SE1 Team 3: Volunteer Portal

18 December 2025

Bruno CUNHA GASPAR
Gabriel BOTELHO RIBEIRO
Cédric BASSONG
Jiahao LIN
Stylianos NTINOS

Agenda

1. Project Overview
2. Functional Requirements
3. Non-Functional Requirements
4. Use Cases
5. Core Use Cases
6. System Architecture Overview
7. Architecture – Component Structure
8. Development Decisions
9. Development Design
10. MVP Demonstration
11. What Worked, What Didn’t, and What We Learned
12. Team Contributions
13. Conclusion

1.Project Overview

Goal:

- Strengthen community engagement by developing a web platform that aims to connect volunteers with local nonprofit organizations
- Making volunteering more accessible, easy and meaningful

Stakeholders:

- Volunteer (Primary)
- Non-profit Organization (Primary)
- Website Admins (Secondary)
- Local Government (External)
- Developers (Hidden)

2.Requirements - Functional Requirements

Functional Requirements

The system must:

- Allow organizations to create accounts
- Allow organizations to edit their organization profile information
- Allow volunteers to create an account
- Allow volunteers to edit their volunteer profile information
- Allow organizations to create, edit, and delete volunteering opportunities
- Allow volunteers to browse and filter available volunteering opportunities
- Allow volunteers to apply to volunteering opportunities
- Notify organizations when a volunteer applies to an opportunity
- Allow organizations to accept or reject volunteer applications
- Notify volunteers of changes in their application status.
- Allow volunteers to report inappropriate or suspicious opportunities
- Allow administrators to review and verify nonprofit organization accounts

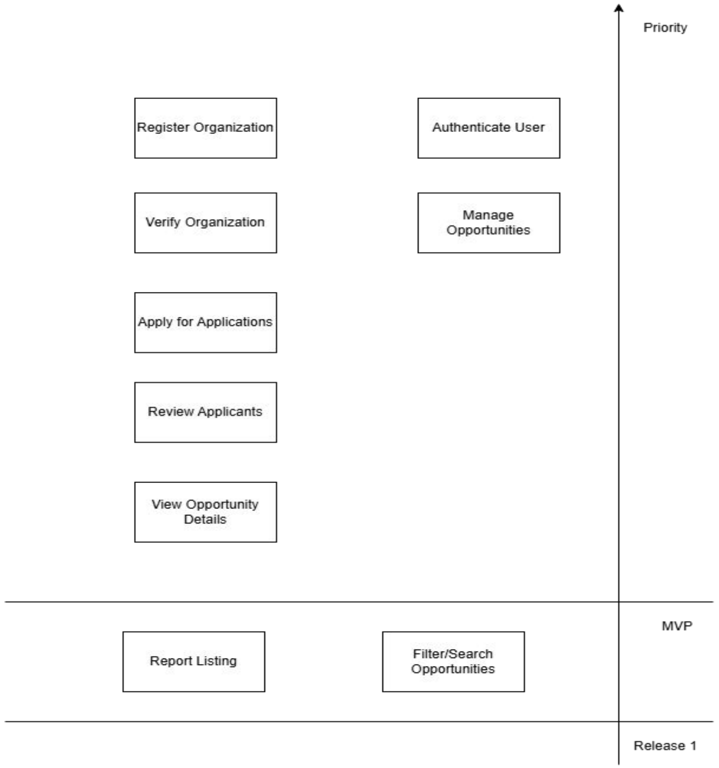
3.Requirements - Non-Functional Requirements

- The system must support at least 100 concurrent authenticated users while maintaining functional availability
- The system must ensure that the Largest Contentful Paint (LCP) occurs within 2.5 seconds for at least 75% of user sessions under normal load
- All client–server communications must be encrypted using HTTPS with TLS 1.2 or higher
- The system must be responsive and usable on desktop and mobile devices, and comply with WCAG 2.1 AA accessibility guidelines
- Access to system functionalities must be restricted based on user roles using server-side authorization checks
- The backend must be structured into independent modules for authentication, user management, opportunity management, and application management, with no direct database access outside each module's data layer
- User-triggered navigation or state updates must provide visible feedback within 200 ms in at least 90% of interactions

4.Use Cases

- UC-01: Moderate Listings
- UC-02: Apply for Opportunity
- UC-03: Filter/Search Opportunities
- UC-04: Manage Opportunities
- UC-05: View Opportunity Details
- UC-06: Verify Organization
- UC-07: Authenticate User
- UC-08: Explore Opportunities
- UC-09: Register Organization
- UC-10: Send Notification
- UC-11: Report Listing
- UC-12: Manage User Access
- UC-13: Review Applicants
- UC-14: Manage Applications

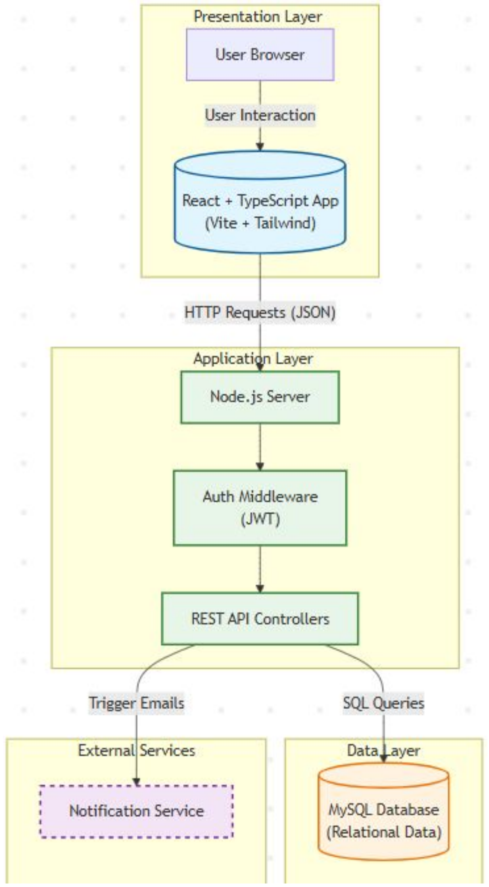
5.Core Use Cases for MVP



6.System Architecture Overview

Architectural Style: Client-Server Architecture (Three-Tier).

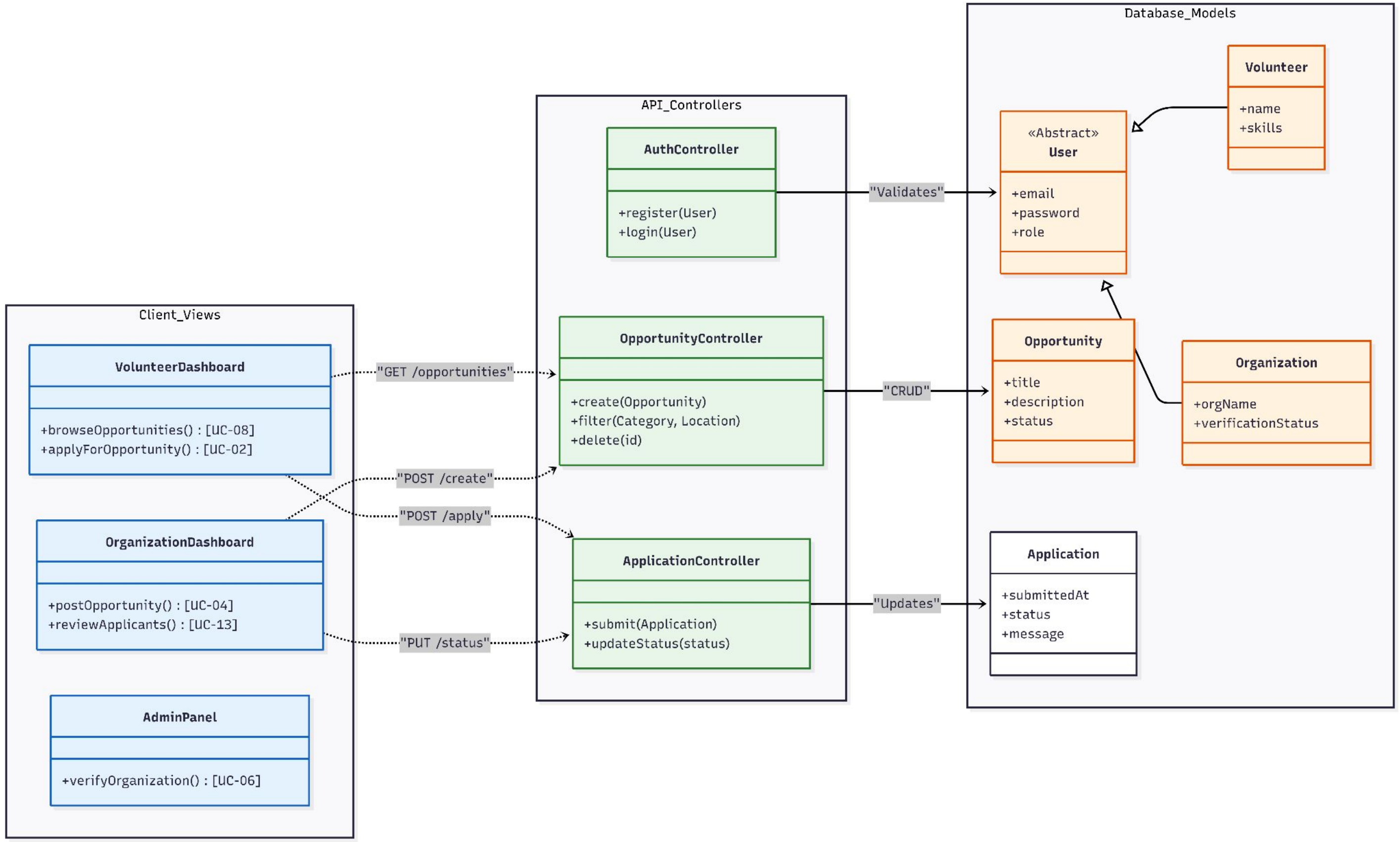
- **Frontend (Presentation Layer):**
    - **Framework:** React + TypeScript (ensures type safety and component reusability).
    - **Responsibility:** Handles user interactions (Volunteers browsing, Organizations posting) and communicates with the backend via REST API.
- **Backend (Application Layer):**
    - **Runtime:** Node.js.
    - **Responsibility:** Processes business logic (e.g., verifying organizations, handling applications) and manages authentication.
- **Database (Data Layer):**
    - **System:** MySQL.
    - **Responsibility:** Stores persistent data including User profiles, Opportunities, and Application statuses.
- **Data Flow:**
    - Frontend sends **HTTP Requests** (GET/POST) →Backend processes logic → Database executes **SQL Queries**



7.Architecture - Component Structure

- **Backend Structure (MVC Pattern):**
  - **Models (Data Entities):** Directly mapped from the Domain Model:
    - `User` (Abstract parent for Volunteer/Organization)
    - `Opportunity` (Attributes: title, location, status)
    - `Application` (Association between Volunteer and Opportunity)
  - **Controllers (Logic):**
    - *AuthController:* Handles login/registration and role-based access.
    - *OpportunityController:* Manages posting, editing, and deleting opportunities (FR2).
    - *AdminController:* Logic for verifying nonprofit accounts (FR6).
- **Frontend Structure (React Components):**
  - **Pages (Routes):**
    - `Home/LandingPage`
    - `BrowseOpportunities` (Implements filtering/searching - FR3).
    - `Dashboard` (Organization view for managing posts).
  - **Components (Reusable UI):**
    - `OpportunityCard` (Displays individual listing details).
    - `ApplicationForm` (Modal or section for applying).
    - `NavBar` (Context-aware based on user role).

7.Architecture - Component Structure



8.Development - Decisions

Development Approach

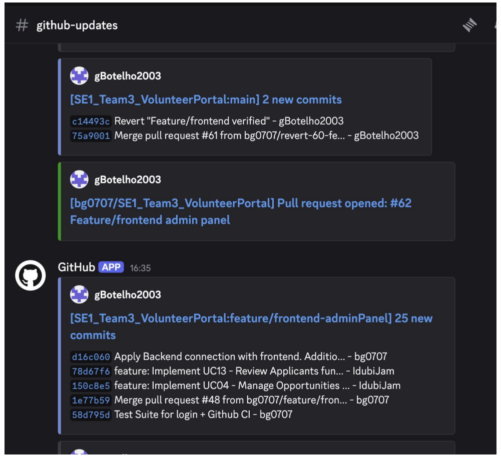
- **Agile development using Kanban**
- Iterative implementation (Sprints) focused on MVP delivery
- Tasks managed with **GitHub Projects**

Testing & CI

- Tests executed on **each commit and pull request**
- GitHub CI pipeline ensures code stability

Architecture Priority

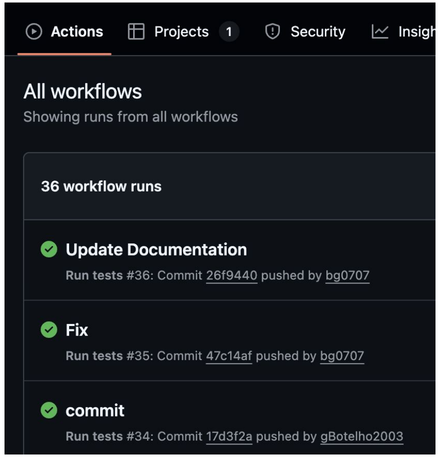
- Client–server architecture
- Functional MVP over visual polish


github-updates

gBotelho2003
[SE1_Team3_VolunteerPortal:main] 2 new commits
c14493c Revert "Feature/frontend verified" - gBotelho2003
75a9001 Merge pull request #61 from bg0707/revert-60-fe... - gBotelho2003

gBotelho2003
[bg0707/SE1_Team3_VolunteerPortal] Pull request opened: #62 Feature/frontend admin panel

GitHub APP 16:35
gBotelho2003
[SE1_Team3_VolunteerPortal:feature/frontend-adminPanel] 25 new commits
d16c060 Apply Backend connection with frontend. Additio... - bg0707
78d67f6 feature: Implement UC13 - Review Applicants fun... - IdubiJam
150c8e5 feature: Implement UC04 - Manage Opportunities ... - IdubiJam
1e77b59 Merge pull request #48 from bg0707/feature/fron... - bg0707
58d795d Test Suite for login + Github CI - bg0707



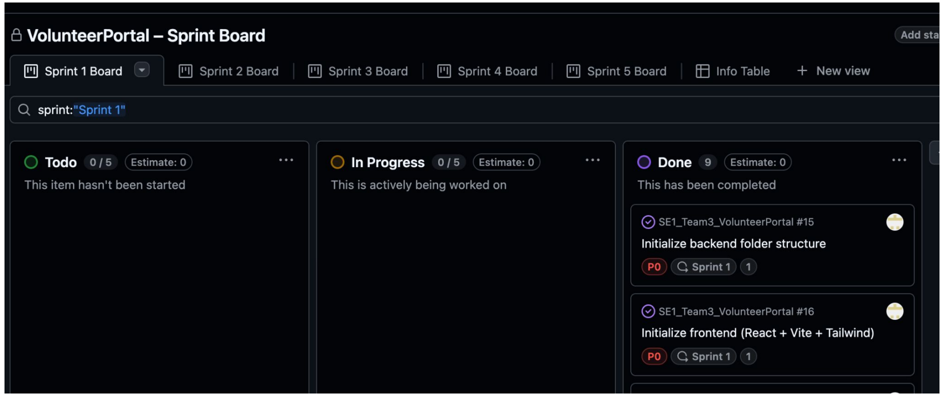
All workflows
Showing runs from all workflows

36 workflow runs

Update Documentation
Run tests #36: Commit 26f9440 pushed by bg0707

Fix
Run tests #35: Commit 47c14af pushed by bg0707

commit
Run tests #34: Commit 17d3f2a pushed by gBotelho2003


VolunteerPortal – Sprint Board

sprint:"Sprint 1"

Todo 0 / 5 Estimate: 0
This item hasn't been started

In Progress 0 / 5 Estimate: 0
This is actively being worked on

Done 9 Estimate: 0
This has been completed

SE1_Team3_VolunteerPortal #15
Initialize backend folder structure
P0 Sprint 1 1

SE1_Team3_VolunteerPortal #16
Initialize frontend (React + Vite + Tailwind)
P0 Sprint 1 1

9.Development - Design

- **Use Case Descriptions**
  Functional requirements expressed as detailed use case descriptions
- **What the User Sees → Frontend**
  UI components and pages implementing the use case
- **What the System Processes → Backend**
  Controllers and services handling business logic
- **What Data We Need → Database**
  Entities, relationships, and queries supporting the use case
- **Implementation**
  Use case flows refined with sequence diagrams and translated into code

11.What worked, what didn’t and what was learnt

What worked

- **Kanban board** for task tracking and visibility
- **Sprint organization** to structure the workflow
- **Weekly meetings + Discord–GitHub sync** to keep everyone aligned

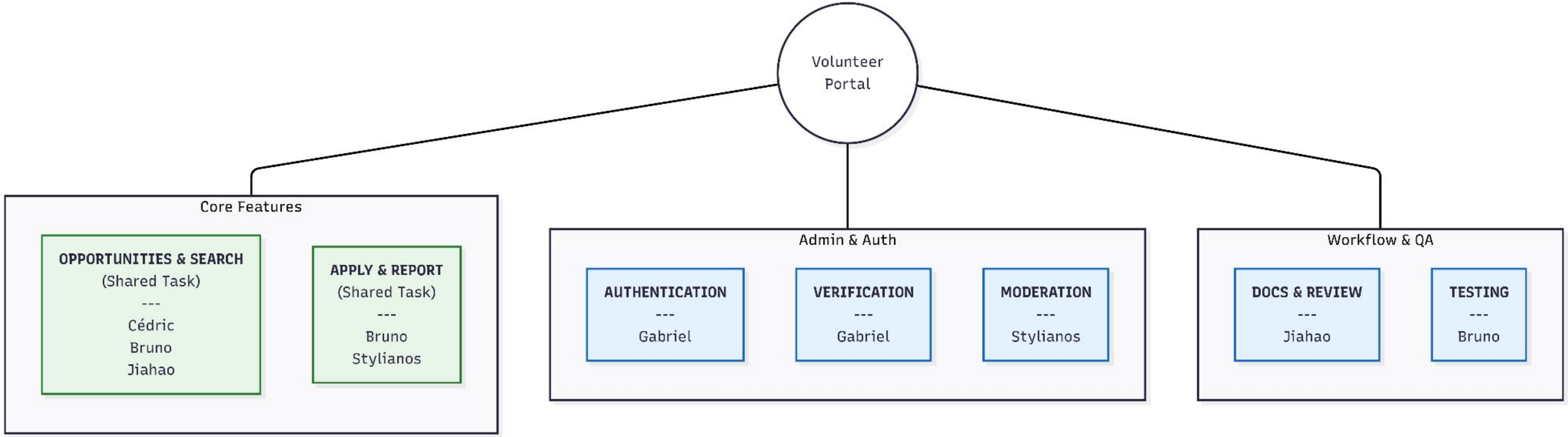
What didn’t work

- **Communication** issues
- Work was sometimes **duplicated**
- **Multiple people** on the **same task** without coordination

What we learned

- Communicate earlier and more clearly within the team
- Define clear roles for larger or shared tasks
- Better-structured meetings improve coordination

12.Contributions



13.Conclusion

- Delivered a functional **MVP Volunteer Portal**
- Met core requirements using a **three-tier architecture**
- Enabled volunteers, organizations, and admins to interact effectively
- Agile development supported iterative progress
- Project provides a solid base for future improvements

10.MVP Demo

Different Points of view (POV)

- Volunteer
- Organization
- Admin

Display of the main functionalities

- Authentication
- Application
- Verification

End of Presentation

Thank you for listening!