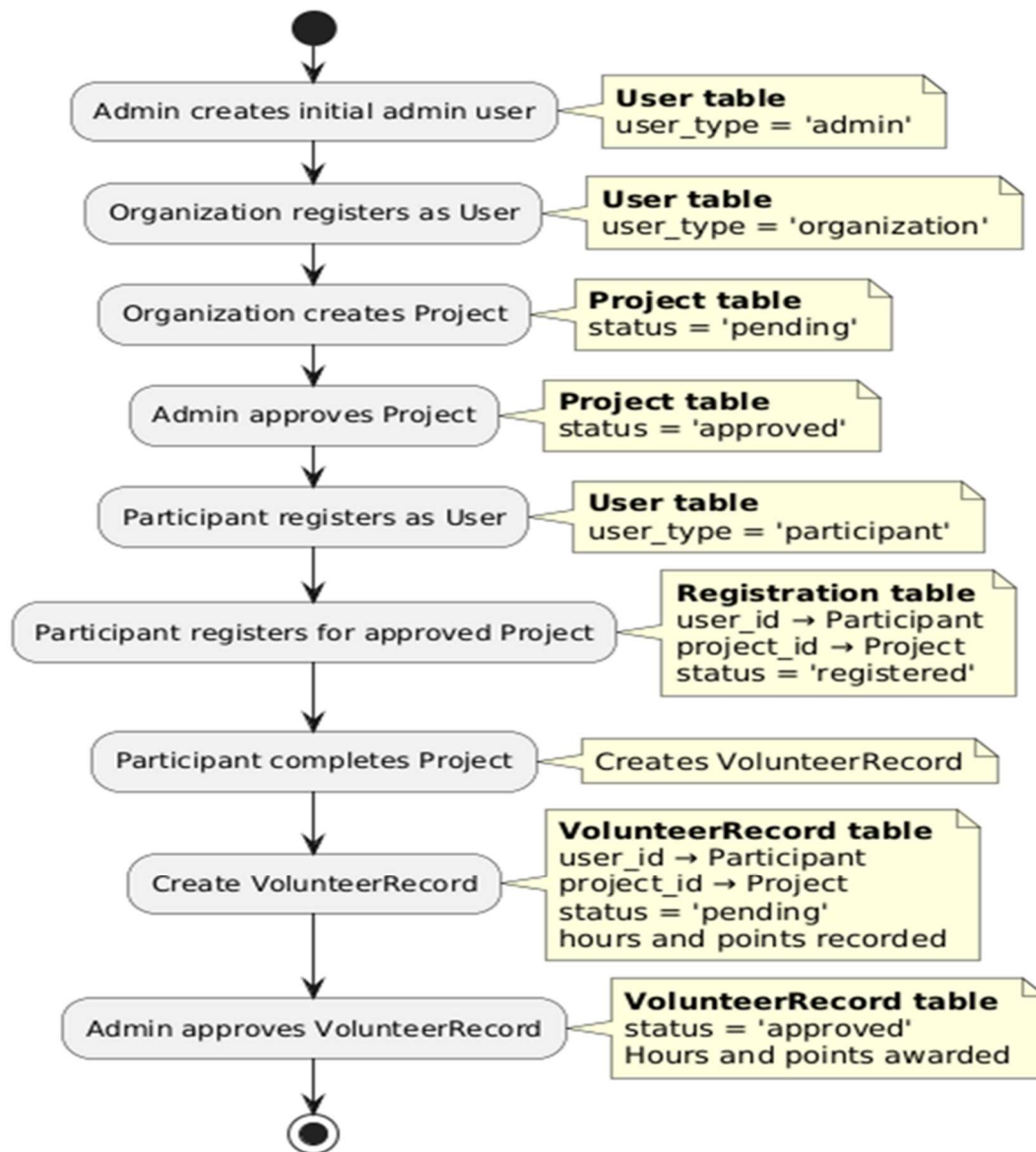
i. User stores all accounts with unique credentials.

ii. Project contains activity details and links to the organization that created it.

iii. Registration connects participants and projects, tracking their status.

iv. VolunteerRecord logs completed work, hours, and points.

Key Relationships:

i. One organization can create many projects (1:N via organization_id).

ii. One participant can register for many projects (1:N via Registration.user_id).

iii. One project can have many registrations (1:N via Registration.project_id).

iv. One participant can have many records (1:N via VolunteerRecord.user_id).

v. One project can generate many records (1:N via VolunteerRecord.project_id).

6.2 Data Population Flow

6.2.1 Activity Diagram:

This activity diagram shows the workflow of how data flows through the system:

i. Initial Setup: Admin creates the first account.

ii. Organization Registration: Organizations sign up.

iii. Project Creation: Organizations create projects (status = pending).

iv. Project Approval: Admins approve projects.
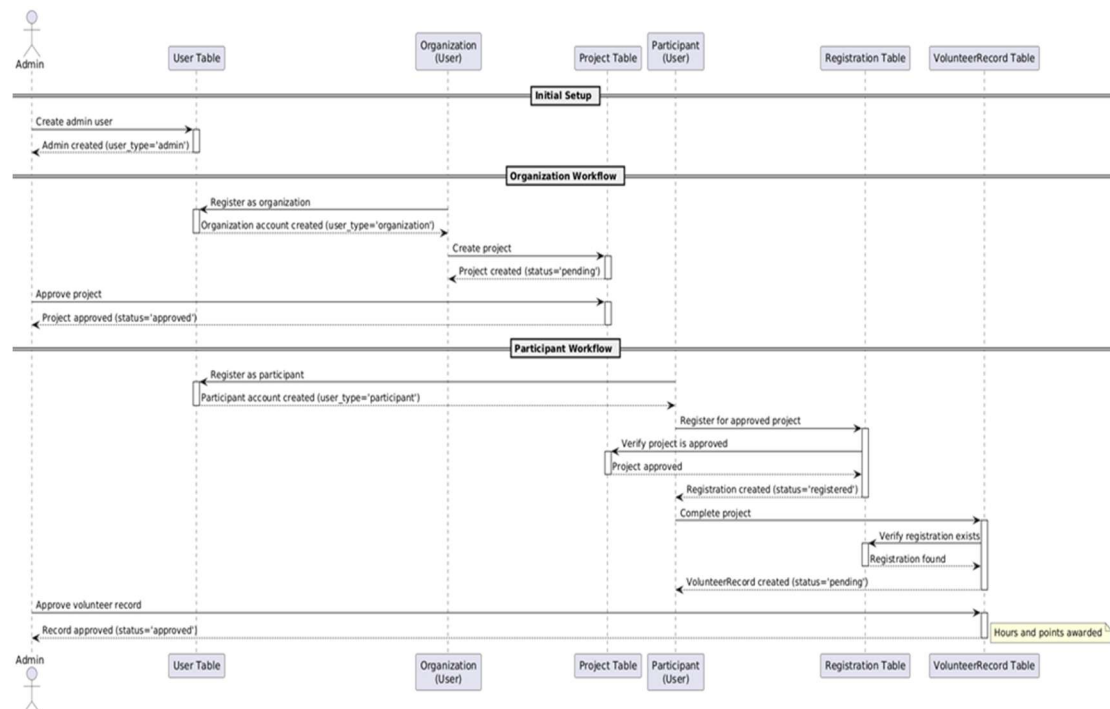
v. Participant Registration: Volunteers sign up.

vi. Project Registration: Participants join approved projects.

vii. Project Completion: Volunteer records are created.

viii. Record Approval: Admins approve records and award hours and points.

The diagram illustrates the complete lifecycle from user registration to volunteer hour certification, showing how different user roles interact with the database at each stage.

6.2.2 Sequence Diagram:

This sequence diagram provides a detailed view of the interactions between system actors and database tables during the data population process:

i. Initial Setup: Admin creates the first user.

ii. Organization Workflow: Organizations register, create projects, and await approval.

iii. Participant Workflow: Illustrates how participants register, search for approved projects, create registration records, complete volunteer work, and finally receive approval for their volunteer hours.

The sequence diagram emphasizes the verification steps (e.g., checking if a project is approved before registration, verifying registration exists before creating volunteer records) and shows the communication flow between different system components. This helps understand the data integrity checks and approval workflows that ensure only valid data enters the system.

7. Conclusion

This project integrates technology and public welfare, building an online platform centered on university volunteer service and sustainable development. It enhances students' social responsibility and offers organizations efficient volunteer management tools - aligning with principles of sustainable development of UCD.