University of Bahrain

College of Information Technology

Department of Information Systems

B.Sc. In Cyber security

Design and Development of a Secure Civic Engagement Platform for Community Problem Reporting and Volunteer Management

Prepared by

Abdulla Ali Abdulla Khamis	202102256
Hussain Habib Sahwan	202102620
Salman Mohammed Altal	202102525

For

ITCY 499 & ITIS 499

Senior Project

Academic Year 2025-2026-Semester 1

Project Supervisor:
Dr. Alya Sayed Husain AlKameli

Dr. Ghassan Mohammed Al Koureiti

1st December 2025

Abstract

This project presents the design and development of "We Are The Change," a civic engagement platform that enables citizens to report local issues and participate in community volunteering through a mobile application. Addressing the problem of inefficient public issue reporting and limited citizen engagement, the system allows users to submit reports with photos and sign up for volunteer activities managed by administrators. The project involved system analysis, security requirements engineering, application of threat modeling methodologies, and the implementation of secure user authentication and data handling with Firebase. Testing demonstrated reliable performance, robust data protection, and smooth user experience across devices. Results show the platform effectively streamlines issue reporting, supports secure data management, and encourages greater community participation, confirming the solution's practical value and scalability.

Keywords: civic engagement, volunteer platform, issue reporting, threat modeling, Firebase, SQUARE methodology, STRIDE, Bahrain civic tech, mobile application, community participation.

Acknowledgments

We would like to express our sincere gratitude to Dr. Alya Sayed Husain AlKameli and Dr. Ghassan Mohammed Al Koureiti for their invaluable supervision, encouragement, and continuous support throughout all stages of this project. Their expert guidance, insightful feedback, and patience were essential in helping us overcome challenges and achieve our goals.
We are also deeply appreciative of Mr. Mohamed Khamis for taking the time to participate in an interview and for sharing his experience and perspectives, which contributed meaningfully to our research and understanding.
Our heartfelt thanks go to our families for their unwavering encouragement and support, and to all colleagues and friends who offered resources, advice, and motivation. Each person's contribution played an important part in this journey.
Thanks to everyone who believed in us and stood by us throughout this project.

Table of Contents

List of Tables

List of Figures

Chapter 1
Introduction

1.1 Introduction

Civic platforms involve the communities with the local government where they report and have all small issues such as broken roads, dirty beaches, or maintenance fixed. In Bahrain, digital government programs such as Tawasul have enhanced a better way of submitting complaints, yet they predominately depend on governmental responses without considering the citizens in solutions. In this project, it is possible to create a secure mobile application, We Are The Change, which was developed by FlutterFlow frontend and Firebase backend and allows users to report issues with photos, become a task volunteer, seek sponsor support, and allows the administrator to manage activities via a dashboard (eParticipation Tools, n.d.).

The digital transformation in Bahrain facilitates these kinds of innovations, such innovations are focused on ensuring transparency and access to public data to enhance the levels of participation. Nonetheless, research indicates that community functions are not exploited when it comes to long-term engagement. The RCSI Bahrain stakeholder interviews revealed actual breakages in practice, including verbal reports with no pictures that lead to delays and misunderstandings (Digital Government Strategy Principles, n.d.).

This senior project thesis statement is as follows: This senior project is a proposal that designs, implements, and thoroughly tests a secure, civic engagement platform that combines issue reporting and volunteer coordination to fulfill the requirements of the Bahrain community of community-based problem-solving and to utilize academic methodologies of illuminated security practices such as SQUARE and STRIDE.

1.2 Problem Statement

The current systems in Bahrain result in delays and passive roles of the community, which puts the country in difficulties with efficient reporting of local issues and citizen participation. The primary issue can be divided into three sub-issues: (1) disjointed, unstructured reporting through emails, WhatsApp, or Tawasul without organized volunteering association; (2) the absence of coordination in reported needs, volunteers, and sponsors, and as a result, low-priority tasks could be unaddressed; (3) security on user information, media upload on volunteer platforms, and API access on volunteer platforms.

These problems have been evidenced: The RCSI Bahrain interview showed that there were inconsistent details in oral reporting, delays in vendors, and lack of transparency in the status updates. Tawasul, being a good tool in government complaints, also does not allow any action on the part of the citizens except to make complaints, which get slow responses on non-urgent fixes. Informal campaigns such as the Bu Jarrah in Kuwait do not need tracking or scalability since they are successful through pressure by the social media. At the national scale, eParticipation tools in Bahrain have a potential yet without the integration of volunteers, the engagement decreases as evidenced by the lack of follow-ups on municipal reports. Such loopholes lower trust and slacken community development in places such as Manama (eParticipation Tools, n.d.).

1.3 Project Objectives

The main objective will be the development of a functional and safe mobile platform where Bahraini citizens will report on community issues and engage in solutions. Particular tasks are: (1) derive requirements by interviewing stakeholders and the SQUARE technique; (2) threat model with STRIDE to discover and work around the potential risks; (3) schema design ERDs, user interface prototypes, and workflows; (4) implement features through FlutterFlow and Firebase with authentication and media support; (5) conduct functional, and security tests; (6) deploy prototype and documentation to be used later.

The resultant system provides a cross-platform application with user roles (citizens/volunteers, sponsors, admins), case reporting and images, volunteer requests, sponsor requests and approval capabilities of the administrator. The target population is residents of Bahrain, community managers, and local non-governmental organizations with a local impact in cities such as universities and neighborhoods. As assumptions, user access to smartphones and the internet has been made; constraints exist in Firebase free tier (capping scale testing), only manual testing, and no integration with government APIs such as Tawasul.

1.4 Relevance/Significance of the project

- Launches the first reporting-volunteering platform in Bahrain, closing gaps between Tawasul (only government) and social media initiatives (Digital Government Strategy Principles, n.d.).
- Displays student-controlled security engineering using SQUARE/STRIDE, prevents attacks such as spoofing, attacks, and information disclosure in real civic tech.
- Enables communities to do their own low-priority work (e.g. cleanups), cutting government work and creating ownership per civic research (AL-Kaabi, 2024).
- Prototypes no-code (FlutterFlow/Firebase) to support quick prototyping, which NGOs/students can do with the cloud-first approach of Bahrain.
- Contributes academically: comprehensive threat tables, testing results, and Agile management as senior project model.

1.5 Report Outline

Chapter 2 is a review of local (Tawasul, Balady) and international (SeeClickFix) civic platforms, which does not show gaps in volunteer integration. Chapter 3 elaborates on Agile process, risk management and Gantt timeline. Chapter 4 includes requirement elicitation based on RCSI interviews, SQUARE security measures, and the system models. ERD, database schemas, user interface designs, and pseudocode are shown in chapter 5. Chapter 6 explains how it is implemented, what the feature is, and testing results (100 per cent functional, threat mitigation). Chapter 7 will summarize the accomplishments, constraints such as funding availability and improvements in the future.

Chapter 2
Literature Review

2.1 Introduction

With the growing demands for public utilities, infrastructure, and sustainability in towns and cities, public engagement has become a crucial component of modern city planning. New channels of communication between the public and the government have been made possible by the integration of online platforms into public involvement mechanisms. The most of current systems, however, rely on problem reporting and offer little or no opportunity for communities to actively participate in identifying solutions. By providing a web-based and/or mobile application that allows Bahraini citizens to report problems like unclean beaches or broken roads and, critically, allows users to volunteer to help fix them, the 'We Are The Change' program aims to close this gap. This assessment of the literature examines similar systems, lists their benefits and drawbacks, and determines the space for innovation that this project seeks to fill.

2.2 Existing Local Models

2.2.1 Tawasul – Government-Centric Reporting

One of the official applications for civic engagement in Bahrain is the Tawasul application, which was introduced by the National Suggestion and Complaint System. It enables locals and citizens to submit complaints or recommendations regarding governmental services. These cases are forwarded by the system to the appropriate government entities, who handle their investigation and resolution (Authority, n.d.).

Even though Tawasul has improved accountability and openness in public services, it is still a top-down procedure where citizens' responsibilities end when they file their complaint. Other than raising awareness with the authorities, there is no way for citizens or civil society to provide input and contribute to the solution. Additionally, the system's reaction times are longer, particularly for low-priority situations with limited government funding, including beach cleaning or minor road maintenance for example.

2.2.2 Balady – Municipality Reporting

Balady is an electronic platform established by the Ministry of Municipalities and Housing in Saudi Arabia, along with municipalities, in an attempt to assist the municipal community based on interactive, electronic, and information services. It allows citizens and residents to report about municipal problems (e.g., a broken road or a pothole) to the app and request a reference number to follow the process by entering the actual location on a map after which they can provide a description, attach pictures, and select a category and a subcategory (e.g., roads and sidewalks). The aim of the platform is to support community partnership, license requests, and decision-making to improve the quality of service, sustainability of urban environments, and satisfaction of residents in the Kingdom (Balady, submitting a report, n.d.) (Balady, balady about, n.d.).

Despite being simplified with features such as the ability to allow guests access to the municipal reporting through a mobile number and Snap and Send database to upload photos with complaints, Balady is a centralized platform where the essential role of the citizens is largely limited to submitting their complaints. Public follow-up input mechanisms and joint resolution mechanisms are limited, and response times may be limited in relation to lower priority maintenance challenges based on municipal resources and funds (Balady, submitting a report, n.d.).

2.2.3 Bu Jarrah – Social Media Advocacy Without Infrastructure

One of the most notable grassroots instances comes from Kuwait, where TikTok influencer Bu Jarrah utilizes his platform to raise awareness of problems that communities face. His videos are meant to challenge the government or other individuals to act by documenting broken streets, old and damaged public spaces. Even in the absence of institutional tools or platforms, his success shows the strength of social influence and public pressure to push for change (AlOthman, n.d.).

However, Bu Jarrah's model is unstructured and informal. It has neither a mechanism for recording issues nor one for measuring results or organizing volunteers. Additionally, because it depends on the influencer's unique brand, it cannot be sustainably duplicated or expanded. Additionally, it faces the potential of unreliability, as the visibility of a case is based on whether or not the influencer notices it.

2.3 International Systems and Civic Technology

People can now report non-emergency issues on a variety of civic tech websites that have emerged globally. One that's clearly identifiable is SeeClickFix, an American website that allows users to report problems to the appropriate authorities, such potholes, graffiti, or faulty lighting. Users are able to see the status of their reports and even provide feedback (SeeClickFix, 2022) due to the two-way technology.

FixMyStreet, created by MySociety in the UK, is an illustration of this. This enables people to notify the appropriate local government regarding problems with public infrastructure (MySociety, n.d.) (Fung, 2006). Both platforms have contributed to increased levels of trust in the local government by providing openness and usability.

However, similar to Tawasul, these websites typically offer little in the way of community-based action. The government must be in charge of solving all problems. Because there is no chance for volunteers on these websites, communities are unable to organize themselves to meet their own needs.

2.4. Community Participation in Public Service

Community-driven approaches to problem-solving are becoming more popular as an alternative to government initiatives. According to civic tech studies, residents feel more ownership and community pride when they are involved in both problem identification and solution (Fung, 2006). Neighborhood associations, volunteer repair programs, and community cleanup efforts have played a big role for many years, but they are usually offline, disorganized, and difficult to scale.

Putting these models online presents an appealing alternative. Civic engagement would be greatly increased with a platform that enables users to plan events, assign teams, exchange updates, and monitor results in addition to reporting.

Platform	**Main Features**	**Volunteer Integration**	**Community Collaboration**	**Reporting Mechanism**	**Limitations**
Tawasul	Government complaints system	None	No	Formal, to government only	No volunteer action, government only solutions.
Balady	Government complaints system	No	No	Formal, to government only	No volunteer action, government only solutions.
Bu Jarrah (Social Media)	Awareness via influencer	Yes	Weak	Social, informal	Not sustainable, not secure.
SeeClickFix / FixMyStreet	Web/app issue reporting	None	Weak	Structured, mapped	Government only solutions.
We Are The Change	Issue reporting + volunteering	Full (built-in)	Strong	Structured, mapped	Still developing.

Table 1: Comparison of Civic Engagement and Reporting Platforms

2.5. Gaps and Innovation Opportunity

According to Table 1: Comparison of Civic Engagement and Reporting Platforms, the current scenario of civic reporting mechanisms reflects one ongoing void: a lack of integration of volunteers. Current solutions either use informal social media outreach (such as Bu Jarrah's TikToks) or government response (such as Tawasul, SeeClickFix, and FixMyStreet). No platform with official reporting and a formally recognized volunteer-based solution approach currently exists in the area.

'We Are The Change' fills the gap by putting out a model that is in the middle: a place where people may report a problem locally and offer to actively participate in resolving it themselves, whether it be by filling potholes on a road, cleaning a beach, or planting a tree (with the proper permissions). Additionally, such a site might display community work, allow residents, non-governmental organizations, and even the government to collaborate, and allow for the tracking of issues' progress.

2.6. Conclusion

Overall, even though there are several organizations and reporting systems both locally and internationally, very few of them have offered a way for regular people to contribute to the solution. "We Are The Change" has proposed a fundamental change away from the community's passive complaint and toward significant change. This project proposes a civic innovation that not only detects problems but also includes citizens in solving them. It does this by combining aspects from official reporting sites like Tawasul and social influence from people like Bu Jarrah.

Chapter 3
Project Management

The chapter explains the most important management practices, development processes, and planning strategies that were adopted to make sure that the application under consideration, i.e. the application "We Are the Change" was launched successfully. It includes the selected software development process model, methods for identifying and mitigating risks, and a thorough activity plan that references the project at each step of development and completion. A suitable project management was necessary because: evolving nature of the requirements, user feedback consulted, and the technical aspect of combining both reporting and volunteering capabilities in one platform to come up with a single platform has some challenges.

3.1 Process Model

In coming up with the development of the project, We Are The Change, Agile Software Development Life Cycle (SDLC) will be used as the main model of the process. Agile was selected due to its flexibility and adaptability which helps to provide constant improvement and feedback to its users during the development phase. Unlike more traditional model like the Waterfall, Agile does not mandate a rigid linear progression, with its construction allowing the incremental build and constant engagement with the users which was also key to success in this project.

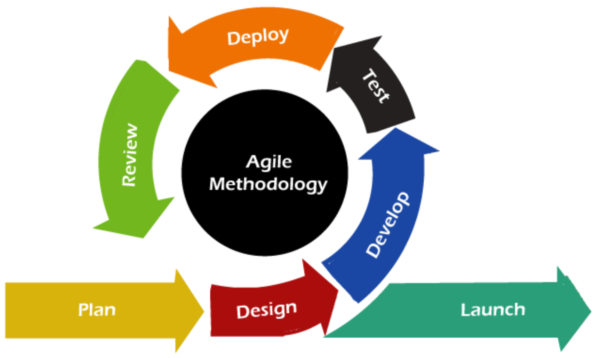


Figure 1: Agile Model

According to Figure 1: Agile Model, the Agile Software Development Life Cycle follows a structured yet flexible approach to building software through iterative phases. It begins with concept and requirement gathering, where stakeholders and product owners work together to define high-level requirements and capture them as user stories in a product backlog. During sprint planning, the development team selects user stories from the backlog to work on in the upcoming sprint, estimates the effort required, and assigns tasks to team members. The design phase employs lightweight and flexible design strategies, focusing on creating just enough structure to begin development while maintaining agility and adaptability. In the development phase, programmers write code based on the selected user stories, often following practices such as test-driven development (TDD) or pair programming to ensure code quality and collaboration. Testing occurs continuously as each feature is developed, with continuous integration enabling frequent testing, rapid feedback, and early bug detection. At the end of each sprint, a deployment phase delivers a potentially shippable product increment, which may be released internally or to customers depending on readiness. Finally, the sprint concludes with a review and retrospective session, where the team demonstrates completed work, discusses what went well, identifies areas for improvement, and adapts their approach for the next sprint to ensure continuous improvement throughout the development process (Liyanage, 2025).

Justification for Selecting Agile

Agile was chosen for this project because it is well suited to handling changing requirements and evolving priorities. During development, the team received new ideas and feedback about the user interface and the importance of specific features, such as reporting, volunteering, and admin tools. Instead of freezing the scope, Agile allowed these changes to be added and re-prioritized in the backlog without disrupting the overall progress. The modular nature of the system, with separate components for authentication, issue reporting, volunteer dashboards, and admin control, also fitted naturally with Agile's incremental delivery. Each module could be planned, developed, and refined in focused iterations. Regular meetings, such as sprint reviews and planning sessions, helped the team stay aligned, communicate clearly, and quickly respond to problems. Finally, Agile's emphasis on delivering working software in short cycles supported early testing and bug fixing, improving the quality and stability of the application before final deployment.

Application of Agile in the Project

In this project, Agile was applied by organizing the work into weekly sprints with clear goals for each iteration. At the start of every sprint, the team held a planning session to select user stories from the backlog, break them into tasks, and assign responsibilities. Progress was tracked using a digital board, where tasks related to UI/UX design, Firebase integration, and core features such as registration, reporting, and volunteer management were moved through stages like "To Do," "In Progress," and "Done." Features such as volunteer registration, report submission, and admin verification were implemented in separate, focused sprints, which kept the codebase modular and easier to maintain. Short stand-up meetings were used to discuss blockers, share progress, and reassign tasks when needed, making the workflow more responsive and proactive. At the end of each sprint, the team reviewed the completed work, gathered feedback, and used a retrospective to decide how to improve their process in the next cycle.

3.2 Risk Management

Table 2: Risk management below indicates the most important risks that were identified in the project and their possible impact, and mitigation approaches taken.

Risk	Level	Risk Mitigation Plan
A developer leaves the team	High	Ensure all members are familiar with each other's work and use Flutter Flow for version control.
Difficulty integrating Firebase backend	Medium	Conducted early research and prototyping. Used official documentation and tutorials.
Time mismanagement	High	Weekly sprints and deadlines were monitored using Trello.
Limited Flutter Flow experience	Medium	Used YouTube and official Flutter Flow documents to accelerate learning curve.
Feature creep (adding too many features)	Medium	Clear scope definition and weekly reassessment of tasks.
App not working on some devices	Low	Used emulator and real device testing throughout the process.

Table 2: Risk management Table

3.3 Project activities Plan

Phase	Start Date	End Date	Duration
Requirement Analysis	1 July 2025	20 July 2025	3 weeks
System Design	21 July 2025	10 August 2025	3 weeks
Implementation	11 August 2025	10 October 2025	2 months
Testing (Functional + Security)	11 October 2025	10 November 2025	1 month
Deployment	11 November 2025	18 November 2025	1 week
Documentation	19 November 2025	26 November 2025	1 week
Final Review	27 November 2025	29 November 2025	3 days

Table 3: Project Gantt chart

3.3.1 Requirement Analysis (1–20 July 2025)

In this phase, the main goal is to understand what the system should do. This includes collecting user needs, reviewing existing solutions, and defining the project scope. Clear requirements help ensure the system will meet user expectations.

3.3.2 System Design (21 July – 10 August 2025)

System design focuses on planning how the system will work internally. This includes designing the architecture, database, interfaces, and workflows. Having a proper design helps reduce errors during implementation.

3.3.3 Implementation (11 August – 10 October 2025)

The development stage is aimed at creating the system with the help of visual tools of FlutterFlow rather than manual code. It involves the development of screens, workflows, database connectors, and user interface components, which are based on the previous design. It is also the most lengthy since majority of features and logic of the system are compiled, set up and tested in the FlutterFlow environment.

3.3.4 Testing (11 October – 10 November 2025)

Testing requires all the functionality and integration in the application to be checked to be sure that the system is functional and secure. This would involve functional testing to ensure that all parts of the system are acting as intended and security testing to ensure that the system is resistant to standard threats like SQL injection and XSS and also to check the application logic to determine any possible vulnerability. The general purpose of this step is to spot and correct mistakes before the system has been implemented so that there is a smooth and safe operation.

3.3.5 Deployment (11–18 November 2025)

The system is prepared for use. This may include setting it up on a server, configuring the database, and making sure everything runs properly in the final environment.

3.3.6 Documentation (19–26 November 2025)

The current stage is dedicated to drafting the thesis chapters in regards to the project. This involves the description of the system, record keeping of the testing results, appraisal of the system, and the written report.

3.3.7 Final Review (27–29 November 2025)

The final phase includes reviewing the whole project, fixing small points if needed, and preparing the presentation slides for the final evaluation or defense.

Chapter 4
Requirement Collection and Analysis

4.1 Requirement Elicitation

In order to gain a deeper insight into the needs and expectations to the topic of the current research, which is the civic engagement platform “We Are The Change”, the structured interview of the Mr. Mohamed Abdulla Khamis, who is the Head of Estates and Support Services at RCSI Bahrain, was realized. The aim of the interviewing was to understand the current state of issue reporting processes, to reveal current problems and issues, and to determine the degree of readiness to involvement of the community, and to obtain opinion of experts on the concept of the proposed application.

4.1.1 Current Processes and Challenges

Mr. Khamis explained the existence of a formal process of maintenance requests in the RCSI Bahrain at present. Any reported problem is reported to the Estates Helpdesk and listed in a Computerized Maintenance Management System (CMMS). These are then prioritized and the workplaces allocated to the in-house teams or to the contractors.

Nevertheless, there are a few issues that were brought out:

- Lack of consistency in reporting and in particular not providing any essential details like location or photos.
- Miscommunication is likely to take place because of verbal reports. Prioritization is hard when reporting high volumes.
- Transparency is not there, and requestors are forced to run the same manually to get their update.
- Making use of the dependencies on the external vendors resulting in delays.
- Such delays caused by external contractors.

Such constraints offer the importance of having a more organized, virtual, and user-friendly solution.

4.1.2 Stakeholder Roles and Reporting Channels

Students and faculty may sometimes be involved in making most of the reports as well as those being reported by administrative staff, security and cleaning teams. There are several reporting channels that have been applied:

- Email (most common)
- Phone calls (urgent cases)
- In-person notifications
- Informal tools such as WhatsApp or MS Teams

The interviewee pointed out that although these options offer flexibility, they usually lead to a highly divided communication and monitoring.

4.1.3 Perceptions on Community Participation

Mr. Khamis cited isolated though good manifestations of student participation in the cleaning and greening up of campuses. He was in favor of enhanced community involvement in a more constructive basis, which should be guaranteed safety and supervision. The action of planting trees, recycling campaigns and beautification programs were regarded to be useful to campus operations as well as civic growth of students.

4.1.4 Feedback on the Proposed Application

The idea of a unified system of reporting of public problems and offering to volunteer at their solution was met very positively. It was suggested that:

- **Image attachments**
- **Real-time tracking** of issue status
- **Volunteer sign-up functionality** with safety guidelines
- **Admin dashboard** for assignment and monitoring

The platform was envisioned as the possible way to mitigate transparency, decrease the presence of the bottlenecks, and foster the civic engagement within students and staff.

4.1.5 Operational and Technical Considerations

A number of highlighted operational and safety concerns were presented:

- Technical and high safety activities (e.g., electrical, HVAC) should not be directed to volunteers.
- It is necessary to have clear role definition, briefing on safety and supervision.
- The system is expected to be integrated with the currently used tools like CMMS or at least export/import the data.
- The level of control of access, reporting options, and mobile optimization are vital during institutional adoption.

4.2 System Requirements

In this section, the most important system behaviors as well as features of the We are the Change platform are described. It holds functional, non-functional, and security requirements which spells the behaviour in which the system must run in order to address the needs of the users and administrators.

4.2.1 Functional Requirements

The core functions of the system will lie in the following areas with reference to the civic reporting and volunteering goals of the platform:

1. Issue Reporting with Media Upload

   The system allows users to report public problems (e.g., broken sidewalks, dirty beaches) by filling out a form that includes issue description, and photos. This ensures accurate and detailed reporting.

2. Volunteer Task Browsing and Sign-up

   The application provides tasks that are initiated by communities (e.g., beach cleanups, mural paint). Users have an opportunity to explore their available tasks and sign up to volunteer. In every job, there is information of the safety, allocation of time and place.

3. Sponsor Task Browsing and Support

   Another way in which the platform can allow sponsors to discover the community-led activities, like clean-up campaigns or community improvement activities, and select the activities they want to sponsor. Under each task, a sponsor has an opportunity to see the main details, such as the objective, the time and place, and resource or financial requirements. The sponsors can then send their acceptance to sponsor a task by providing their contact information or preferred mode of support and the organizing team or administrators can contact them on the same to make the arrangements.

4. Admin Dashboard for Task and Report Management

   Reported problems, volunteer tasks, or sponsor supports can be seen, prioritized, and assigned by the administrators via a special dashboard. They also have the ability of closing tasks, editing tasks, or escalating tasks.

4.2.2 Security Requirements

Security Quality Requirements Engineering (SQUARE) Methodology, is a structured nine-step process for identifying, categorizing, and prioritizing security requirements during the early stages of information technology (IT) system development. It was developed by the Software Engineering Institute at Carnegie Mellon University to address common industry challenges where security is often overlooked in requirements engineering, resulting in costly vulnerabilities and weaknesses later in the system lifecycle (Mead, Hough, & Stehney, 2005).

The report provides a detailed description of the SQUARE process, including each step: agreeing on definitions, identifying security goals, developing relevant system artifacts (like use cases and attack trees), conducting a risk assessment, selecting and applying elicitation techniques, and ultimately inspecting and validating the final set of security requirements. It presents case studies to demonstrate the methodology in real-world projects and offers guidance on tailoring the process to different organizational needs (Mead, Hough, & Stehney, 2005).

4.2.2.1 Agreement on Definitions

Term	**Definition**
User	Any individual accessing the platform such as volunteer, sponsor, or admin.
Volunteer	A registered user who signs up for community tasks.
Sponsor	Organizations, institutions, or individuals who support community projects financially or logistically, and may monitor task progress and impact.
Admin	Authorized personnel overseeing reports, volunteers, and system operations.
Issue Report	A user-submitted case regarding a public problem, including text, and photos.
Personal Data	Any data tied to a user’s identity (name, phone, email, volunteer history).
Authentication	Verifying identity before granting system access.
Access Control	Permissions determining which actions users can perform.
Firebase	Backend database and authentication provider used for secure data storage and access.

Table 4: Agreement on definitions

Table 4: Agreement on definitions provides a set of key terms and their definitions used within the system. It explains the different user roles—such as User, Volunteer, Sponsor, and Admin—and clarifies their responsibilities and access levels within the platform. It also defines important

system-related concepts like Issue Report, Personal Data, Authentication, Access Control, and Firebase. Together, these definitions create a shared understanding of how the platform functions, who interacts with it, and how data and access are managed. This table serves as a foundational glossary to ensure consistency and clarity throughout the project documentation.

4.2.2.2 Identification of Security Goals

The primary security goals for the system are:

- Protect personal information of users and volunteers.
- Prevent unauthorized access to system functions and data.
- Ensure integrity and accuracy of submitted reports and volunteer assignments.
- Provide secure communication and data transmission.
- Ensure data availability and recovery in case of system failure.
- Prevent fraudulent or harmful volunteer activities.

4.2.2.3 Develop Artifacts to Support Security Requirements Definition

Artifacts used:

- Use-case scenarios (Report issue, Volunteer sign-up, Admin dashboard actions)
- System architecture (User → Firebase Auth → Database → Admin Panel)
- Stakeholder input (interview in report)
- Functional & non-functional requirements (Chapter 4)

These artifacts validate where security controls are needed.

4.2.2.4 Perform Risk Assessment

Risk	**Impact**	**Likelihood**	**Notes**
Unauthorized access to admin dashboard	High	Medium	Could lead to false updates or misuse.
Data breach of user information	High	Medium	Sensitive personal + location data involved.
Fake or malicious reporting	Medium	High	May overload system or cause operational misuse.
Volunteer impersonation or unauthorized task participation	High	Medium	Health & safety risk to community.
Data loss or corruption	High	Low	System must ensure backups.
Insecure media upload	Medium	Medium	Malicious file uploads risk.

Table 5: Risk Assessment

Table 5: Risk Assessment outlines the key risks associated with the system and evaluates each one based on its potential impact, likelihood, and relevant notes. It highlights major security and operational threats, such as unauthorized access to the admin dashboard, data breaches involving sensitive user information, fake or harmful reports, and impersonation of volunteers. Technical risks like data loss, corruption, or insecure media uploads are also included. By categorizing each risk with its severity and probability, the table helps identify which areas require stronger controls, monitoring, or mitigation strategies. This assessment supports safer system design and ensures that high-impact risks are prioritized during development and deployment.

4.2.2.5 Select Elicitation Techniques

Techniques used and recommended:

- Interview
- Document review
- System use-case analysis
- Brainstorming for cyber threats
- Reference to civic/volunteer systems

4.2.2.6 Elicit Security Requirements

Core security requirements gathered from stakeholders and technical context:

- Protect user accounts and data.
- Ensure safe volunteer participation and verification.
- Secure media uploads and location data.
- Enforce admin-level protection and role-based permissions.
- Track and log activities for accountability.

4.2.2.7 Categorize Requirements

- System: Access control, encryption, availability, monitoring, safety workflows.
- Software: Input validation, role management, frontend secure design.
- Constraints: Legal compliance, language support for security notices.

4.2.2.8 Prioritize Requirements

Prioritization is based on the impact to critical security goals:

- High: Fundamental for system protection and safe operations.
- Medium: Enhances resilience, user safety, or usability but not essential for core security.
- Low: Useful, but has indirect or supporting security impact.

4.2.2.9 Requirements Inspection

A review ensures requirements are:

- Clear and testable
- Aligned with project scope
- Address identified risks
- Feasible given Firebase & Flutter Flow platform

Security Requirement	**Description**	**Priority**
User authentication	Use Firebase to securely authenticate users before access	High
Role-based access control	Admin has privileged permissions separate from users	High
Data encryption in transit & storage	HTTPS + Firebase encryption ensures data confidentiality	High
Secure media upload filter	Validate image uploads and restrict file types	High
Input validation	Sanitize all user inputs to prevent malicious data submission	High
Audit logging	Log admin and user actions for accountability	Medium
Backup & recovery	Maintain data backups and restore ability	Medium
Session timeout	Auto-logout inactive sessions	Medium
Volunteer identity verification	Verify users before volunteer assignment for safety reasons	Medium
Report abuse mechanism	Allow reporting fake/misuse cases to admin	Medium

Table 6: Security Requirements

Table 6: Security Requirements is a summary of the major security requirements of the system including what protection is required, how it will be provided and its priority. It includes some of the key precautions, including user authentication, role-based access control, data encryption, secure file upload, input validation, and location data protection. Other features such as audit logging, backup and recovery, session timeouts, volunteer identity verification and abuse reporting assistance may reinforce the total system reliability and safety of users. The priority column enables the most important controls to be prioritized in order of implementation to have a clear roadmap of creating a secure and trustworthy platform.

4.2.2.10 Threat Modeling and Security Analysis

This section applies the STRIDE threat modeling methodology to analyze security threats facing the "We Are The Change" platform. The platform allows citizens to report public issues and volunteer for community tasks, involving sensitive user data, location

information, media uploads, and administrative functions. STRIDE categorizes threats into six types: Spoofing, Tampering, Repudiation, Information Disclosure, Denial of Service, and Elevation of Privilege. Each identified threat includes its source, type, potential impact, likelihood, and proposed mitigation measures.
The analysis focuses on three critical components identified in the system architecture: the mobile application frontend, Firebase authentication and database backend, and the admin dashboard. Data flows include user registration, issue reporting with photos, volunteer sign-ups, and administrative management of reports and tasks.

4.2.2.10.1 Spoofing Threats

Component	**Details**
Threat Source/Actor	External attackers, malicious users
Description	Attackers attempt to gain unauthorized access by impersonating legitimate users through stolen or weak credentials.
Technical Impact	Unauthorized access to user accounts, ability to submit false reports, sign up for volunteer tasks under false identity, view personal volunteer history.
Business Impact	Compromised trust in platform, fraudulent reports causing resource misallocation, safety risks from unverified volunteers, potential legal liability.
Likelihood	Medium - Common attack vector for web/mobile applications, especially with weak password policies.
Proposed Mitigation	Implement Firebase authentication with multi-factor authentication (MFA), enforce strong password policies (minimum 8 characters with complexity requirements), implement account lockout after failed login attempts, use email verification for new accounts.

Table 7: Threat S-1 User Identity Spoofing

Table 7: Threat S-1 User Identity Spoofing describes a certain security risk when external attackers or malicious users seek to gain access to the legitimate users by using

weak or stolen credentials. It describes the technical impact like unauthorized access, false report submissions, and abuse of the volunteer features, business impact that can include loss of trust, safety concerns, and possible legal problems. This risk is rated medium because credential-based attacks in web and mobile apps are quite a common occurrence. The mitigation planned to maintain this threat would be through the implementation of Firebase authentication with multi-factor authentication, strong password policy, account lock-out due to repeated failure and check emails on new accounts.

Component	**Details**
Threat Source/Actor	Unauthorized users, compromised accounts
Description	Attackers gain access to admin credentials to impersonate system administrators.
Technical Impact	Full control over reported issues, ability to manipulate task assignments, access to all user data, deletion or modification of critical reports.
Business Impact	Complete system compromise, manipulation of community priorities, loss of platform credibility, regulatory compliance violations.
Likelihood	Medium - High-value target but protected by role-based access controls.
Proposed Mitigation	Implement mandatory MFA for all admin accounts, use separate authentication realm for admin access, implement IP whitelisting for admin access, enforce stronger password requirements, conduct regular admin account audits.

Table 8: Threat S-2 Admin Impersonation

Table 8: Threat S-2 Admin Impersonation outlines one of the most impactful security threats to the system where attackers had access to administrator accounts, which made them impersonate system admins. This violation would provide unauthorized users complete

authority over reports, assignment of tasks and sensitive user information so that they can modify or delete essential information. The business effects are that there is a complete compromise of the systems, community will lose trust and there is likelihood of breaking up of the regulations. Admins are a valuable target although the risk has been rated as medium since role-based protections are present. To reduce this risk, the system must implement mandatory MFA to all the accounts of administrators, separate authentication environment of the administrators, IP whitelisting, strong passwords, and periodic audit of administrator accounts.

4.2.2.10.2 Tampering Threats

Component	**Details**
Threat Source/Actor	Malicious users, compromised accounts
Description	Unauthorized modification of issue reports, including description, location, category, or attached photos after submission.
Technical Impact	Corrupted report data, misleading location information, altered priority levels, manipulated status updates
Business Impact	Misallocation of community resources, delayed response to genuine issues, erosion of data integrity, potential harm from misdirected volunteer efforts.
Likelihood	Medium - Firebase security rules provide protection, but misconfiguration is common
Proposed Mitigation	Implement Firebase security rules restricting report modification to original submitter and admins only, use data versioning to track changes, implement digital signatures or checksums for uploaded media, create audit logs for all report modifications

Table 9: Threat T-1 Report Data Manipulation

Based on Table 9: Threat T-1 Report Data Manipulation, this risk is concerned with the possibility of such malicious users or compromised accounts changing the submitted issue reports without permission. This manipulation may change the details of reports, places, type, or the media put on them, resulting in corrupted or distorted information. The technical impact comprises wrong priorities and false status reports, whereas the business impact spans between the misconstrueance of resources along with the lateness of reactions, up to the mistrust to the integrity of the data supplied by the platform. Despite medium rating of likelihood because Firebase rules are effective but can be configured in a wrong way, strong mitigation is needed. The controls that should be recommended are a strict Firebase security policy, only admins or the original reporter should be able to edit, data versioning and media integrity checks, and audit logs that may be used to monitor changes.

Component	**Details**
Threat Source/Actor	External attackers
Description	Attackers inject malicious code through input fields to manipulate database queries and modify backend data.
Technical Impact	Unauthorized data modification, deletion of critical records, bypassing authentication mechanisms, potential data exfiltration.
Business Impact	Loss of data integrity, system downtime, compromised user trust, regulatory violations.
Likelihood	Low-Medium - Firebase's NoSQL structure provides inherent protection, but input validation gaps increase risk.
Proposed Mitigation	Implement strict input validation and sanitization for all user inputs, use Firebase's parameterized queries, apply principle of least privilege to database access, conduct regular security code reviews.

Table 10: Threat T-2 Database Injection Attacks

Table 10: Threat T-2 Database Injection Attacks explains the danger of outsider hackers, who may possibly perform database injection attacks by injecting malicious code into the input fields to modify the queries executed at the backend. In case these attacks are successful, they can result in the unauthorized alteration of data, the erasure of crucial records, bypassing of authentication, or even theft of data. The negative effects on the business include loss of data integrity, possible downtime, deterioration of user trust, and even regulatory penalties. However, the risk has been classified as low to medium which is creditable to the NoSQL architecture of Firebase that provides some inherent

security yet poor input validation can increase the risk. Therefore, to thwart such situations, the system must strictly enforce input validation, incorporate Firebase's parameterized queries, adopt least-privilege data access, and conduct frequent security code reviews.

Component	**Details**
Threat Source/Actor	Malicious users, automated bots
Description	Upload of malicious files disguised as images (e.g., executables, scripts) to compromise system or other users.
Technical Impact	Server-side code execution, storage of malware, potential cross-site scripting (XSS) attacks through image metadata.
Business Impact	System compromise, platform downtime, spread of malware to other users, legal liability.
Likelihood	Medium - Common attack vector for platforms with file upload functionality.
Proposed Mitigation	Implement file type validation (whitelist image formats only), use content-type verification, scan uploads with antivirus/malware detection, store uploads in isolated storage with restricted execution permissions, implement file size limits, strip metadata from uploaded images.

Table 11: Threat T-3 Malicious Media Upload

Table 11: Threat T-3 Malicious Media Upload is about the risk of malicious users or bots uploading harmful files that are disguised as images in order to compromise the system or other users. Such uploads may result in the execution of code on the server, the storage of malware, or the occurrence of XSS attacks hidden in the image metadata. The business impact is major and includes system takeover, loss of uptime, the spread of malware, and, possibly, legal consequences. The probability is thus estimated as being medium since file upload functionalities are usually the most targeted by attackers. In order to lower such risk, the system should implement strict validation of allowed file types, content type verification, file scanning for viruses, uploading in

isolated and execution-restricted locations, file size limiting, and continue with metadata removal from uploaded images.

4.2.2.10.3 Repudiation Threats

Component	**Details**
Threat Source/Actor	Registered volunteers
Description	Volunteers deny signing up for tasks they committed to, causing coordination failures and incomplete community projects.
Technical Impact	Lack of verifiable records linking volunteers to their commitments, inability to prove accountability.
Business Impact	Failed community initiatives, wasted resources in planning, reduced volunteer reliability, damaged platform reputation.
Likelihood	Medium - Common in volunteer-based systems without strong accountability mechanisms.
Proposed Mitigation	Implement comprehensive audit logging of all volunteer sign-ups with timestamps, require explicit confirmation of task participation, send email confirmations with digital receipts, implement non-repudiation through digital signatures for task acceptance, maintain immutable volunteer history records.

Table 12: Threat R-1 Denial of Volunteer Participation

Table 12: Threat R-1 Denial of Volunteer Participation shows the risk of registered volunteers backing out of tasks they previously committed to. This can cause coordination problems and incomplete community projects. Without reliable records, the system cannot verify or enforce responsibility. This leads to failed initiatives, wasted planning resources, and a decrease in trust in volunteer dependability. The likelihood is rated as medium, particularly in platforms that do not have strong accountability measures. To reduce this threat, the system should create detailed audit logs, require volunteers to confirm when they accept tasks, send email receipts, use digital signatures to prevent denial, and keep permanent records of volunteer activity.

Component	Details
Threat Source/Actor	System administrators
Description	Administrators deny performing critical actions such as deleting reports, changing task priorities, or manipulating user data.
Technical Impact	Inability to trace unauthorized administrative actions, lack of accountability for system changes.
Business Impact	Compromised system integrity, inability to investigate incidents, potential legal complications, loss of stakeholder trust.
Likelihood	Low-Medium - Depends on audit logging implementation.
Proposed Mitigation	Implement detailed audit logging for all administrative actions with timestamps and user identification, use tamper-resistant log storage (write-once storage), implement blockchain-based audit trails for critical actions, enforce regular log reviews and anomaly detection, maintain separate backup of audit logs.

Table 13: Threat R-2 Denial of Administrative Actions

System administrators face a threat when they refuse to acknowledge their responsibility for deleting reports and changing task priorities, and user data. The system fails to track these actions because it lacks proper tracking which prevents it from identifying individual users. This creates challenges for incident investigation while removing accountability. The system integrity becomes at risk through this process which damages stakeholder trust and creates potential legal problems. The probability of occurrence depends on the strength of audit logging systems which ranges from low to medium levels. The system needs to create secure audit logs which cannot be modified when it stores data on write-once or blockchain systems. Regularly monitoring logs for suspicious activity and keeping backup copies of audit trails will help protect the system from threats based on Table 13: Threat R-2 Denial of Administrative Actions.

4.2.2.10.4 Information Disclosure Threats

Component	**Details**
Threat Source/Actor	External attackers, unauthorized users, malicious insiders
Description	Unauthorized access to sensitive user information including names, emails, phone numbers, volunteer history, and reported issues.
Technical Impact	Breach of personally identifiable information (PII), exposure of user activity patterns, compromise of authentication credentials .
Business Impact	Violation of data protection regulations, legal penalties, loss of user trust, platform abandonment, reputational damage.
Likelihood	High - User data is a prime target for attackers.
Proposed Mitigation	Implement end-to-end encryption for data in transit (HTTPS/TLS), encrypt sensitive data at rest using Firebase encryption, implement strict role-based access control limiting data access to authorized users only, apply data minimization principles, anonymize or pseudonymize user data where possible, implement secure API authentication, conduct regular security audits

Table 14: Threat I-1 Exposure of Personal User Data

The threat exists because outside attackers together with unauthorized users and malicious insiders can gain entry to protected user data which includes personal information and contact details and volunteer records and reported problems. The system faces major risks because unauthorized users can obtain personal information and track user activities while also getting access to login systems. The platform experiences major business effects which result in regulatory breaches and legal penalties and user trust erosion and platform abandonment. The threat level reaches its highest point because user data stands as a valuable asset when it gets combined with location information. The system requires end-to-end encryption for data transmission and sensitive data protection at rest and strict role-based access control and data minimization and anonymization and secure API access and continuous security audits to solve this problem based on Table 14: Threat I-1 Exposure of Personal User Data.

Component	**Details**
Threat Source/Actor	External attackers, unauthorized developers
Description	Exploitation of insecure API endpoints to extract large amounts of system data including reports, user lists, and task details.
Technical Impact	Mass data exfiltration, exposure of system architecture, access to non-public information.
Business Impact	Data breach, competitive intelligence loss, regulatory violations, system abuse.
Likelihood	Medium - Common vulnerability in mobile and web applications.
Proposed Mitigation	Implement API authentication using Firebase tokens, enforce rate limiting to prevent data scraping, implement API authorization checks, use pagination for data queries to limit bulk extraction, monitor API usage for anomalies, disable unnecessary API endpoints .

Table 15: Threat I-2 Unauthorized API Data Access

Table 15: Threat I-2 Unauthorized API Data Access describes external attackers exploiting insecure API endpoints to extract sensitive data like reports and user lists. It details technical/business impacts and medium likelihood, with mitigations including Firebase token authentication, rate limiting, and pagination (Firebase security checklist, n.d.).

4.2.2.10.5 Denial of Service Threats

Component	Details
Threat Source/Actor	Malicious users, automated attackers
Description	Uploading extremely large or numerous image files to exhaust storage resources and bandwidth.
Technical Impact	Storage quota exhaustion, increased cloud costs, slow system performance, inability for legitimate users to upload media.
Business Impact	Unexpected infrastructure costs, degraded user experience, potential system outage, administrative burden.
Likelihood	Medium - Requires some effort but achievable with automated tools.
Proposed Mitigation	Implement strict file size limits (e.g., 5MB per image), limit number of uploads per report, implement per-user upload quotas, use image compression before storage, implement storage monitoring and alerts, deploy content delivery network (CDN) for efficient delivery.

Table 16: Threat D-1: Media Upload Resource Exhaustion

Table 16: Threat D-1: Media Upload Resource Exhaustion outlines the threat of media upload resource exhaustion, where malicious users or automated attackers upload large or numerous files to deplete storage and bandwidth. The system faces three direct consequences from this issue because storage limits become exceeded and system slowing occurs while legitimate users lose their ability to upload media. The business side faces multiple challenges because of this which include increased infrastructure expenses and poor user satisfaction and system failures and elevated administrative duties. The likelihood is considered medium, as it requires some effort but can be automated. The suggested security measures need file size restrictions and upload limits together with user limits and image compression and storage monitoring and content delivery network (CDN) optimization.

4.2.2.10.6 Elevation Of Privilege Threats

Component	Details
Threat Source/Actor	Malicious users, compromised accounts
Description	Regular users exploit vulnerabilities to gain administrative privileges allowing them to manage all reports, users, and system settings.
Technical Impact	Unauthorized administrative access, ability to modify any system data, bypass all security controls, access confidential information
Business Impact	Complete system compromise, data integrity loss, potential sabotage of community initiatives, regulatory violations, platform shutdown.
Likelihood	Low-Medium - Protected by Firebase security rules but vulnerable if misconfigured.
Proposed Mitigation	Implement strict role-based access control (RBAC) with Firebase security rules, enforce principle of least privilege, separate admin authentication mechanism, conduct regular security audits of permission configurations, implement privilege change logging and alerts, use Firebase custom claims for role management.

Table 17: Threat E-1: User to Admin Privilege Escalation

Table 17: Threat E-1: User to Admin Privilege Escalation is an overview of one of the most important security threats to the system, which is the risk of unauthorized access to the administrative system by unauthorized users. It includes the origin of the threat, how the attackers can take advantage of the vulnerabilities, and the potential technical and business consequences, including loss of data, loss of control over the system, and failure of the service. The probability of this threat is also evaluated in the table and mitigation measures such as role based access control, enforcing of least privileges and frequent security audits are recommended. It is meant to clearly expose the risk and inform on the undertaking of security measures within the project.

Component	Details
Threat Source/Actor	External attackers
Description	Attackers use injection attacks to manipulate database queries and modify user roles or permissions.
Technical Impact	Unauthorized role modification, bypass of authentication mechanisms, access to administrative functions
Business Impact	System compromise, unauthorized control, data breach, service disruption.
Likelihood	Low - Firebase provides protection but input validation failures increase risk.
Proposed Mitigation	Implement comprehensive input validation and sanitization, use parameterized queries, apply Firebase security rules at database level, conduct regular penetration testing, implement web application firewall (WAF), perform regular security code reviews.

Table 18: Threat E-2 SQL/NoSQL Injection for Privilege Escalation

Table 18: Threat E-2 SQL/NoSQL Injection for Privilege Escalation involves the injection attacks of external attackers. It gives a brief introduction on how these attacks may compromise the system and data breach and cause disruption of service through manipulation of database queries, changing user roles and bypassing authentication. It is believed that the probability is small because of Firebase protection, however, vulnerability of input validation can pose a greater threat. In order to solve this, suggesting mitigation strategies, which consist of: input validation, parameterized queries, Firebase database security rules, frequent penetration testing, and web application firewall.

Threat ID	**Threat Name**	**STRIDE category**	**Impact**	**Likelihood**	**Risk Level**
S-1	User Identity Spoofing	Spoofing	High	Medium	High
S-2	Admin Impersonation	Spoofing	Critical	Medium	High
T-1	Report Data Manipulation	Tampering	High	Medium	High
T-2	Database Injection	Tampering	Medium	Low-Medium	Medium
T-3	Malicious Media Upload	Tampering	Medium	Medium	Medium
R-1	Denial of Volunteer Participation	Repudiation	Medium	Medium	Medium
R-2	Denial of Administrative Actions	Repudiation	High	Low-Medium	Medium
I-1	Personal Data Exposure	Information Disclosure	Critical	High	Critical
I-2	Unauthorized API Access	Information Disclosure	High	Medium	High
D-1	Report Flooding	Denial of Service	Medium	High	High
D-2	Media Upload Exhaustion	Denial of Service	Medium	Medium	Medium
E-1	Privilege Escalation to Admin	Elevation of Privilege	Critical	Low-Medium	High
E-2	Injection-based Escalation	Elevation of Privilege	Critical	Low	Medium

Table 19: Risk Summary Table

Table 19: Risk Summary Table provides a systematic perspective of the most prominent security threats may occur in the system in the context of the STRIDE threat-modeling. It lists all the dangers in each category, estimated impact, probability of occurrence, and the degree of risk. It is supposed that the table clearly shows the threats the most dangerous ones such as in data exposure, spoofing, and privilege escalation and help to prioritize the mitigation efforts based on the degree and probability of risk on the scale. The given summary enables making decisions that are easier to make in terms of establishing security controls in the project.

4.2.2.10.7 Conclusion

This threat analysis identified 15 major threats across all STRIDE categories, with critical risks in information disclosure (personal data exposure) and high risks in spoofing, tampering, and privilege escalation. The proposed mitigation measures align with the security requirements already identified in the system requirements section and leverage Firebase's built-in security features while adding additional application-layer protections. Implementation of these mitigations should be prioritized based on risk level, with critical and high-risk threats addressed before platform launch. Regular security audits, penetration testing, and continuous monitoring are essential for maintaining the security posture of the "We Are The Change" platform as it evolves.

4.2.3 Non-Functional Requirements

To ensure that the application is practical, secure, and scalable, the following non-functional requirements are implemented:

1. Security

   Firebase authentication is what ensures that data and access are secure to the user. The data gets secured using HTTPS and designated user ownership (admin, staff, volunteer) with restrictions of the access rights to features to guarantee a safe use of features.

2. Usability and User-Friendliness

   The app is created based on the ideas of Google Material Design and is adapted to both the mobile and web application. The user experience is enhanced by use of clear icons, consistencies in the UI, and easily understood navigations particularly by the first-time users.

3. Performance and Responsiveness

The system is meant to accommodate numerous requests of various users and remain non-laggish. Firestore enables the real-time sync (e.g. status changes of issues) between different devices.

4. Maintainability

   Codebase is modular with compliance to clean architecture. This will enable future updates to integrate it in a government system or with local municipalities with the least code rewrites.

4.3 System Models

System models offer pictorial figures that explain how the data in the We Are The Change works and how the user interacts with this system. The diagrams below will present the functionality and security requirements outlined in Section 4.2 into more organized forms that will inform the implementation but also confirm completeness with the stakeholder requirements.

This part will provide two important models: the Data Flow Diagram (DFD) that demonstrates the flow of information among users, processes, and Firebase storage; and Use Case Diagrams that depict the interaction of the core actors (volunteers, sponsors, admins). Both models guarantee the traceability of requirements to design and ensure that reporting, volunteering, and approval workflows are covered and security controls such as authentication gates are mentioned.

Level 0 of DFD gives high-level data flows and detailed use cases dissect scenarios. Collectively, these models are used as blueprints of Chapter 5 ERD and UI design to minimize risks of implementation by visualizing them early.

4.3.1 Data Flow Diagram (DFD)

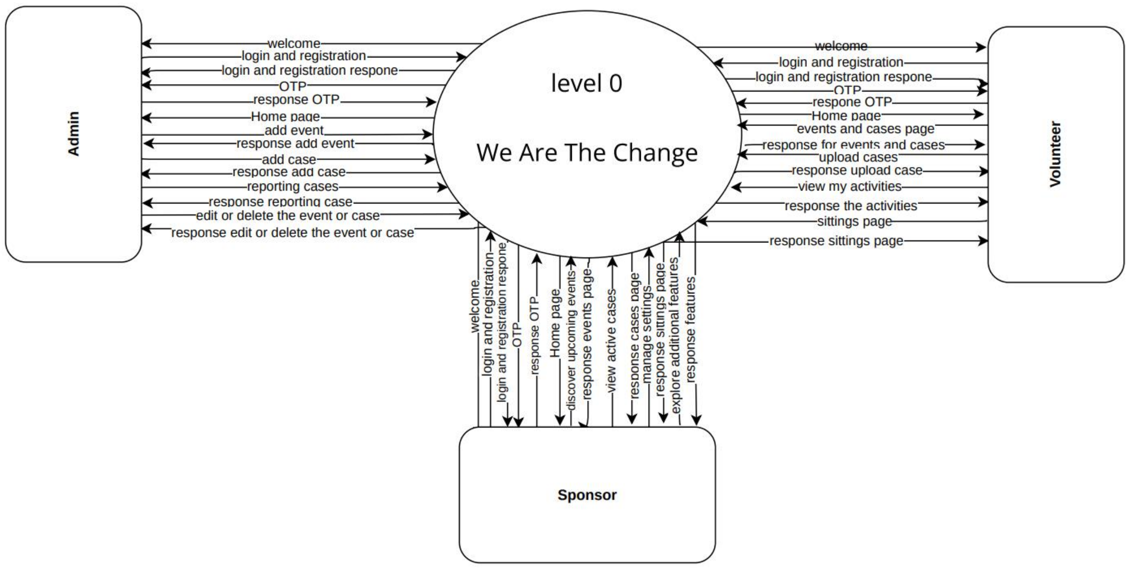


Figure 2: DFD level 0

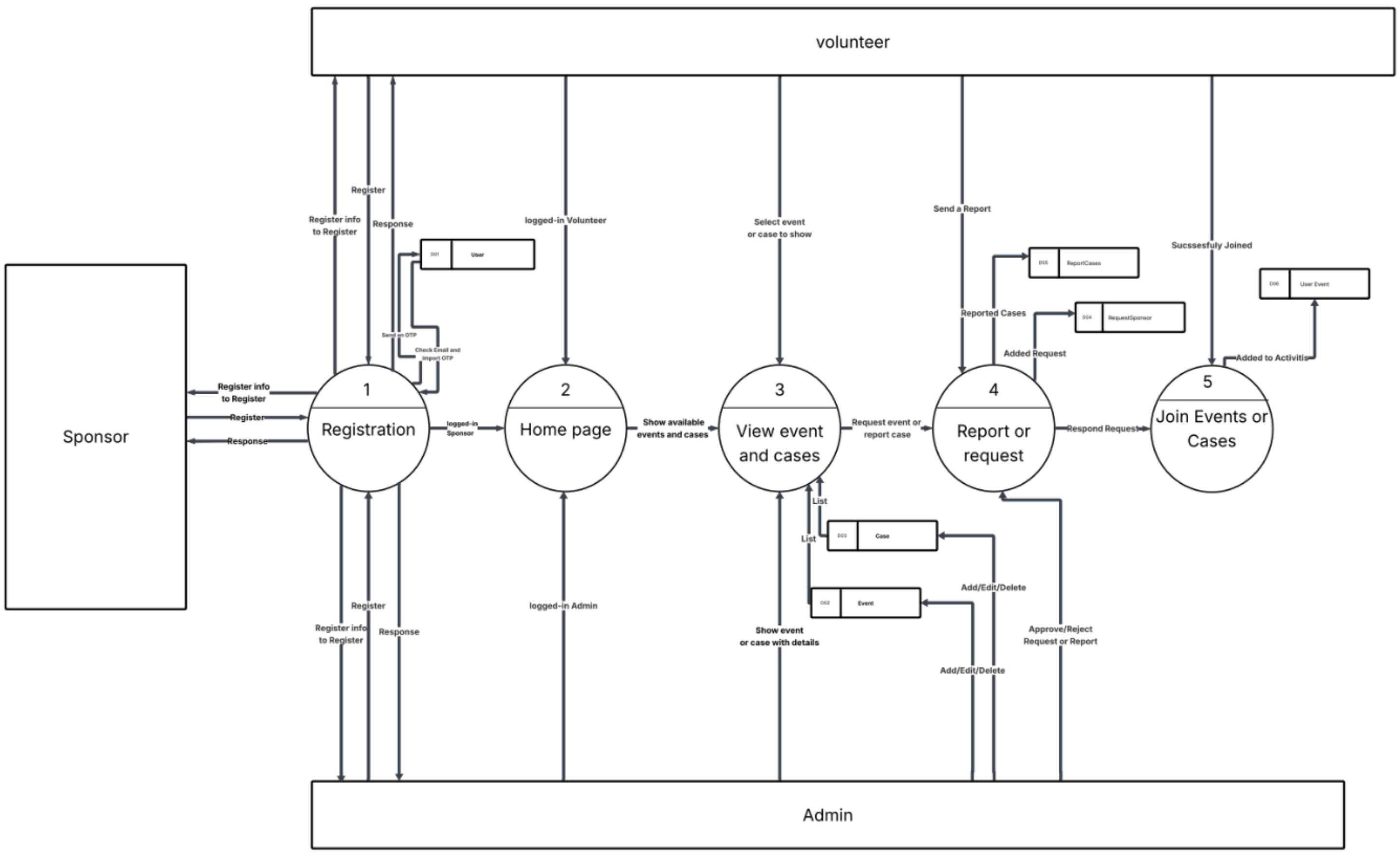


Figure 3: DFD level 1

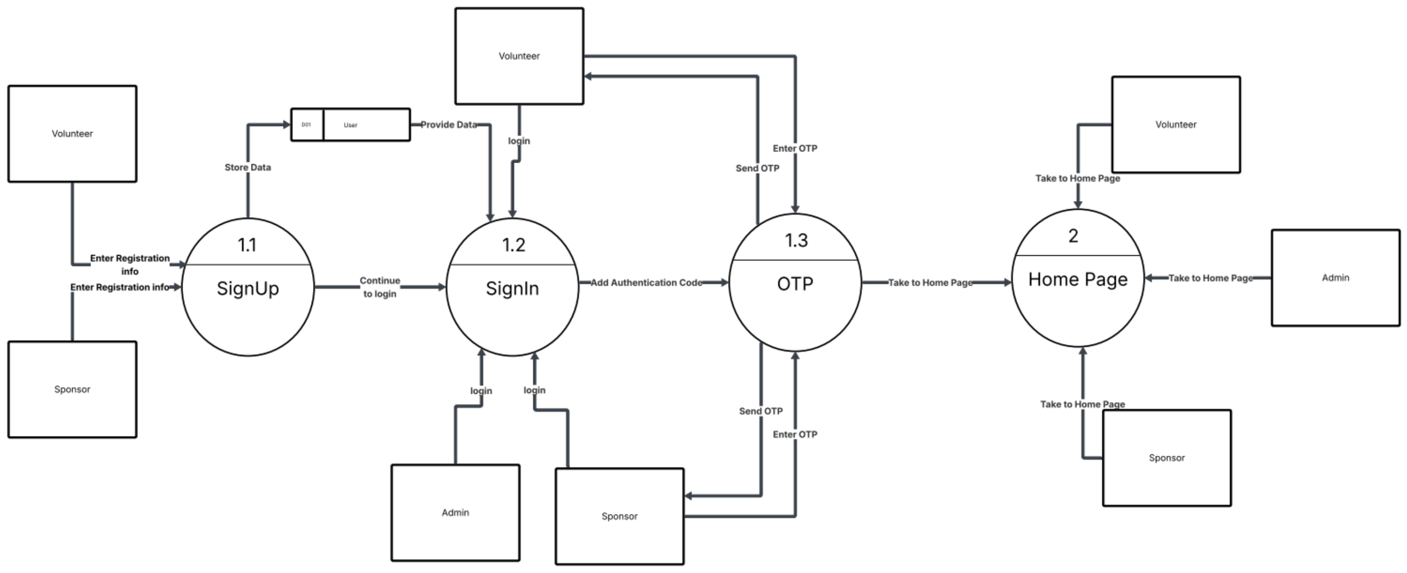


Figure 5: DFD level 2 Registration

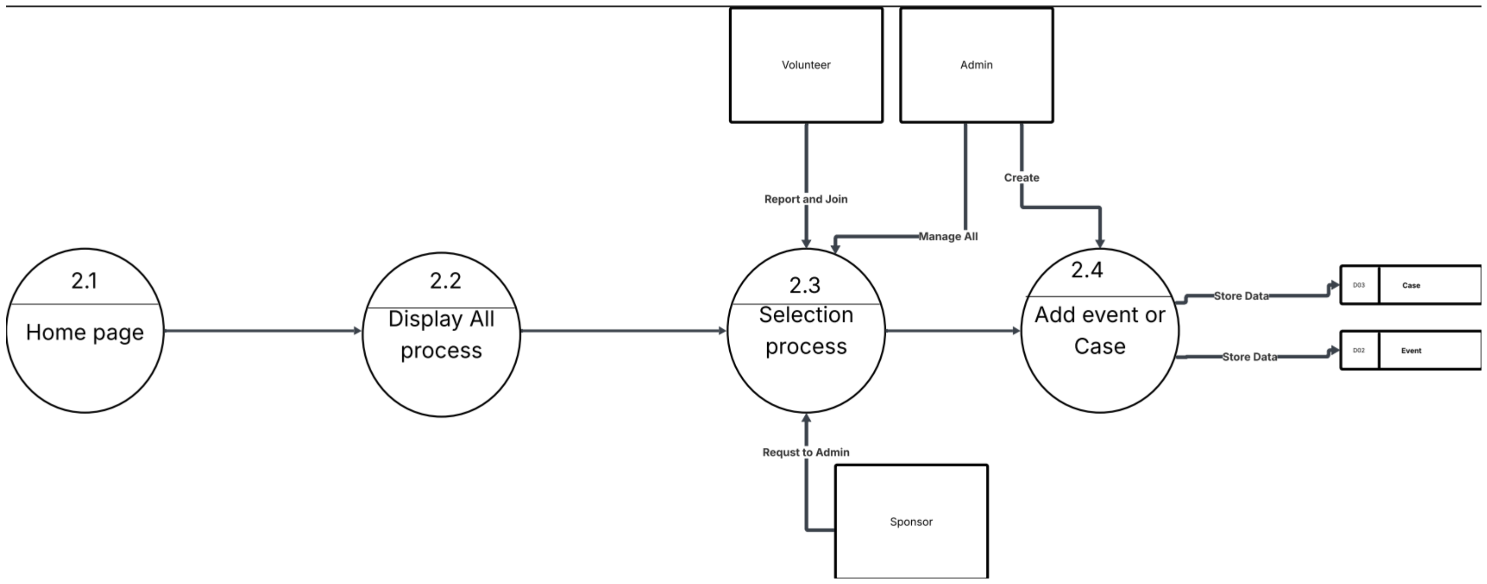


Figure 4: DFD level 2 Home Page

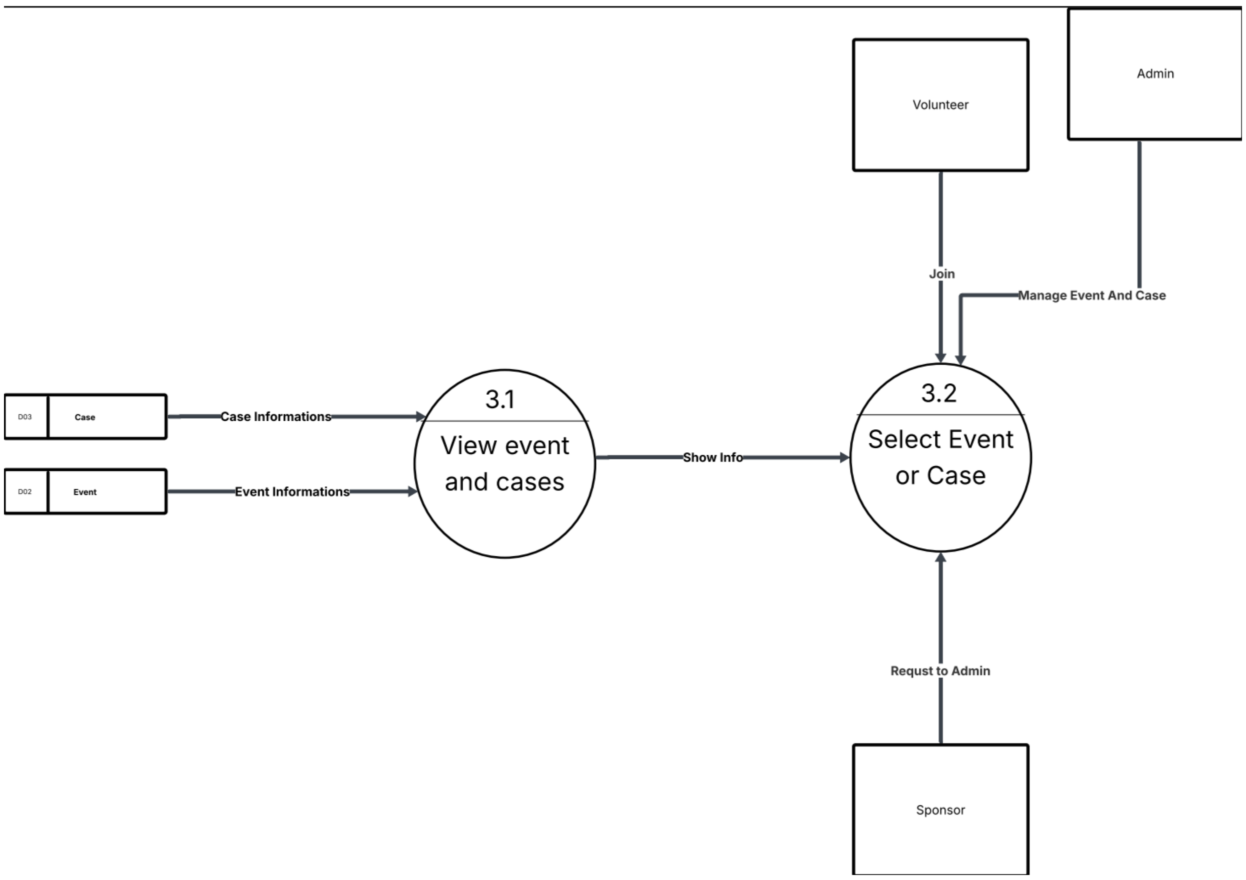


Figure 7: DFD level 2 View event and cases

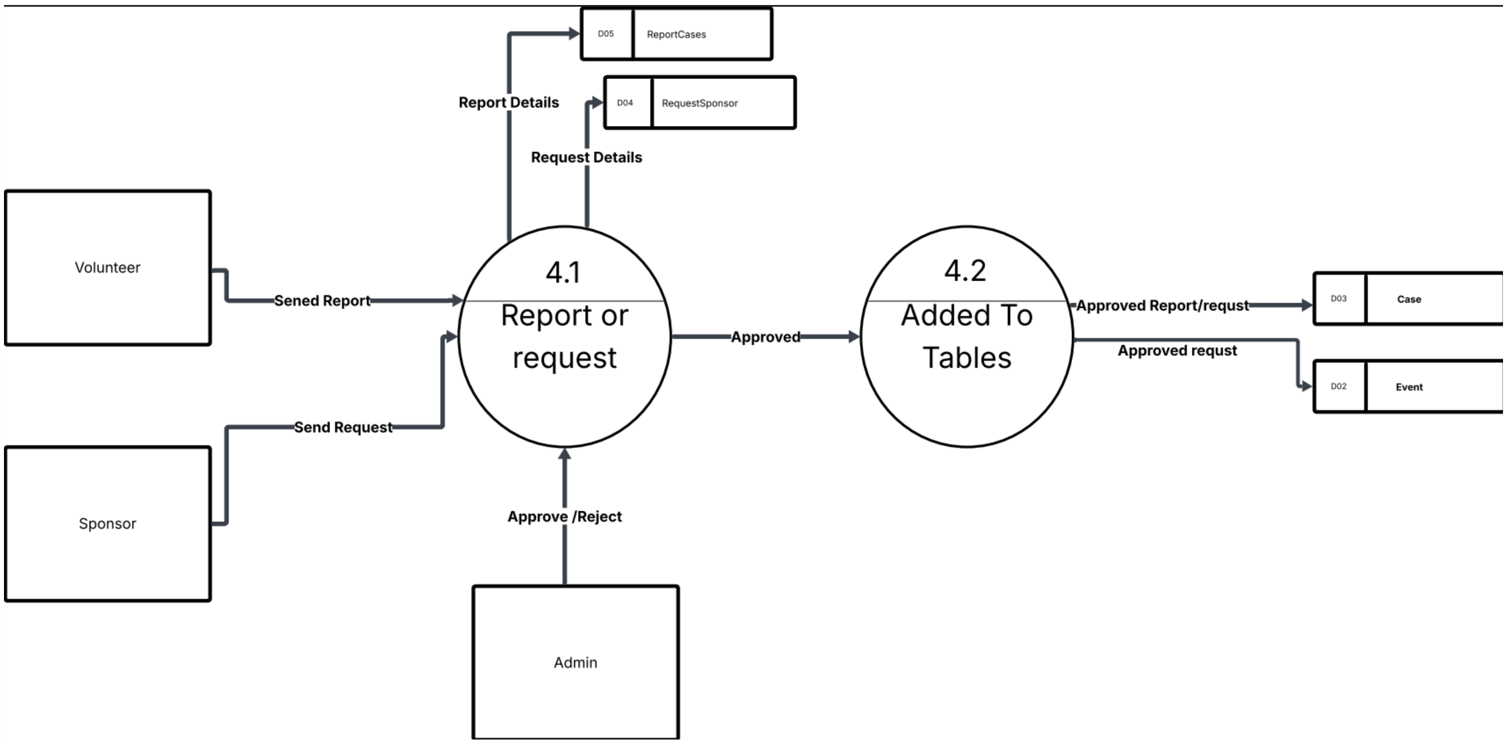


Figure 6: DFD level 2 Report or Request

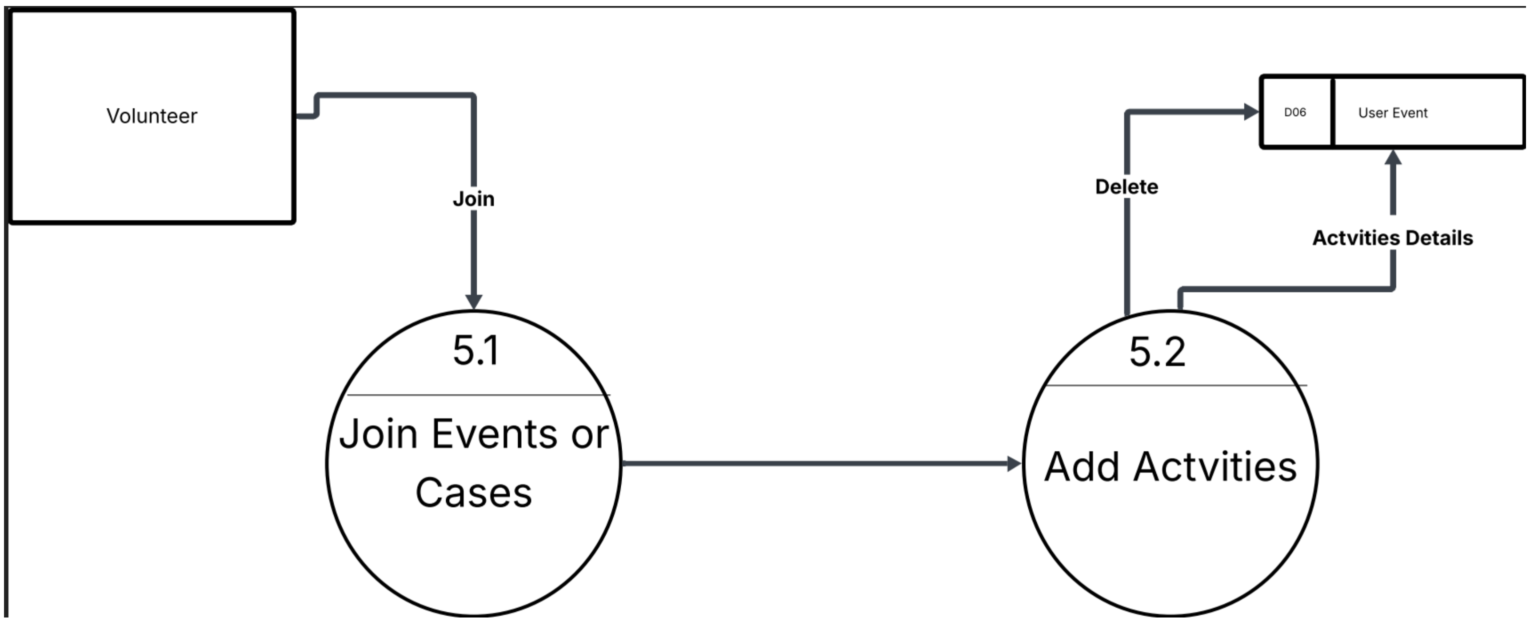


Figure 8: DFD level 2 Join events or cases

4.3.1.1 Data Flow Diagram (DFD) level 0

Figure 2: DFD level 0 of the We Are The Change gives us a high-level overview of how the key actors interact with the system and what the exchange of information between the key actors and the system entails. The single process called We Are The Change is placed in the middle of the diagram and it represents the entire platform as a single logical system that integrates the authentication, case management, volunteering and events.

The Admin actor interacts with the system with the left to carry out management functions. The inputs made by the admin include registering and logging in, one-time password (OTP), adding event, adding case, reporting case and updating or deleting event/case. The system, in turn, provides such answers as login response, registration response, OTP response, home page, response add event, response add case, response reporting case, and response edit/ delete event/case. These streams demonstrate how the admins manage what is posted and check what is being done on the site.

At the right, there is the Interface with the system of user-facing operations. One of the inputs offered by volunteers is the login and registration information, OTP, home page visit, events and cases page, upload cases, my activity page, and settings page. There can be similar system outputs such as login and registration response, OTP response, home page, response to events and cases, response upload case, response the activities and response settings page. These flows depict the process where volunteers can be registered, Browse the available cases and events, make reports, and track their activity.

The Sponsor actor then transfers the data at the bottom with the same central process. Sponsors provide data like the input of their login and registration details, OTP, home page access, events and cases viewing, sponsor request, and other options. The system provides the response of a login and registration, OTP response, home page, response events and cases, response sponsor request, and response other features. This section of the DFD indicates that the sponsors can authenticate and review the community needs and offer support via the platform. On balance, the Level 0 DFD is an overview of all external actors and their key data streams, and it is ensured that the core functionalities, which are to be outlined, including authentication, reporting, volunteering, and sponsoring, are viewed at the utmost level of abstraction.

4.3.1.2 Data Flow Diagram (DFD) level 1

Figure 3: DFD level 1 of We Are The Change splits the main system into five central processes, that is, the way in which volunteers, sponsors and admins engage with specific system functions. With this level of data flows, it is possible not only to identify the person who communicates with the platform, but also to validate and store the user inputs, as well as to utilize them to control the events and cases in various modules.

Process 1 (Registration): This process deals with the creation of accounts of volunteers, sponsors, and admins. Registration information is entered by the users and sponsors and compared with the User data store and OTP verification and a registration response is sent by the system and a logged-in user or sponsor session is created.

Process 2 (Home page) accepts the users (volunteer, sponsor, admin) who are logged in and presents the home interface where the available events and cases appear depending on the user.

Process 3 (View event and cases) enables the volunteers to choose a certain event/case among the lists stored in Case and Event data stores. The system reacts with the display of detailed information about the chosen item.

Process 4 (Report or request) allows volunteers to create new case reports or sponsors to create new support requests; new case reports and support request data are stored in ReportCases and RequestSponsor and admins are able to approve, reject, add, edit, or delete records.

Process 5 (Join Events or Cases) allows the volunteers to enroll to approved events or cases with the result of updating the UserEvent data store and adding successful completion of their activities to the list. Collectively, the processes demonstrate how the system facilitates secure registration, browsing, reporting, sponsoring, and volunteering processes in a comprehensive manner.

4.3.1.3 Data Flow Diagram (DFD) level 2 Registration

Figure 5: DFD level 2 Registration is dedicated to the process of the authentication process decomposition in We Are The Change into three detailed sub-processes: SignUp (1.1), SignIn (1.2), and OTP verification (1.3) and the further navigation to the Home Page (2). It demonstrates the process of the volunteers, sponsors, and admins creating accounts, logging in, and being redirected to their primary interface with the successful verification.
In process 1.1 (SignUp), volunteers and sponsors can input their registration information that is stored in the User data store. Once the data is saved, the system lets the user know that it was registered successfully, and the user can now move on to logging in. Process 1.2 (SignIn) deals with the process of the login of volunteers, sponsors, and admins; the system compares the entered credentials with the User data store and, in case they are valid, runs the process 1.3 by including the requirement of the additional authentication code. Within process 1.3 (OTP), the system sends the user a one-time password, accepts the typed OTP, verifies it and on success, hands the control over to process 2 (Home Page). Lastly, process 2 presents the valid home page view as the role (volunteer, sponsor or admin) being logged in is, and finally, a secure multi-factor authentication process is completed before system features are granted access.

4.3.1.4 Data Flow Diagram (DFD) level 2 Home Page

Error! Reference source not found. describes how “We Are The Change” presents and deals with all the available cases and events of the various user roles. Once process 2.1 (Home Page) has completed successful login and OTP verification, the control flows to process 2.2 (Display All process) whereby the system will get the full list of the active cases and events using the Case and Event data stores. This will guarantee that volunteers, sponsors and admins have a current summary of what is going on in the community as soon as they access the home interface. Process 2.3 (Select event or case) gives the user the opportunity to interact with particular records. This is how volunteers look at the specifics of a case, submit an issue or enter an event or case, and how sponsors look at cases and submit support requests. Admins use the same process to handle all the items such as approval, amending, or deleting events and cases. The system reacts and provides the chosen case or event information in the relevant data store and sends user requests (join, report, manage, and request support) to the relevant processes of higher levels. The breakdown reveals the use of home page as central point of navigation linking user roles to the underlying case and event management features in a systematic manner.

4.3.1.5 Data Flow Diagram (DFD) level 2 View Events and Cases

Figure 7: DFD level 2 View event and cases describes the operations of the system in retrieving and presenting detailed information on all available cases and events and how the users act on this information. Process 3.1 (View event and cases) will read Case data store (and Case Informations) and Event data store (and Event Informations) before combining them into a single list that is displayed in the user interface.

Process 3. 2 (Select Event or Case) is activated when a volunteer, an administrator, or even a sponsor selects a certain item on this list. This process allows volunteers to participate in an event or case and admins to run events and cases (such as approving or updating). Sponsors are able to place a request to an administration about a selected case or occasion, including provision of funds or material. The process then sends the chosen item information and user activities to the corresponding upper level workflows (joining, managing, or sponsoring) so that each interaction will be associated with the corresponding case or event record in the data stores.

4.3.1.6 Data Flow Diagram (DFD) level 2 Report or Request

Figure 6: DFD level 2 Report or Request covers the process of the creation, review, and storage of new community reports and sponsor requests in the system. The process 4.1 (Report or request) also has the inputs of send report by volunteers and send request by sponsors as input. ReportDetails and RequestDetails are reported data that is stored in the ReportCases and the RequestSponsor data stores respectively. The admin then examines such entries and makes an approve/reject decision which is sent back to process 4.1.

In case a report or request has been approved, it is referred to process 4.2 (Added To Tables). It transforms the authorized report/request into operational records which are written into the Case and Event data stores. Approved reports are new or updated cases whereas approved sponsor requests are associated with relevant events. By doing so, the Level 2 layout will provide a complete life cycle of the input of the community to formally accepted cases and events, and the administration of the control of the validation and quality of the input before the entries can influence the main platform.

4.3.1.7 Data Flow Diagram (DFD) level 2 Join Events or Cases

Figure 8: DFD level 2 Join events or cases describes the process of making a decision to join the system, which is registered as an activity of a volunteer. In process 5.1 (Join Events or Cases), volunteer submits a Join request when he or she has identified a particular approved event or case through the interface. The process confirms the event or case to be joinable (not full or closed) and then transmits the confirmed participation to process 5.2 (Add Activities).

Process 5.2 (Add Activities) adds or modifies a record in the User Event data store containing the activity detail of a volunteer, including user ID, event/ case-ID and status (joined/active). As can be seen in the diagram, activities may subsequently be removed out of this data store to represent cancellations or events that have been accomplished. By doing so, the Level 2 model explains how the system will monitor volunteering promises in an organized fashion, which will allow such an option as My Activities and future reporting about the level of participation.

4.3.2 Use cases

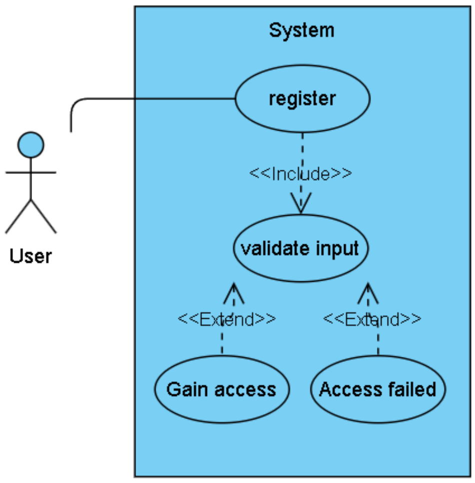


Figure 9: User Registration and Access Validation Use Case

Figure 9: User Registration and Access Validation Use Case outlines the way a user operates within the system to create an account and make an attempt to log in. It begins with the user making registration action, and the system checks the data inputted in the process of registration. Based on the outcome of the validation, the flow is extended either to a successful point where the user is allowed to access, or to a failure point where access is denied and the registration is not finalized.

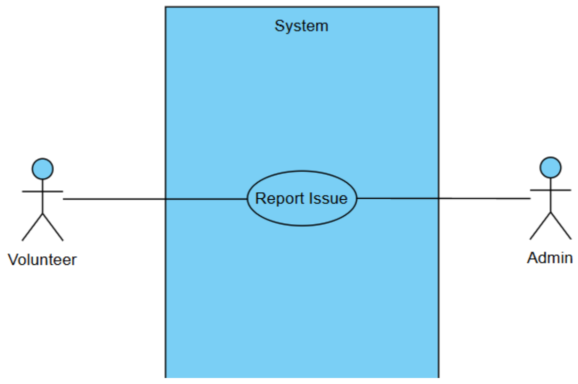


Figure 10: Issue Reporting Use Case

Figure 10: Issue Reporting Use Case explains the interaction of the volunteer with the system to report a problem that needs the attention of the administration. This happens with the volunteer posting an issue via the system that logs the information and provides it to the admin. The admin would then look at the reported issue through the system interface and take subsequent measures, including may be resolving it.

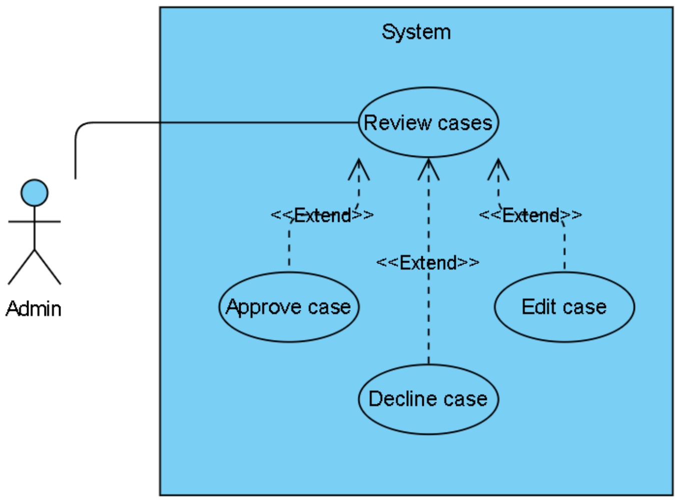


Figure 11: Case Review and Decision Use Case

Figure 11: Case Review and Decision Use Case defines the interaction between an admin with the system, where the admin views reported cases and makes a decision. It starts with the admin getting the list of pending cases and reviewing them in the function of Review cases. Based on the case information, the admin may extend the flow by approving the case, declining the case, or editing the case details before making a final decision.

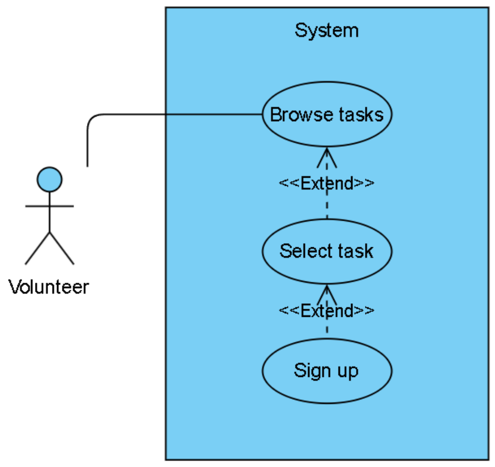


Figure 12: Volunteer Task Browsing and Signup Use Case

Figure 12: Volunteer Task Browsing and Signup Use Case outlines the way in which a volunteer uses the system to browse and volunteer to do particular tasks. This starts when the volunteer goes through the list of published tasks and reads their description. Then, the volunteer will choose a particular task of interest and the flow is continued by registering to that task which captures the participation of the volunteer and links him up with the activity of choice in the system.

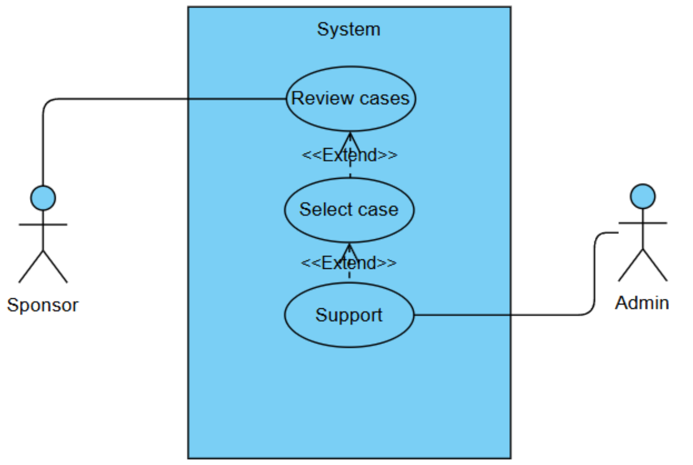


Figure 13: Sponsor Case Review, Support, and Admin Decision Use Case

Figure 13: Sponsor Case Review, Support, and Admin Decision Use Case describes how a sponsor and an admin interact with the system to process support for reported cases. The sponsor begins by using the "Review cases" function to browse the list of available cases and then selects a specific case to support. After the sponsor submits support for the selected case, the admin reviews the supported case and either accepts or denies the support request, with the final decision recorded and reflected in the system.

Chapter 5
System Design

5.1 Introduction

The chapter about the System Design explains the organization of the platform of We Are The Change and how the key elements of the platform collaborate to fulfill the demands stated in the previous chapters. It converts the business and security needs into a technical design, which is used in implementation and testing. In this chapter, the database schema is introduced, being used to organize the main data including users, reports, and volunteer tasks; the design of the user interface through mobile screens; the design of the procedure as it defines how the essential functions like registration, issue reporting, and task assignment are carried out; and the pseudocode, which is used to describe the workflows through which the programmers should implement the program. It also describes the flow of the system as a whole using diagram and makes a description of any significant algorithms or logic that were employed to process operations. With their clear description, the System Design chapter is able to guarantee that the application is maintainable, scalable, and secure as well as offer a common technical perspective among the developers and the stakeholders.

5.2 Entity Relationship Diagram (ERD)

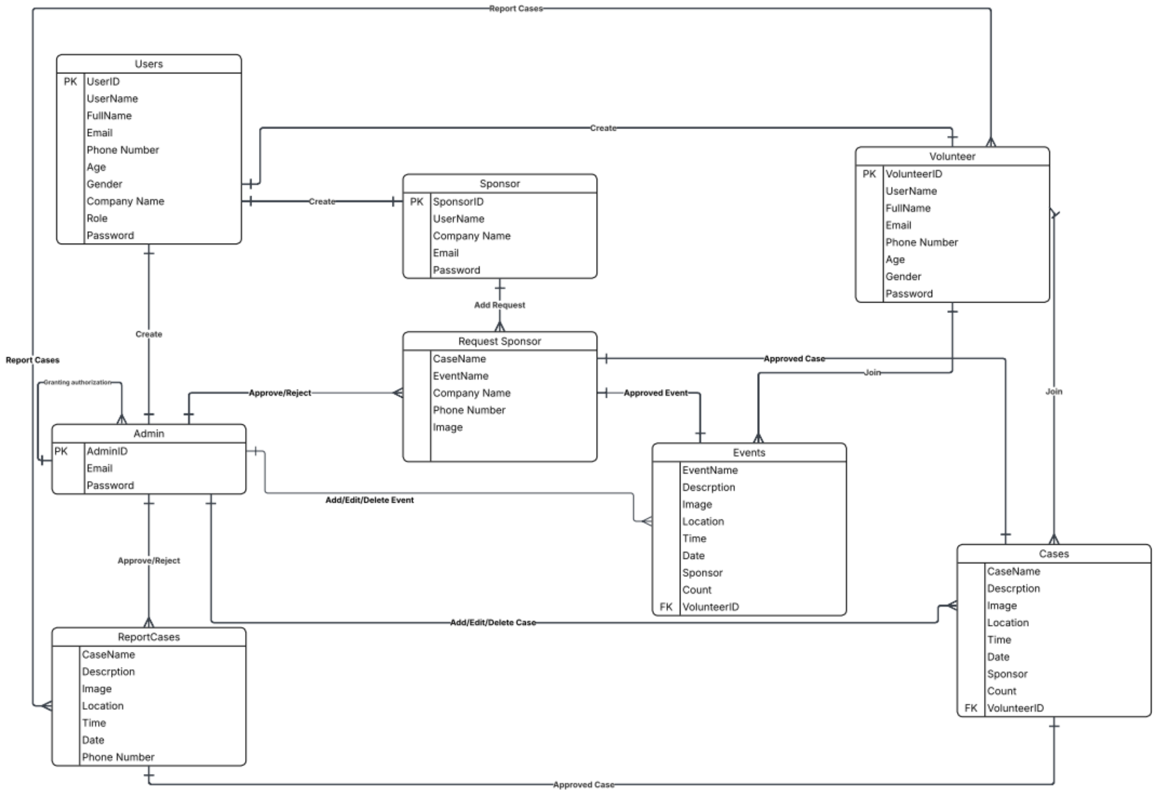


Figure 14: Entity Relationship Diagram (ERD)

5.3 Database Schema

A database schema is the formal description of how data is structured inside a database system. It specifies the types of things that can be stored (such as entities and their attributes), how these things are related to each other (relationships and cardinality constraints), and the rules that the data must follow for storage and querying. In relational databases, this is usually represented as tables with columns and keys, while in conceptual diagrams it appears as entities and relationships, but in all cases the schema's main purpose is to organize data efficiently for storing, retrieving, and maintaining it (Uschold, 2015).

5.4 Data Dictionary

Data dictionary is a tool that provides details and information in a data set which usually includes key insights into how the data can be used and meaning of the attributes and how it is created (Buchanan, et al., 2021) and serves as a foundation for the system database, and based on "We Are The Change" mobile application entity diagram the following data dictionary tables will be created.

5.4.1 Users

Attribute name	Data type	Primary Key	Foreign Key	Null	Unique	Other Constraint
UserID	String	PK	-	-	Yes	-
UserName	String	-	-	-	-	-
FullName	String	-	-	-	-	-
Email	String	-	-	-	Yes	-
Phone Number	Number	-	-	-	Yes	-
Age	Integer	-	-	-	-	-
Gender	String	-	-	-	-	-
Company Name	String	-	-	-	-	-
Role	String	-	-	-	-	-
Password	String	-	-	-	No	-

Table 20: Users Table

5.4.2 Volunteer

Attribute name	Data type	Primary Key	Foreign Key	Null	Unique	Other Constraint
VolunteerID	String	PK	-	No	Yes	-
UserName	String	-	-	No	Yes	-
FullName	String	-	-	No	-	-
Email	String	-	-	No	Yes	-
Phone Number	Number	-	-	-	Yes	-
Age	Integer	-	-	-	-	-
Gender	String	-	-	-	-	-
Password	String	-	-	-	No	-

Table 21: Volunteer table

5.4.3 Sponsor

Attribute name	Data type	Primary Key	Foreign Key	Null	Unique	Other Constraint
SponsorID	String	PK	-	No	Yes	-
UserName	String	-	-	No	Yes	-
Company Name	String	-	-	No	-	-
Email	String	-	-	No	Yes	-
Password	String	-	-	No	Yes	-

Table 22: Sponsor Table

5.4.4 Admin

Attribute name	Data type	Primary Key	Foreign Key	Null	Unique	Other Constraint
AdminID	String	PK	-	No	Yes	-
Email	String	-	-	No	Yes	-
Password	String	-	-	No	-	-

Table 23: Admin Table

5.4.5 Cases

Attribute name	Data type	Primary Key	Foreign Key	Null	Unique	Other Constraint
CaseName	String	-	-	No	Yes	-
Description	String	-	-	Yes	-	-
Image	Image path	-	-	-	-	-
Location	String	-	-	-	Yes	-
Time	Time	-	-	-	Yes	-
Date	Date	-	-	-	-	-
Sponsor	integer	-	-	-	-	-
Count	integer	-	-	-	-	-

Table 24: Cases table

5.4.6 Events

Attribute name	Data type	Primary Key	Foreign Key	Null	Unique	Other Constraint
EventName	String	-	-	-	Yes	-
Description	String	-	-	-	-	-
Image	Image path	-	-	-	-	-
Location	String	-	-	-	Yes	-
Time	Time	-	-	-	Yes	-
Date	Date	-	-	-	-	-
Sponsor	String	-	-	-	-	-
Count	integer	-	-	-	-	-

Table 25: Events Table

5.4.7 Request Sponsor

Attribute name	Data type	Primary Key	Foreign Key	Null	Unique	Other Constraint
CaseName	String	-	FK	No	No	-
EventName	String	-	FK	No	No	-
Company Name	String	-	-	No	No	-
Phone Number	Number	-	-	-	Yes	-
Image	Image path	-	-	-	Yes	-

Table 26: Request Sponsor Table

5.4.8 Report Cases

Attribute name	Data type	Primary Key	Foreign Key	Null	Unique	Other Constraint
CaseName	String	-	-	-	Yes	-
Description	String	-	-	-	-	-
Image	Image path	-	-	-	-	-
Location	String	-	-	-	Yes	-
Time	Time	-	-	-	Yes	-
Date	Date	-	-	-	-	-
Phone Number	Number	-	-	-	-	-

Table 27: Report Cases Table

5.5 We Are The Change mobile application UI Design

5.5.1 Introduction

The following section depicts the user interface (UI) design of the We Are The Change mobile application, which has features of managing volunteer cases and events, sponsor donations and administrative control. The design is based on the basic functionalities such as user authentication to sponsors, volunteers and administrators; home pages with tabs to the cases and events; the case upload and support pledges based on form; and tools to review, edit, approve, reject, delete or add content directly to the page, as discussed in previous sections. Such interfaces place a lot of emphasis on intuitive navigation, clarity, and effective completion of tasks to improve the usability between the different user roles. Further subsections are used to demonstrate the important screens and interactions by using figures to show how it conforms to the best practices of mobile UI design in terms of accessibility and interaction.

5.5.2 User Signup and Sign in

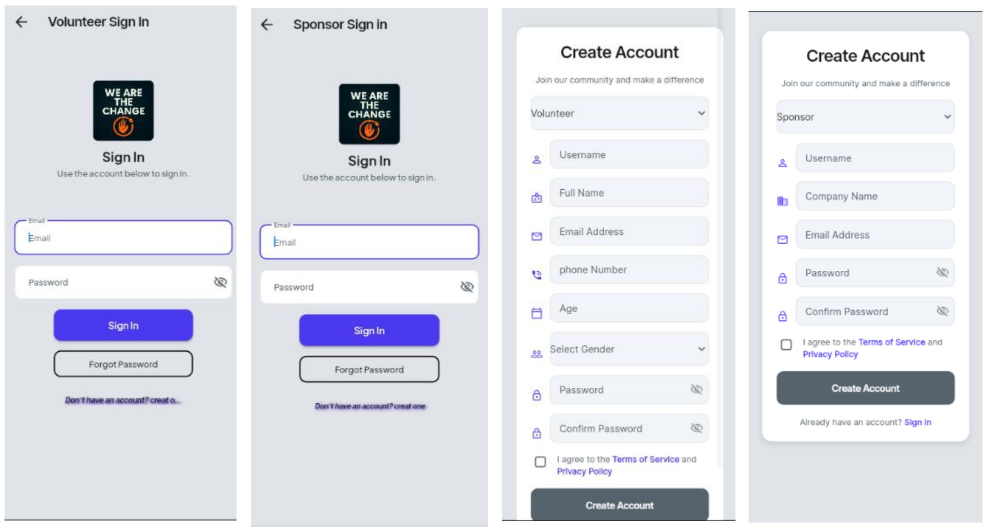
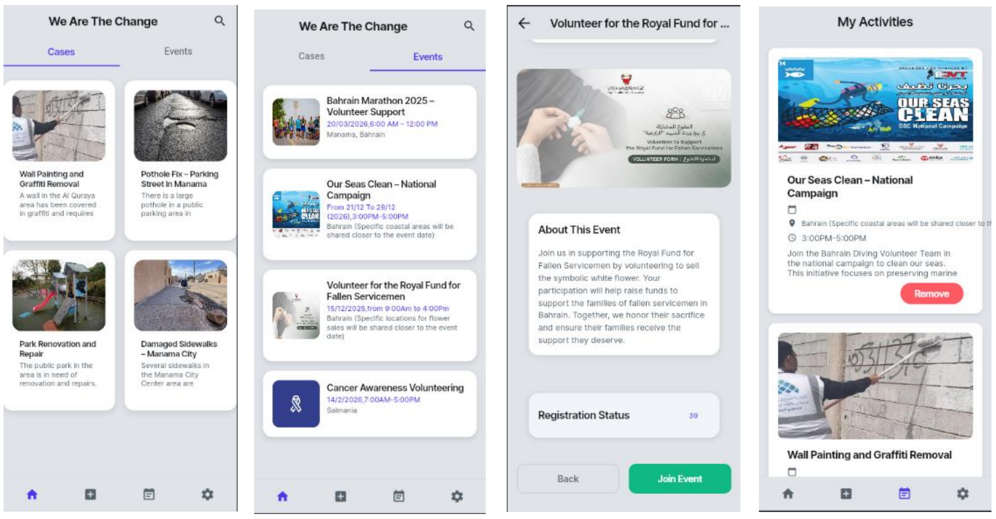


Figure 15: Signup / Login Interface

As a user, be it a sponsor or a volunteer, a user can sign in with valid credentials. In case the user is not a registered user, they may create an account. The system has the administrator accessing with the sponsor or volunteer log-ins, as shown in Figure 15: Signup / Login Interface.

5.5.3 Volunteer Case/Event Sign up


We Are The Change
Cases
Events
Wall Painting and Graffiti Removal
Pothole Fix – Parking Street in Manama
Park Renovation and Repair
Damaged Sidewalks – Manama City
We Are The Change
Cases
Events
Bahrain Marathon 2025 – Volunteer Support
Our Seas Clean – National Campaign
Volunteer for the Royal Fund for Fallen Servicemen
Cancer Awareness Volunteering
Volunteer for the Royal Fund for ...
About This Event
Registration Status
39
Back
Join Event
My Activities
Our Seas Clean – National Campaign
Remove
Wall Painting and Graffiti Removal

Wall Painting and Graffiti Removal
A wall in the Al Quraya area has been covered in graffiti and requires

Pothole Fix – Parking Street in Manama
There is a large pothole in a public parking area in

Park Renovation and Repair
The public park in the area is in need of renovation and repairs.

Damaged Sidewalks – Manama City
Several sidewalks in the Manama City Center area are

Bahrain Marathon 2025 – Volunteer Support
20/03/2026,6:00 AM – 12:00 PM
Manama, Bahrain

Our Seas Clean – National Campaign
From 21/12 To 29/12 (2026),3:00PM-5:00PM
Bahrain (Specific coastal areas will be shared closer to the event date)

Volunteer for the Royal Fund for Fallen Servicemen
15/12/2025,from 9:00Am to 4:00Pm
Bahrain (Specific locations for flower sales will be shared closer to the event date)

Cancer Awareness Volunteering
14/2/2026,7:00AM-5:00PM
Salmania

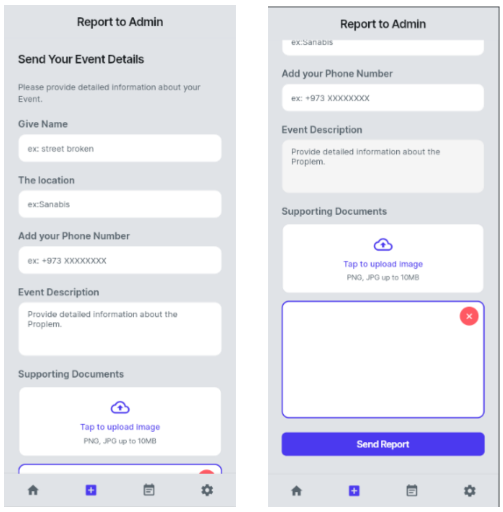
About This Event
Join us in supporting the Royal Fund for Fallen Servicemen by volunteering to sell the symbolic white flower. Your participation will help raise funds to support the families of fallen servicemen in Bahrain. Together, we honor their sacrifice and ensure their families receive the support they deserve.

Our Seas Clean – National Campaign
Bahrain (Specific coastal areas will be shared closer to th
3:00PM-5:00PM
Join the Bahrain Diving Volunteer Team in the national campaign to clean our seas. This initiative focuses on preserving marine

Figure 16: Volunteer Case/Event Sign up

As shown in Figure 16: Volunteer Case/Event Sign up, on the volunteer home page, there are two main tabs, which are Cases and Events. These tabs will allow volunteers to have in-depth information on available cases and events. The volunteers can subsequently enroll to any chosen case or event through the respective tab.

5.5.4 Volunteer Upload Cases


Report to Admin

Send Your Event Details

Please provide detailed information about your Event.

Give Name

ex: street broken

The location

ex:Sanabis

Add your Phone Number

ex: +973 XXXXXXXX

Event Description

Provide detailed information about the Proplem.

Supporting Documents

Tap to upload image

PNG, JPG up to 10MB

Report to Admin

ex:Sanabis

Add your Phone Number

ex: +973 XXXXXXXX

Event Description

Provide detailed information about the Proplem.

Supporting Documents

Tap to upload image

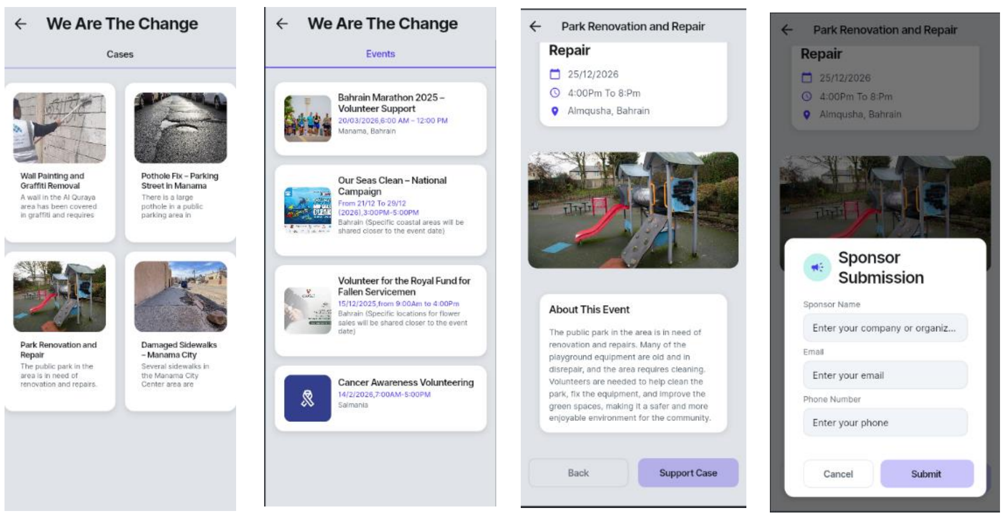
PNG, JPG up to 10MB

Send Report

Figure 17: Volunteer Upload Case

Figure 17: Volunteer Upload Case depicts that the volunteers are able to add new cases by filling an upload case form that has case title, location, phone number, description and photo attachments sections. This option allows volunteers to effectively provide comprehensive information on the case to review and take additional steps.

5.5.5 Sponsor Case/Event Support


We Are The Change

Cases

Wall Painting and Graffiti Removal
A wall in the Al Quraya area has been covered in graffiti and requires

Pothole Fix – Parking Street in Manama
There is a large pothole in a public parking area in

Park Renovation and Repair
The public park in the area is in need of renovation and repairs.

Damaged Sidewalks – Manama City
Several sidewalks in the Manama City Center area are

We Are The Change

Events

Bahrain Marathon 2025 – Volunteer Support
20/03/2026,6:00 AM – 12:00 PM
Manama, Bahrain

Our Seas Clean – National Campaign
From 21/12 To 29/12 (2026),3:00PM-5:00PM
Bahrain (Specific coastal areas will be shared closer to the event date)

Volunteer for the Royal Fund for Fallen Servicemen
15/12/2025,from 9:00Am to 4:00Pm
Bahrain (Specific locations for flower sales will be shared closer to the event date)

Cancer Awareness Volunteering
14/2/2026,7:00AM-5:00PM
Salmania

Park Renovation and Repair

Repair
25/12/2026
4:00Pm To 8:Pm
Almqusha, Bahrain

About This Event
The public park in the area is in need of renovation and repairs. Many of the playground equipment are old and in disrepair, and the area requires cleaning. Volunteers are needed to help clean the park, fix the equipment, and improve the green spaces, making it a safer and more enjoyable environment for the community.

Back | Support Case

Park Renovation and Repair

Repair
25/12/2026
4:00Pm To 8:Pm
Almqusha, Bahrain

Sponsor Submission

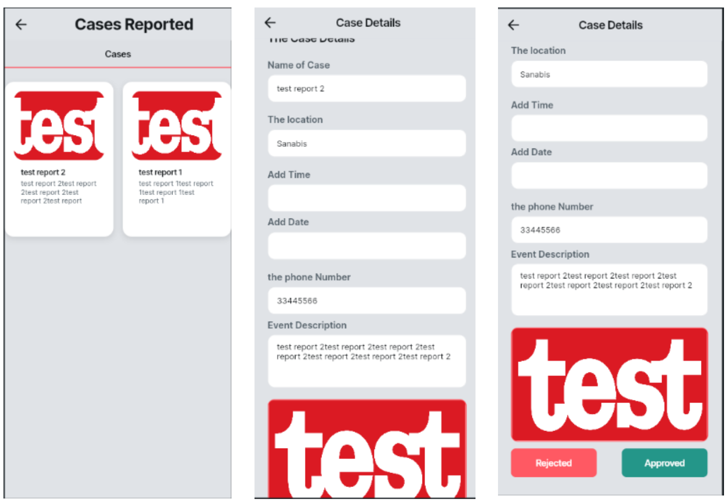
Sponsor Name: Enter your company or organiz...
Email: Enter your email
Phone Number: Enter your phone

Cancel | Submit

Figure 18: Sponsor Case/Event Support

The sponsor home page has two main tabs, which include Cases and Events. Sponsors will be able to use either tab to see existing cases or events and support them by filling out a special form. This form gathers crucial information of supporters such as name, email address and phone number as shown in Figure 18: Sponsor Case/Event Support.

5.5.6 Admin Approving Cases


Cases Reported

Cases

test report 2
test report 2test report 2test report 2test report 2test report

test report 1
test report 1test report 1test report 1test report 1

Case Details

Name of Case: test report 2
The location: Sanabis
Add Time:
Add Date:
the phone Number: 33445566
Event Description: test report 2test report 2test report 2test report 2test report 2test report 2test report 2

Case Details

The location: Sanabis
Add Time:
Add Date:
the phone Number: 33445566
Event Description: test report 2test report 2test report 2test report 2test report 2test report 2test report 2

Rejected | Approved

Figure 19: Admin Approving Cases

As illustrated in Figure 19: Admin Approving Cases, the administrator has the capability to view cases submitted by volunteers. These cases can then be considered by the administrator and some actions of approval, rejection, or modification can be undertaken. Upon approval and proper refinement, the cases are publicized under the section entitled the public cases where they become accessible to both the volunteers and sponsors to get involved in and support them.

5.5.7 Admin Adding Events/Cases

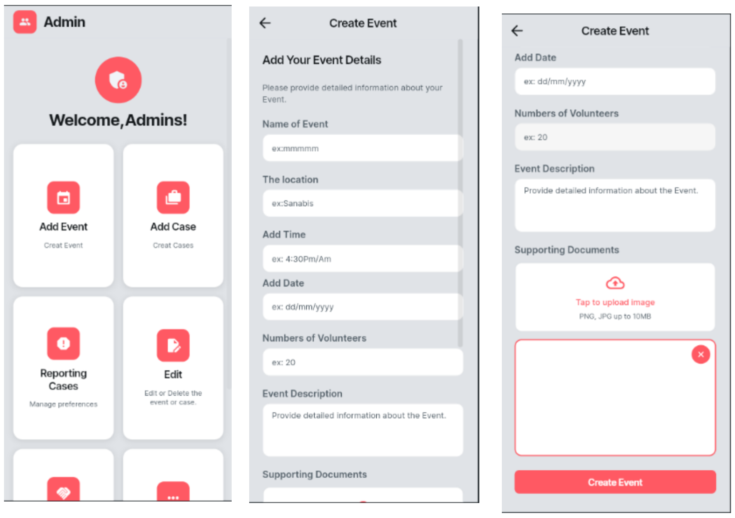
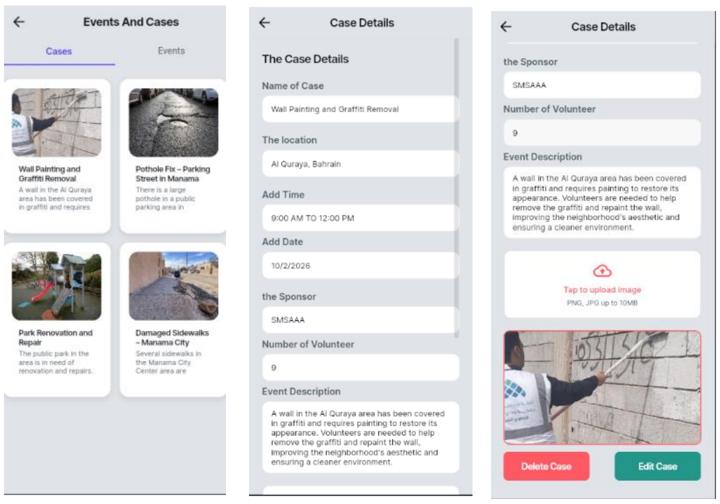


Figure 20: Admin Adding Events/Cases

As depicted in Figure 20: Admin Adding Events/Cases, the administrator can directly add cases or events to the application by completing their respective forms.

5.5.8 Admin Edit/Delete Events/Cases



Wall Painting and Graffiti Removal
A wall in the Al Quraya area has been covered in graffiti and requires

Pothole Fix – Parking Street in Manama
There is a large pothole in a public parking area in

Park Renovation and Repair
The public park in the area is in need of renovation and repairs.

Damaged Sidewalks – Manama City
Several sidewalks in the Manama City Center area are

The Case Details

Name of Case: Wall Painting and Graffiti Removal

The location: Al Quraya, Bahrain

Add Time: 9:00 AM TO 12:00 PM

Add Date: 10/2/2026

the Sponsor: SMSAAA

Number of Volunteer: 9

Event Description: A wall in the Al Quraya area has been covered in graffiti and requires painting to restore its appearance. Volunteers are needed to help remove the graffiti and repaint the wall, improving the neighborhood's aesthetic and ensuring a cleaner environment.

the Sponsor: SMSAAA

Number of Volunteer: 9

Event Description: A wall in the Al Quraya area has been covered in graffiti and requires painting to restore its appearance. Volunteers are needed to help remove the graffiti and repaint the wall, improving the neighborhood's aesthetic and ensuring a cleaner environment.

Figure 21: Admin Edit/Delete Events/Cases

As illustrated in Figure 21: Admin Edit/Delete Events/Cases, administrators can edit or delete any cases or events previously uploaded to the application.

5.6 System Workflow Pseudocode

This section presents the core system logic for the "We Are The Change" platform pseudocode mentioned in Appendix B System Workflow Pseudocode, a structured form of algorithmic description that communicates processes in a simple, language-neutral style. Pseudocode is used here to outline the main workflows and control flows of the application, making the logic easily understandable without focusing on any specific programming syntax. Each block of pseudocode describes an essential feature—such as user registration and login, case reporting, volunteer and sponsor interaction, and administrative functions—and breaks down the step-by-step operations required for system functionality. By modeling these algorithms in pseudocode, this provides clear guidance for later development, testing, and analysis, ensuring that every process is logically organized and easy to translate into implementation code.

5.6.1 User Registration and Login

The first part of the pseudocode describes the logic for handling user registration and login in the application. The flow begins when the user chooses either the "Register" or "Login" option from the interface.

Registration flow:
When the user selects "Register", the system prompts them to enter an email, a password, and any required personal information. These values are treated as input fields. The application then performs input validation, which means checking that the email format is correct, the password meets security rules (for example, minimum length), and mandatory fields are not empty. If all inputs are valid, a new user account is created using Firebase Authentication, which is a cloud-based service that securely manages user identities. After successful account creation, the user is redirected to the Login page so they can authenticate and access the system. If validation fails, an error message is displayed to inform the user that the registration data is invalid.

Login flow with OTP:

When the user selects “Login”, the system prompts for a username (or email) and password. These credentials are sent to Firebase Authentication for verification. If the credentials are valid, the system then sends a one-time password (OTP) to the user’s email as a second authentication factor. The OTP is a temporary code that adds an extra security layer. The user must provide this OTP, and the system verifies that both the original credentials and the OTP are correct. If both checks pass, the user is redirected to their dashboard, which is the main area of the application. If either the credentials or the OTP are incorrect, an error message is shown. Overall, this algorithm ensures secure access by combining standard email/password login with an additional OTP verification step.

5.6.2 Case Upload (Issue Reporting)

This pseudocode represents the process for allowing users to submit new issue reports, called “cases”, into the system.

Input collection and validation:

The flow starts when the user chooses “Upload Case”. The system then asks for several details: a title for the case, a description of the problem, a phone number for contact, the location of the issue, and at least one photo. These inputs describe the problem and provide enough information for administrators and volunteers to understand and locate it. After collecting the data, the system validates the inputs. Typical validation checks include ensuring that the phone number is in a valid format, and a location and photo are provided.

Data storage and feedback:

If validation succeeds, the case is stored in the database. This means a new record is created that contains all the entered information and links it to the reporting user. The system then notifies an administrator, via a dashboard update, so that the case can be reviewed and processed. Finally, a success message is displayed to confirm that the case was uploaded correctly. If validation fails at any stage, an error message is shown instead, prompting the user to correct the input. This algorithm ensures that all reported issues are complete, consistent, and immediately visible to admins.

5.6.3 Volunteer Browsing and Sign-Up

This segment describes how users browse available volunteer opportunities and register themselves for a specific task.

Browsing flow:
The process starts when the user selects “Browse Volunteer Cases/Events”. In response, the system retrieves and displays a list of active cases or events that require volunteers. This list typically includes basic information such as the case title, date, and location.

Sign-up logic:
If the user decides to help with a particular case or event and chooses “Volunteer”, the system adds the user to the volunteer list for that selected task. This usually means creating an association in the database between the user’s account and the chosen case/event. A confirmation message is then displayed to inform the user that the sign-up was successful. If the user does not select “Volunteer”, the algorithm simply returns to the list of tasks without making any changes. This flow makes it easy for volunteers to discover available opportunities and register with a single action.

5.6.4 Sponsor Browsing and Support Request

This pseudocode models how potential sponsors view cases and indicate their interest in supporting them.

Browsing flow:
The sponsor-related process begins when the user selects “Browse Cases/Events”. Similar to the volunteer flow, the system displays a list of available cases or events that may need financial or logistical support.

Support request handling:

If the user selects “Support” for a particular case, the system displays a message informing the user that administrators will contact them. At the same time, the system then notifies an administrator, via a dashboard update, so they know a sponsor is interested. This could trigger a follow-up process outside the app (e.g., email or phone call). If the user does not choose “Support”, the algorithm returns to the list view without storing any new data. This logic provides a simple way for sponsors to express interest without exposing complex financial workflows in the pseudocode.

5.6.5 Admin Dashboard Functions

The final pseudocode block describes the main view and capabilities available to administrators after they log in.

Admin access and overview:

After a successful admin login, the system displays the Admin Dashboard. This dashboard acts as a control panel for managing platform activity. It shows a list of cases uploaded by volunteers or citizens, and a list of support requests submitted by sponsors. These lists allow admins to quickly review which issues are pending and which sponsorship offers are waiting.

Management actions:

In addition to viewing data, the dashboard provides options to edit or delete existing cases and events, as well as create new ones. Editing allows admins to correct information or update statuses, while deletion is used to remove invalid, duplicate, or completed cases/events from the system. Creating new cases or events enables admins to proactively post opportunities for volunteering or sponsorship. Although the pseudocode ends after displaying these options, the implication is that each button on the dashboard triggers further logic (e.g., opening forms, saving changes) defined elsewhere. Overall, this block shows that administrators have centralized control over the core entities of the platform.

5.6.6 Summary of How the Algorithms Work Together

Across all five pseudocode sections, the logic defines a complete workflow for the civic engagement platform. User registration and login control secure access. Case upload allows users to report issues with structured data. Volunteer browsing and sponsor browsing create participation pathways for community members and supporting organizations. The admin dashboard provides oversight and moderation to keep the system organized and trustworthy. Each step—input collection, validation, database operations, and feedback messages—is necessary to maintain data quality, security, and a smooth user experience from both the citizen and administrator perspectives.

Chapter 6
System Implementation and Testing

6.1 Introduction

6.2 System implementation

This part explains the process of implementing the proposed system as a working prototype and the reasoning behind the choice of technologies, component integration, and the general working process flow of user interaction to backend processing. Its implementation was focused on achieving a balance between rapid development, scalability and maintainability and being practical within the limits of an undergraduate project.

6.2.1 Technology Stack and Rationale

The FlutterFlow platform was used to implement the mobile application, and it is constructed on the top of the Flutter platform. FlutterFlow was chosen due to the ability to create cross-platform interfaces (Android and iOS) fast but generate clean Flutter code that can be extended as required. Its drag-and-drop interface, constructed in terms of widgets, and its connection to the backend services minimized the number of boilerplate codes and enabled the team to concentrate on the system logic and user experience over the implementation of low-level user interfaces. The fact that the same codebase was used across various platforms also made testing and deployment easier. Firebase was embraced as the backend solution offering authentication as well as the cloud-hosted database. The user accounts were securely managed with Firebase Authentication, which included email/password log-in and has potential functionality to extend to other identity providers in the future. The database layer was realized with the help of Cloud Firestore that provides a scalable and document-oriented data structure that can be used to store users, cases, and events. The close support between FlutterFlow and Firebase reduced configuration cost and gave real-time data synchronization, which is vital in maintaining volunteer activity, and sponsor activity across devices.

6.2.2 System Architecture and Component Integration

The overall architecture follows a client–cloud model. FlutterFlow-generated mobile application forms the client side and provides the implementation of the presentation layer and simple client-side validation. The cloud side is made up of Firebase services, which store data, perform authentication, and server-side logic. Further backend functionality, including validation rules may be captured in Firebase Cloud Functions and security rules. This isolation of concerns provides that sensitive operations and access control policy may be imposed on the server side and the client is kept lightweight and responsive. Firebase SDKs that are configured in FlutterFlow are used to integrate between components. A collection and documents in the database are attached to each screen in the application. An example is a cases collection where reports given by users are stored, each document containing some attributes like description, location, and references to associated tasks or sponsorship records. The action system of FlutterFlow is intended to perform database operations (create, update, delete) and authentication operations (sign-in, sign-up, sign-out) directly on the UI elements. It permits coherent definition of user interface, flow of navigation and backend operations within the same environment, eliminating integration errors and making debugging easier.

6.2.3 Implementation Decisions and Justification

The FlutterFlow selection was the rational decision based on the time and resource limitations of student works. There would be much more effort and platform-specific knowledge to implement the same functionality manually in native Android and iOS. Through FlutterFlow, the team was able to rapidly iterate to make design changes, collect feedback and refine user flows without having to rewrite significant amounts of code. Meanwhile, since FlutterFlow produces standard Flutter code, the system is expandable: more complex logic or custom widgets may be added in the future in case the project is not yet at the prototype stage. Firebase was chosen instead of conventional server hosting because of a number of reasons. To start with, its pay-as-you-go and ample free plan can be appropriate with academic prototypes with low budgets. Second, Firebase offers curated infrastructure, meaning that there is no need to manualize or maintain servers, databases, and authentication systems. Third, it has real-time synchronization features which are especially useful in this kind of system, as the case statuses can change often, either because of the actions of the administration, or the activities of the volunteers, or because of the will of the sponsors. Role-based access is also easy to enforce through the Firebase security rules, like limitation on specific operations to an account of admins or sponsors.

6.2.4 System Flow and Screen Descriptions

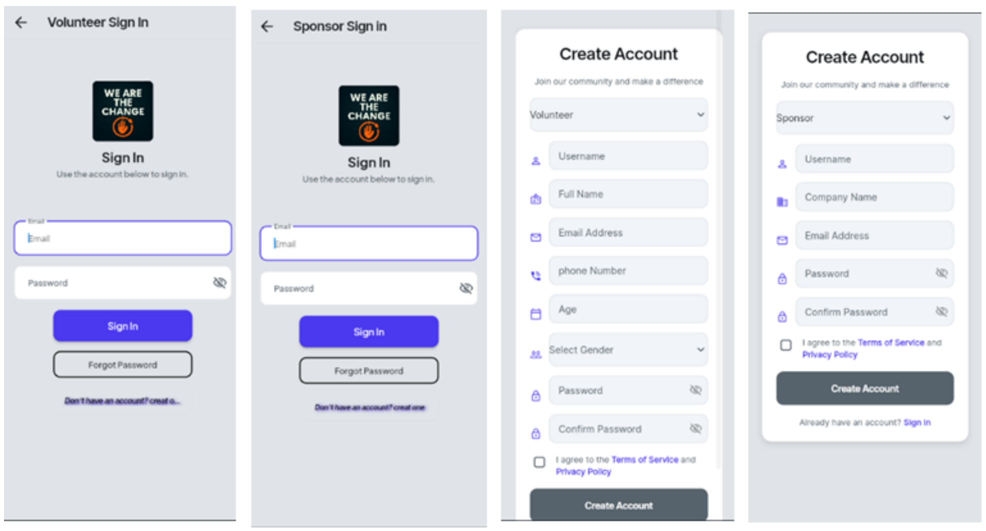


Figure 22: User Sign up and login

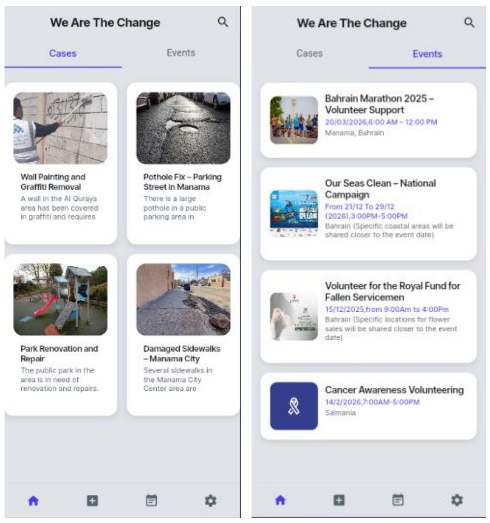
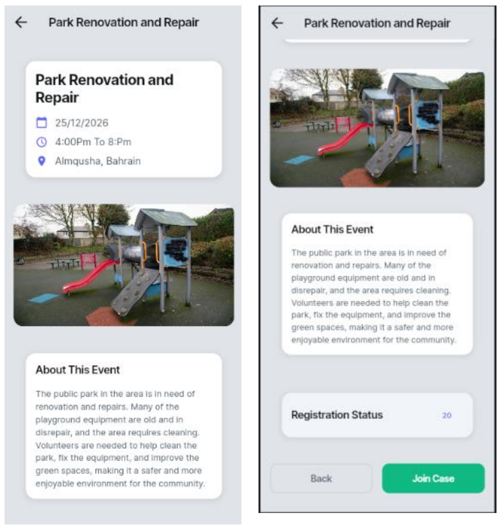
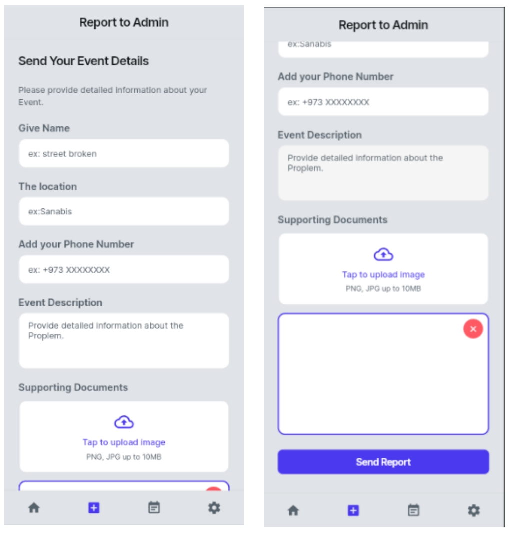

We Are The Change
Cases
Events
Wall Painting and Graffiti Removal
A wall in the Al Quraya area has been covered in graffiti and requires
Pothole Fix – Parking Street in Manama
There is a large pothole in a public parking area in
Park Renovation and Repair
The public park in the area is in need of renovation and repairs.
Damaged Sidewalks – Manama City
Several sidewalks in the Manama City Center area are
We Are The Change
Cases
Events
Bahrain Marathon 2025 – Volunteer Support
20/03/2026,6:00 AM – 12:00 PM
Manama, Bahrain
Our Seas Clean – National Campaign
From 21/12 To 29/12 (2026),3:00PM-5:00PM
Bahrain (Specific coastal areas will be shared closer to the event date)
Volunteer for the Royal Fund for Fallen Servicemen
15/12/2025,from 9:00Am to 4:00Pm
Bahrain (Specific locations for flower sales will be shared closer to the event date)
Cancer Awareness Volunteering
14/2/2026,7:00AM-5:00PM
Salmania

Figure 23: Volunteer home page



The public park in the area is in need of renovation and repairs. Many of the playground equipment are old and in disrepair, and the area requires cleaning. Volunteers are needed to help clean the park, fix the equipment, and improve the green spaces, making it a safer and more enjoyable environment for the community.

Figure 24: Show case/event


Report to Admin

Send Your Event Details

Please provide detailed information about your Event.

Give Name

ex: street broken

The location

ex:Sanabis

Add your Phone Number

ex: +973 XXXXXXXX

Event Description

Provide detailed information about the Proplem.

Supporting Documents

Tap to upload image

PNG, JPG up to 10MB

Report to Admin

ex:Sanabis

Add your Phone Number

ex: +973 XXXXXXXX

Event Description

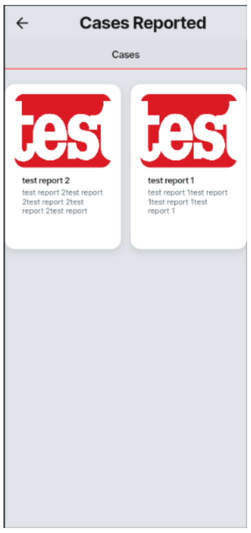
Provide detailed information about the Proplem.

Supporting Documents

Tap to upload image

PNG, JPG up to 10MB

Figure 25: Upload case



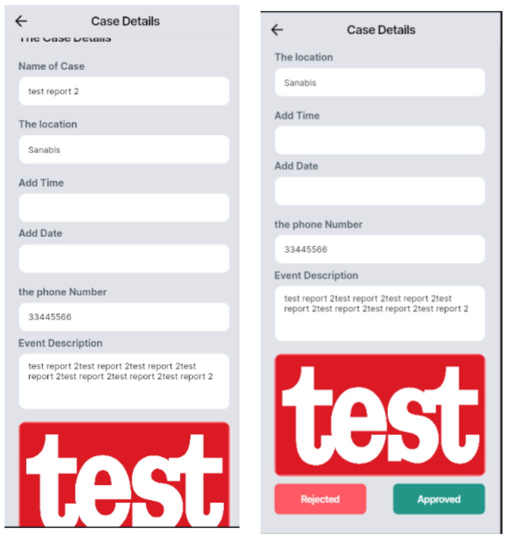
test report 2

test report 2test report 2test report 2test report 2test report

test report 1

test report 1test report 1test report 1test report 1

Figure 26: Admin show cases page


Case Details

The Case Details

Name of Case

test report 2

The location

Sanabis

Add Time

Add Date

the phone Number

33445566

Event Description

test report 2test report 2test report 2test report 2test report 2test report 2test report 2test report 2

Case Details

The location

Sanabis

Add Time

Add Date

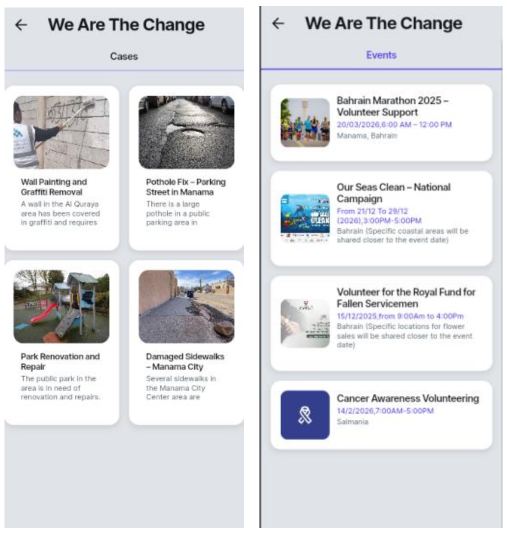
the phone Number

33445566

Event Description

test report 2test report 2test report 2test report 2test report 2test report 2test report 2test report 2

Figure 27: Admin edit/delete cases



We Are The Change

Cases

Wall Painting and Graffiti Removal
A wall in the Al Quraya area has been covered in graffiti and requires

Pothole Fix – Parking Street in Manama
There is a large pothole in a public parking area in

Park Renovation and Repair
The public park in the area is in need of renovation and repairs.

Damaged Sidewalks – Manama City
Several sidewalks in the Manama City Center area are

We Are The Change

Events

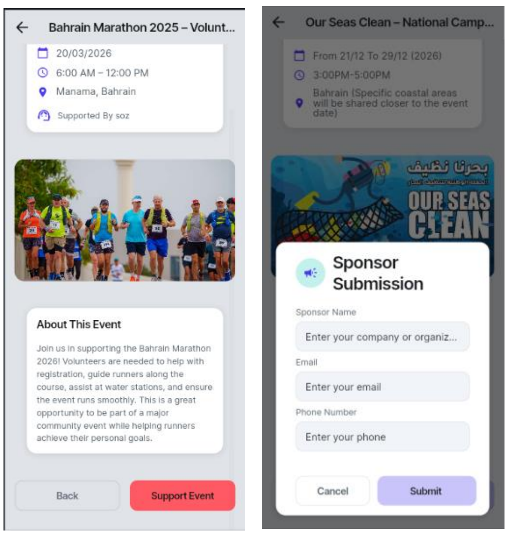
Bahrain Marathon 2025 – Volunteer Support
20/03/2026,6:00 AM – 12:00 PM
Manama, Bahrain

Our Seas Clean – National Campaign
From 21/12 To 29/12 (2026),3:00PM-5:00PM
Bahrain (Specific coastal areas will be shared closer to the event date)

Volunteer for the Royal Fund for Fallen Servicemen
15/12/2025,from 9:00Am to 4:00Pm
Bahrain (Specific locations for flower sales will be shared closer to the event date)

Cancer Awareness Volunteering
14/2/2026,7:00AM-5:00PM
Salmania

Figure 28: Sponsor view of cases/events



Bahrain Marathon 2025 – Volunt...

20/03/2026
6:00 AM – 12:00 PM
Manama, Bahrain
Supported By soz

About This Event

Join us in supporting the Bahrain Marathon 2026! Volunteers are needed to help with registration, guide runners along the course, assist at water stations, and ensure the event runs smoothly. This is a great opportunity to be part of a major community event while helping runners achieve their personal goals.

Back | Support Event

Our Seas Clean – National Camp...

From 21/12 To 29/12 (2026)
3:00PM-5:00PM
Bahrain (Specific coastal areas will be shared closer to the event date)

Sponsor Submission

Sponsor Name: Enter your company or organiz...
Email: Enter your email
Phone Number: Enter your phone

Cancel | Submit

Figure 29: Sponsor show event/case page

The flow of the system begins with the authentication of the user. A registration and a login screen would be displayed in Figure 22: User Sign up and login and would be developed with the authentication widgets in FlutterFlow that were linked to Firebase Authentication. The screen enables one to create an account or log in, and the basic input validation (e.g., email format, password length) is performed. Upon verification, the user is redirected to the respective dashboard according to his/her role (e.g., admin, volunteer, sponsor).

The key actions and a list of the cases of interest would be presented in Figure 23: Volunteer home page as the main dashboard of the volunteers. When a case is tapped, a detailed view screen Figure 24: Show case/event is opened and the user is able to view the full description, the information attached as well as the current status of the submission enabling him or her to have an idea on the progress of his/her submission.

In the case of the reporting functionality, the Figure 25: Upload case would display a form that said Submit Case. This type has such fields as description of the problem, location, and attachments (which are optional). The Submit button causes an operation of create in the Firebase database, which creates a new case document associated with the ID of the user and initial value of the document as submitted. FlutterFlow form logic takes care of any required client-side validation, with other restrictions (e.g., required fields, role restrictions) being ultimately implemented by Firebase rules or backend functions.

On the administrative front, there would be another dashboard Figure 26: Admin show cases page with the interface of Review Cases, which is available to only the administrators. All the submitted cases are listed on this screen. When a case is selected, another screen opens giving a chance to view the details and take such actions as Approve, Decline, or Edit. Such activities update the status column of the case document as shown in Figure 27: Admin edit/delete cases. As changes are made, they are automatically reflected to other stakeholders (e.g., the original reporter, volunteers or sponsors) due to the real time synchronization between Firebase and the client.

For sponsor functionality, Figure 28: Sponsor view of cases/events would illustrate a Review Cases To Support screen which would retrieve cases that are tagged as eligible or requiring support. The sponsor is able to pick a case and scroll to a screen named Support Case whereby the sponsor types the name of the sponsor and contact information in Figure 29: Sponsor show event/case page. The filing of this form generates or adds a support record to the case. Lastly, an administration screen would display all pending support entries where the administration can accept or reject the sponsorship. Acceptance of the support not only updates the status of the case but also the support document involved, and completes the process of sponsor action to administrative confirmation.

To sum, the system flow is as follows: users log in via the frontend of FlutterFlow linked to Firebase Authentication; users make or perform operations on cases via the UI; FlutterFlow converts these commands to database actions on Firebase; and real-time updates are propagated back to all the connecting clients. FlutterFlow and Firebase, when combined therefore allows one to come up with a fully working prototype that illustrates the main interactions of the proposed system and can be viable and maintainable in the context of undergraduate research.

6.3 Evaluation and testing

6.3.1 Introduction

This chapter describes the testing and evaluation process for the "We Are The Change" mobile application. Once the main features were implemented using FlutterFlow and Firebase, a systematic testing was performed in order to check the functionality, and security. The testing process was done in normal stages such as unit testing, integration testing, functional testing, and security testing. These testing sessions confirmed the system was satisfactory per the Chapter 4 functional and security requirements and was dependable to the community users. The whole testing was done on actual devices and emulators to replicate the real usage conditions (Mobile Application Testing Process Explained, n.d.).

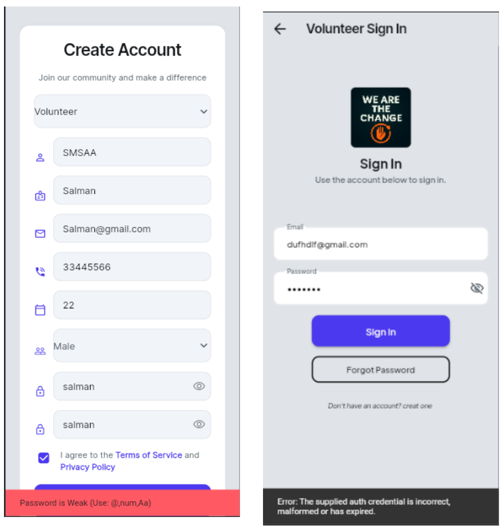
6.3.2 Testing Phases Overview

The testing process was divided into two main phases: functional testing, and security testing. Functional testing ensured that all primary functionalities were functioning as was required in the requirements. This involved user registration, login, case submission, sign-ups as a volunteer, requesting sponsor support, and operations on the dashboard of the administrator. Each user role (volunteer, sponsor, admin) was developed into test cases and run manually in Android emulators. Positive and negative conditions were checked, including valid/ invalid login credentials. No critical defects in all tests of functions (Critical Mobile App Testing Scenarios Every QA Team Should Use, n.d.).

Security Testing aimed at securing user information and avoiding popular attacks. Simple penetration testing tools were used, such as SQL injection, cross-site scripting (XSS), and insecure direct object reference. The Firebase Emulator Suite was used to validate firebase security rules. The tests ensured different role-based access control and input validation was done to block unauthorized access and data manipulation (Patrich, n.d.).

6.3.3 Functional Testing Results

Functional testing covered all major features from the use cases in Chapter 4. Test cases were executed on Android (ZTE Z2472).


Create Account

Join our community and make a difference

Volunteer

SMSAA

Salman

Salman@gmail.com

33445566

22

Male

salman

salman

✓ I agree to the Terms of Service and Privacy Policy

Password is Weak (Use: @,num,Aa)

Volunteer Sign In

Sign In

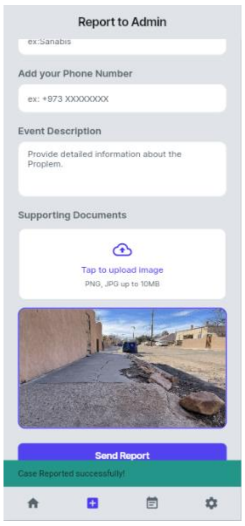
Use the account below to sign in.

Email: dufhdlf@gmail.com

Password: •••••••

Error: The supplied auth credential is incorrect, malformed or has expired.

Figure 30: Invalid inputs error


Report to Admin

ex:Sanabis

Add your Phone Number

ex: +973 XXXXXXXX

Event Description

Provide detailed information about the Proplem.

Supporting Documents

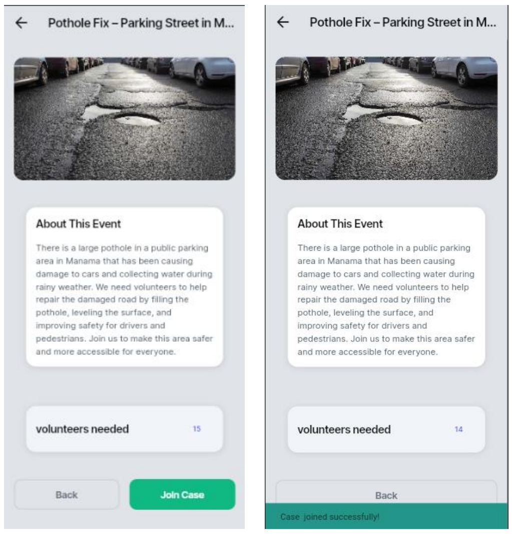
Tap to upload image

PNG, JPG up to 10MB

Send Report

Case Reported successfully!

Figure 31: Reporting cases


Pothole Fix – Parking Street in M...

About This Event

There is a large pothole in a public parking area in Manama that has been causing damage to cars and collecting water during rainy weather. We need volunteers to help repair the damaged road by filling the pothole, leveling the surface, and improving safety for drivers and pedestrians. Join us to make this area safer and more accessible for everyone.

volunteers needed	15

Back | Join Case

Pothole Fix – Parking Street in M...

About This Event

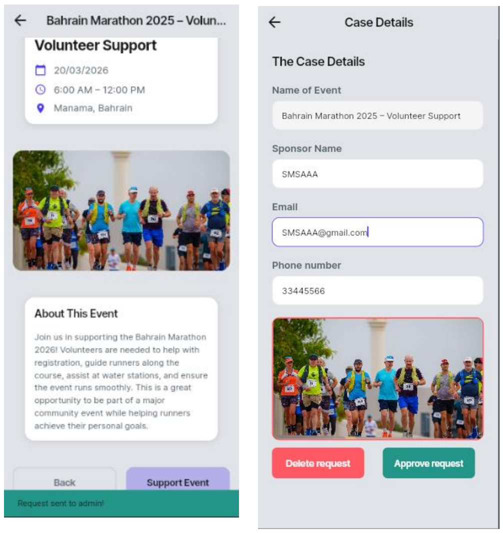
There is a large pothole in a public parking area in Manama that has been causing damage to cars and collecting water during rainy weather. We need volunteers to help repair the damaged road by filling the pothole, leveling the surface, and improving safety for drivers and pedestrians. Join us to make this area safer and more accessible for everyone.

volunteers needed	14

Back

Case joined successfully!

Figure 33: Joining cases


Bahrain Marathon 2025 – Volun...

Volunteer Support

20/03/2026

6:00 AM – 12:00 PM

Manama, Bahrain

About This Event

Join us in supporting the Bahrain Marathon 2026! Volunteers are needed to help with registration, guide runners along the course, assist at water stations, and ensure the event runs smoothly. This is a great opportunity to be part of a major community event while helping runners achieve their personal goals.

Back | Support Event

Request sent to admin!

Case Details

The Case Details

Name of Event

Bahrain Marathon 2025 – Volunteer Support

Sponsor Name

SMSAAA

Email

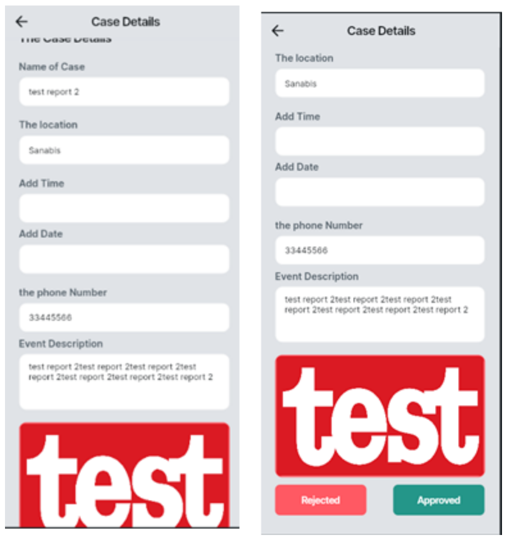
SMSAAA@gmail.com

Phone number

33445566

Delete request | Approve request

Figure 32: Sponsor request support


Case Details

Name of Case
test report 2

The location
Sanabis

Add Time

Add Date

the phone Number
33445566

Event Description
test report 2test report 2test report 2test report 2test report 2test report 2test report 2

Case Details

The location
Sanabis

Add Time

Add Date

the phone Number
33445566

Event Description
test report 2test report 2test report 2test report 2test report 2test report 2test report 2

Figure 34: Admin approve/reject/edit/delete operations

- User Registration/Login: test cases passed. Valid credentials redirected to dashboards; invalid inputs showed clear error messages according to Figure 30: Invalid inputs error.
- Case Upload: test cases passed. Forms with photos saved correctly according to Figure 31: Reporting cases.
- Volunteer Sign-up: test cases passed. Users successfully joined tasks; duplicate sign-ups prevented according to Figure 33: Joining cases.
- Volunteer Sign-up: test cases passed. Users successfully joined tasks; duplicate sign-ups prevented according to Figure 32: Sponsor request support.
- Volunteer Sign-up: test cases passed. Users successfully joined tasks; duplicate sign-ups prevented according to Figure 34: Admin approve/reject/edit/delete operations.

6.3.4 Security Testing

Security testing used manual penetration techniques and Firebase's Security Rules Simulator. Common vulnerabilities from ethical hacking courses were tested, including SQL/NoSQL injection, XSS, and authentication bypass. All tests passed due to Firebase's built-in protections and input sanitization.

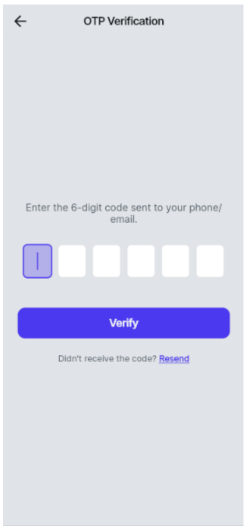
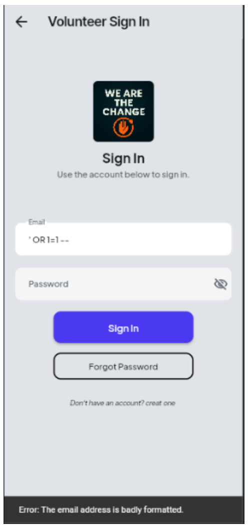

OTP Verification
Enter the 6-digit code sent to your phone/ email.
Verify
Didn't receive the code? Resend

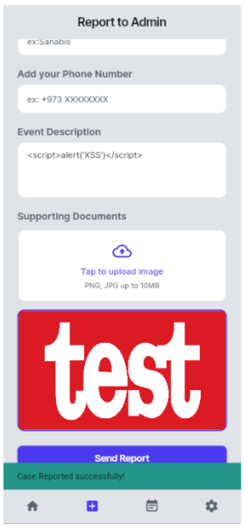
Figure 35: MFA authentication



' OR 1=1 --

Error: The email address is badly formatted.

Figure 36: SQL/NoSQL Injection



Report to Admin

ex:Sanabis

Add your Phone Number

ex: +973 XXXXXXXX

Event Description

<script>alert('XSS')</script>

Supporting Documents

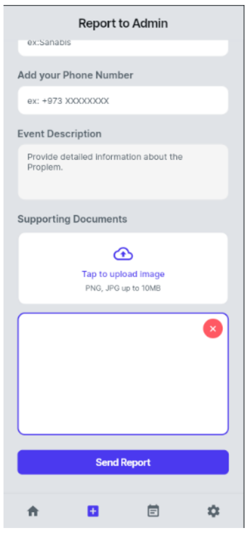
Tap to upload image

PNG, JPG up to 10MB

Send Report

Case Reported successfully!

Figure 37: Cross-Site Scripting (XSS)



Report to Admin

ex:Sanabis

Add your Phone Number

ex: +973 XXXXXXXX

Event Description

Provide detailed information about the Proplem.

Supporting Documents

Tap to upload image

PNG, JPG up to 10MB

Send Report

Figure 38: File type validation

Testing Criteria	**Target**	**Result**	**Explanation**
SQL/NoSQL Injection	All input forms (login, case title, description)	Pass	Malicious payloads like ' OR 1=1 -- rejected by Firebase input validation and parameterized queries. No database access granted as shown in Figure 36: SQL/NoSQL Injection.
Cross-Site Scripting (XSS)	Case description, comments	Pass	<script>alert('XSS')</script> sanitized and stored as plain text. No JavaScript execution occurred according to Figure 37: Cross-Site Scripting (XSS).
Authentication Bypass	Login form, API endpoints	Pass	Invalid tokens were rejected. Firebase custom claims enforced role-based access correctly.
Insecure Direct Object Reference (IDOR)	Case editing, volunteer lists	Pass	Users could only access their own cases/tasks. Firebase security rules blocked unauthorized paths.
File Upload Vulnerability	Photo uploads	Pass	Only image MIME types are allowed (JPEG, PNG). Executable files rejected. File size limited to 5MB according to Figure 38: File type validation.
Session Hijacking	Active sessions	Pass	Tokens expired after 1 hour. No persistent sessions without re-authentication.
Role Escalation	User to Admin dashboard	Pass	Regular users cannot redirect to admin panel. Firebase roles strictly enforced.
Multi-Factor Authentication (MFA)	Admin, sponsor, volunteer login.	Pass	Email OTP required after password entry. Invalid codes rejected as shown in Figure 35: MFA authentication.

Table 28: Security requirements testing

6.3.6 Results Discussion

The test results validate the fact that we are the change is successfully achieving its functional and security requirements. The functional testing had a 100 percent success rate in all user roles, just as seen with commercial civic applications such as (SeeClickFix, 2022) discussed in Chapter 2. Security analysis prevented the standard attacks with the help of the robust Firebase security, better than unstructured social media tactics (e.g., Bu Jarrah) that are not designed around security.

6.3.7 Conclusion

The testing stages showed that the introduced system is effective and safe. The entire core features are reliable on all user roles and the security measures prevent typical threats. The findings are consistent with the design options of Chapter 5 and the fact that the platform is ready to interact in the real world of civic participation (Mobile Application Testing Process Explained, n.d.).

Chapter 7
Conclusion and Future Work

7.1 Conclusion

This project successfully designed, implemented, and tested the "We Are The Change" civic engagement platform that fills the gap in the existing systems in Bahrain in terms of reporting the community issues and volunteer recruitment. The system design, based on the requirement analysis, conducted as interviews and SQUARE methodology (Chapter 4), produced complete ERDs schema, UI prototypes and STRIDE threat models (Chapter 5). It was implemented with FlutterFlow on the frontend and Firebase on the secure backend service, providing all the essential functionality: user creation, case creation with photos, Volunteer sign-up, sponsor support request creation, and management of the dashboard by the administrator.

System reliability was tested and proved to be reliable at every stage. User roles and workflows had 100% pass in functional tests. Firebase protections and implementation of MFA prevented the common threats of SQL injection, XSS, and API abuse with security testing. The platform showed that it was ready to be used in the real world, bridging the gap of volunteer integration that was found in the literature review of Chapter 2 in relation to systems such as Tawasul and SeeClickFix.

The importance of these findings is in establishing the first integrated civic platform in Bahrain, which allows mixing orderly reporting and action in the community. The system allows active involvement of citizens by allowing them to not only diagnose issues but also volunteer remedies to the problems, thereby minimizing the effectiveness of government solutions. This fulfills the project goals of enhancing local issue-solving using technology and ensuring high security to the users (Write conclusion, n.d.).

7.2 Limitations

Although the project was successfully implemented, it had realistic constraints that were characteristic of work by students. Lack of funding also meant that they could not afford more advanced features such as native mobile app development (not even FlutterFlow no-code) and professional server infrastructure. Free tier on firebase had a limit on concurrent users and storage which may impact scalability in times of peak traffic of the community.

The coverage of manual testing would be extensive, though not complete; automated test suites would be more helpful in regression testing. The testing on stakeholders was restricted to the interview with RCSI Bahrain and lacked general community validation. These restrictions had no impact on core functionality but only the feature depth (Conclusion thesis, n.d.).

7.3 Future Work

The platform can be extended with a number of practical improvements:

- Push Notifications: Implement the use of Firebase Cloud Messaging to send real-time notifications on updates in the status of cases, position of volunteers, and reminders to users in case of events to enhance user interaction.
- Geolocation Tagging: This is an extension of the case report that will have automatic GPS location and display a map, which can be used to assign cases to specific locations and track their progress.
- Arabic Language Support: Provide bilingual UI to make it more accessible to the key users of the language in Bahrain.
- Advanced Analytics: Add admin dashboards showing case resolution rates, volunteer participation trends, and geographic hotspots using Firebase Analytics.
- Load Testing & Scaling: Conduct stress tests for 1000+ concurrent users and upgrade to Firebase paid tiers with CDN for media delivery.
- Integration with Government Systems: API to Tawasul integration Develop APIs to automate escalating critical cases.

This would turn the prototype into a production-ready civic platform fit to be adopted by municipalities, and overcome existing limitations by adding to proven core functionality (How to write an excellent thesis conclusion [with examples], n.d.).

References

AL-Kaabi, R. (2024). Factors Affecting Citizen Engagement in the Kingdom of.

AlOthman, A. (n.d.). Retrieved from https://www.alqabas.com/article/5932698-%D8%A8%D9%88%D8%AC%D8%B1%D8%A7%D8%AD-%D9%8A%D8%BA%D9%8A%D8%B1/

Authority, I. &. (n.d.). Retrieved from https://www.bahrain.bh/tawasul

Balady. (n.d.). *balady about*. Retrieved from balady services: https://balady.gov.sa/en/about-balady

Balady. (n.d.). *submitting a report*. Retrieved from balady services: https://balady.gov.sa/en/services/submitting-report

Buchanan, E. M., Crain, S. E., Cunningham, Johnson, Stash, Papadatou-Pastou, . . . Aczel, B. (2021). Getting started creating data dictionaries: How to create a shareable data set. *Advances in Methods and Practices in Psychological Science, 4*(1). doi:10.1177/2515245920928007

Conclusion thesis. (n.d.). Retrieved from topscriptie: https://www.topscriptie.nl/en/conclusion-thesis-3/

Critical Mobile App Testing Scenarios Every QA Team Should Use. (n.d.). Retrieved from testevolve: https://www.testevolve.com/blog/critical-mobile-app-testing-scenarios-every-qa-team-should-use

Digital Government Strategy Principles. (n.d.). Retrieved from bahrain: https://bahrain.bh/wps/portal/en/BNP/HomeNationalPortal/ContentDetailsPage/!ut/p/z1/vVRNc5swEP0rysFHRgIJGY64dvxRu3bskMRcMgJkRwkIAjKt--sr0uk0hhonl-rCoH1Pu1q9tzCADzCQrBJ7pkQmWaL_twF9NKc2mjhfEFoNiYVu7tw1HSxMtBhjeH8KmHxbDRFdTpy-1R9g4lkwOAnf-A6i1ny2pNYAL0cmvIM

eParticipation Tools. (n.d.). Retrieved from bahrain: https://bh.bh/new/en/eparticipation-govaffairs_en.html

Firebase security checklist. (n.d.). Retrieved from Firebase: https://firebase.google.com/support/guides/security-checklist *flutter_performance_testing*. (n.d.). Retrieved from pub: https://pub.dev/documentation/flutter_performance_testing/latest/

Fung, A. (2006). Varieties of participation in complex governance. *Public Administration Review, 66*(s1), 66. doi:https://faculty.fiu.edu/~revellk/pad3003/Fung.pdf

How to write an excellent thesis conclusion [with examples]. (n.d.). Retrieved from paperpile: https://paperpile.com/g/thesis-conclusion/

Liyanage, A. (2025, May 4). *medium*. Retrieved from https://medium.com/@nld.anuradha/the-agile-model-in-sdlc-a-flexible-approach-to-software-development-664bf60ad9dc

Mead, N. R., Hough, E. D., & Stehney, T. R. (2005). *Security Quality Requirements Engineering (SQUARE) Methodology.* Carnegie Mellon University.

Mobile Application Testing Process Explained. (n.d.). Retrieved from testilo: https://testlio.com/blog/step-step-mobile-application-testing-process/

MySociety. (n.d.). Retrieved from https://www.fixmystreet.com

Patrich. (n.d.). *Guide To Building Secure Backends In Firebase In 2024*. Retrieved from slashdev: https://slashdev.io/-guide-to-building-secure-backends-in-firebase-in-2024

SeeClickFix. (2022). Retrieved from https://www.seeclickfix.com

Uschold, M. (2015). Ontology and database schema: What's the difference? *Applied Ontology, 10*, 243–258. doi:10.3233/AO-150158

Write conclusion. (n.d.). Retrieved from Scribbr: https://www.scribbr.com/dissertation/write-conclusion/

Appendix A
Requirements Collection Meeting Questionnaire

Section 1: Current Processes and Pain Points

1. Can you describe the current process for reporting issues such as damaged infrastructure, unclean areas, or general maintenance needs on campus?
2. Who typically reports these problems—students, staff, security personnel?
3. What channels are used for reporting (email, forms, verbal communication, etc.)?
4. What challenges do you face in receiving, tracking, or resolving these reports?
5. How long does it typically take for an issue to be resolved once it's reported?

Section 2: Perception of Community Participation

6. Have there been any initiatives where students or staff volunteered to help with maintenance or cleanup activities?
7. Do you believe there is potential for more structured community participation (e.g., beach cleanups, small repairs, greening projects)?
8. Would your department be open to collaborating with student or community volunteers if the work is properly coordinated and approved?

Section 3: Feedback on the Proposed Application

9. What are your thoughts on a platform where users can both report problems and volunteer to help fix them?
10. Would a system that allows real-time issue tracking and resolution status be helpful for your team?
11. What features would you consider essential for such a system to be useful from a facilities management point of view?
12. Do you foresee any operational, legal, or safety concerns with allowing volunteers to assist in small maintenance activities?

Section 4: Technical and Administrative Concerns

13. Would your team benefit from an admin dashboard to prioritize, assign, or monitor reported tasks?
14. How important is integration with existing systems (like your current maintenance request process or IT systems)?
15. Would you prefer the system to send automatic reminders or notifications regarding unresolved issues?

Section 5: Final Insight

16. What would make this app more appealing or useful for institutional settings like RCSI?
17. Are there specific types of tasks or reports you think should or shouldn't be open to volunteers?
18. Would you be willing to pilot or test the app with your team if a beta version were made available?
19. Do you have any additional suggestions or recommendations that could help improve the effectiveness or usability of this application, either from a management or user perspective?

Appendix B
System Workflow Pseudocode

```
---------------------------------------------------------
PSEUDOCODE: USER REGISTRATION AND LOGIN
---------------------------------------------------------

Start

If User selects "Register" then
    Prompt User for:
        - Email
        - Password
        - Required personal information
    Validate inputs

    If inputs are valid then
        Create account using Firebase Authentication
        Redirect User to Login Page
    Else
        Display "Error: Invalid Registration Information"
    EndIf
EndIf


If User selects "Login" then
    Prompt User for:
        - Username
        - Password

    Authenticate using Firebase
    Send OTP to User's email

    If OTP and credentials are correct then
        Redirect User to Dashboard
    Else
        Display "Error: Incorrect Credentials"
    EndIf
```

```
EndIf



----------------------------------------------------------
PSEUDOCODE: CASE UPLOAD (ISSUE REPORTING)
----------------------------------------------------------

Start

User selects "Upload Case"

Prompt User for:
    - Title
    - Description
    - Phone Number
    - Location
    - Photo Upload

Validate inputs

If inputs are valid then
    Store case in database
    Notify Admin
    Display "Case Uploaded Successfully"
Else
    Display "Error: Invalid Input"
EndIf



----------------------------------------------------------
PSEUDOCODE: VOLUNTEER BROWSING AND SIGN-UP
----------------------------------------------------------

Start

User selects "Browse Volunteer Cases/Events"

Display list of available cases/events

If User selects "Sign Up" then
```

```
    Add User to volunteer list for selected task
    Display "You Have Successfully Signed Up"
Else
    Return to task list
EndIf


---------------------------------------------------------
PSEUDOCODE: SPONSOR BROWSING AND SUPPORT REQUEST
---------------------------------------------------------

Start

User selects "Browse Cases/Events"

Display list of available cases/events

If User selects "Support" then
    Display "Admins Will Contact You"
    Notify Admin
Else
    Return to task list
EndIf


---------------------------------------------------------
PSEUDOCODE: ADMIN DASHBOARD FUNCTIONS
---------------------------------------------------------

Start

Admin logs in

Display Admin Dashboard containing:
    - List of cases uploaded by volunteers
    - List of support requests from sponsors
    - Options to edit or delete cases/events
    - Options to create new cases or events

End
```

ملخص البحث

يُقدّم هذا المشروع تصميم وتطوير منصة "نحن التغيير" للمشاركة المجتمعية، التي تمكّن المواطنين من الإبلاغ عن المشكلات المحلية والمشاركة في أعمال التطوع عبر تطبيق هاتفي. يعالج المشروع مشكلة الإبلاغ غير الفعّال عن القضايا العامة والمشاركة المحدودة للمواطنين، حيث يتيح النظام للمستخدمين تقديم التقارير مع الصور والتسجيل للأنشطة تحت إدارة المشرفين. شمل المشروع تحليل النظام، وهندسة متطلبات الأمان، وتطبيق منهجيات نمذجة التهديدات، وتنفيذ مصادقة المستخدمين الآمنة ومعالجة البيانات باستخدام Firebase. أظهرت الاختبارات أداءً موثوقًا، وحماية قوية للبيانات، وتجربة مستخدم سلسة عبر الأجهزة. تُظهر النتائج أن المنصة تبسّط الإبلاغ عن المشكلات، تدعم إدارة البيانات الآمنة، وتشجّع على مشاركة مجتمعية أكبر، مما يؤكّد قيمة الحل العملية وقابليته للتوسّع.

الكلمات المفتاحية :مشاركة مجتمعية، منصة تطوّع، إبلاغ عن المشكلات، نمذجة التهديدات، Firebase، منهجية SQUARE، STRIDE، تقنية مدنية بحرينية، تطبيق هاتفي، مشاركة مجتمعية.