SOFTWARE ENGINEERING POE

ST10298850

Table of Contents

Question 1

Part 1

This document outlines the requirements for developing a custom software solution for the Tree-Saving Non-Profit Organisation. The goal is to create a web platform that enables the organisation to manage tree-planting initiatives, enhance communication with supporters, and simplify donation processing. The platform will allow users to register, track environmental progress, participate in events, and securely donate. It also provides administrators with tools to monitor tree-planting progress, coordinate volunteers, and oversee donation impact (Burak, 2023).

The goal of this software solution aligns closely with the non-profit's broader objectives by focusing on improving operational efficiency, strengthening community engagement, and fostering transparency with supporters. By integrating essential processes—such as volunteer management, donation tracking, and event scheduling—the software reduces administrative workload, enabling the organisation to direct more resources toward its core mission of environmental sustainability. Additionally, the platform's communication features facilitate timely updates and outreach, reinforcing donor relationships and boosting engagement. In summary, this project supports daily operations while enabling scalability, ensuring that the organisation can expand its impact, uphold accountability, and enhance donor satisfaction (Burak, 2023).

Functional Requirements

User Registration and Profile Creation:

Users will be able to set up login information, register, and establish profiles on the system. For the convenience of users, the registration procedure will be shortened while maintaining device compatibility (Burak, 2023).

Participation in Events:

Users will be able to see and register for events that the NGO is hosting. Administrators will oversee the event listing, guaranteeing current scheduling.

Donation System:

Through third-party gateways like PayPal, Visa, and Mastercard, users will be able to make donations using this system. Users can keep track of their contributions, see who is donating the most, and measure the impact of their contributions. The impact of the donation will match the quantity of trees bought.

Notification and Alerts:

To keep registered users up to date on the NGO's activities, they will receive emails or SMS notifications on events and fundraising initiatives.

Tree Location and Growth Tracking:

The system will offer a virtual map that is powered by an API service that displays all of the planted trees' locations and growth status. Pins will be used to identify each planted area.

Tree-Planting and Activity Impact Reports:

Subscribers will have access to comprehensive reports that highlight the sites of trees planted and the environmental effects of the organization's work. These reports will provide a thorough understanding of the organization's overall operations and advancement by containing data on the impact of tree-planting projects on CO2 emissions and other environmental indicators.

Admin Dashboard:

To manage tree-planting campaigns, volunteers, donations, and events, administrators will have access to a dashboard. This guarantees effective management and supervision of activities.

Donation Management:

Using the system, NGO employees will be able to track and oversee donations, as well as access donor details and gift history for auditing purposes.

Event Reports:

After an event is over, the system will provide comprehensive reports that include information about the number of trees planted, their species, the location, the date, and the supporters who took part.

Announcements page:

To ensure clear communication with users and supporters, administrators can post essential updates, alerts, and information about forthcoming events on the announcements page.

External Interface Requirements

External interface requirements define how the system connects with external systems, devices, and users (Burak, 2023). These specifications guarantee that the system can interact with external components in an efficient manner. The way the system interacts with humans, integrates with external hardware, and communicates with external software systems are the three main interactions that are described below.

- Payment gateways – The system will integrate with multiple payment providers to securely process donations from supporters. This will support a variety of payment methods, such as credit cards and digital wallets.
- Mapping API's – A third-party mapping service will be used to display the locations of planted trees. This will allow supporters and volunteers to track the progress and impact of their efforts visually.
- Messaging Integration - Users who have linked their contact details will receive notifications from the system via connections to external email and SMS services. This will enable effective communication with supporters and

participants by enabling real-time alerts for events, updates, or announcements. Secure and effective notice delivery will be achieved by utilising third-party suppliers.

Non-functional Requirements

Performance:

The web app should deliver fast and efficient responses, even during peak usage. This ensures users can seamlessly log in, donate, and sign up for events without delays. Performance optimizations will help ensure that features like event scheduling and donation processing are smooth, enhancing user satisfaction. As the organization grows, the system should continue to perform well under increased load, such as handling more events or users simultaneously (Indeed Editorial Team, 2024).

Security:

To protect sensitive information, security measures such as two-factor authentication will be employed. Role-based access ensures that only authorized personnel can manage administrative functions like scheduling events or accessing donor data (Indeed Editorial Team, 2024). Using external payment gateways (e.g., PayPal, Visa, Mastercard) eliminates the need to store sensitive financial details, reducing the risk of data breaches. Each donation will be validated for accuracy and transparency, ensuring that donation histories are securely maintained and accurate.

Usability:

The platform will be intuitive, accessible to users of all tech levels, and compatible across devices (desktop, mobile, and tablet). It will adhere to accessibility standards, ensuring usability for people of various abilities. The email and SMS notifications are optional, and users can opt out of receiving alerts through their profile settings, allowing them to customize their communication preferences (Indeed Editorial Team, 2024).

Availability:

High availability is essential for users relying on the platform to stay informed about events or make donations. A target uptime of 99% ensures minimal disruption to users, which is especially critical during key activities like event registration. Regular system backups will ensure that even in the case of an outage, data such as user participation and donation history is not lost, preserving the integrity and continuity of operations (Burak, 2023).

Scalability:

The system should be designed with growth in mind. As the organization expands its efforts, the platform should be able to handle more users, events, and donations without needing significant architectural changes (Indeed Editorial Team, 2024). Scalability ensures that the software remains cost-effective and efficient as user demand increases. The ability to add features and support more users seamlessly will allow the NGO to continue its mission without technological bottlenecks.

Appendix A

This appendix outlines the changes made to the introduction and scope of the document.

The initial submission provided a good foundation on the project's purpose and scope. However, after reviewing the feedback, adjustments were made to the introduction and scope of the document. By highlighting the goals of the software and align them with the goals of the organisation. In response to this, we have made the necessary adjustments. This was done by mentioning how the software solution directly supports the organisation's primary goals: improving operational efficiency, increasing community engagement, and maintaining transparency with the stakeholders.

In addition, we mentioned the expected benefits, including the reduced administrative workload and improved communication with supporters. The updated version provides a more goal orientated version that provides a clear and concise explanation of what the project is and how it aims to improve the organisation.

Part 2

For Tree-Planting Foundation I would suggest the ideal architecture pattern is the layered architecture pattern. When taking Tree-Planting Foundations focus on managing donations, creating and registering for events, along with an alerts system and user interaction. The layered architecture pattern provides a clear separation of concerns, leaving room to adapt and scale the software as needed. The layered architecture consists of four layers: presentation, business, persistence, and database. Because each layer is independent from one another changes can be made to one layer without the other layer being affected. This is due to a concept known as loose coupling which refers to minimizing the dependencies between the layers which allows each layer to function independently (Visual Paradigm, 2024) (Sommerville, 2021).

The presentation layer is responsible for displaying information to the user. The presentation layer collects input from the user and sends it to the application layer and receives data from the application layer and presents it to the user in a meaningful way (Sommerville, 2021) (GeeksforGeeks, 2021).

The business layer handles the core application functionality, by enforcing business rules and processing data. This layer serves as a bridge between the presentation layer and persistence layer. This layer processes user inputs from the presentation layer and applies business rules to manage the behaviour of the application (GeeksforGeeks, 2021) (Sommerville, 2021).

The persistence layer manages all the interactions that take place with the database layer by saving and retrieving data. This is done to separate the database from higher layers in the architecture. This layer is responsible for interacting with the database, performing tasks like retrieving, updating or storing data. It abstracts the complexity of database operations from the business logic (Sommerville, 2021) (GeeksforGeeks, 2021).

The database layer stores all the application's data, including user information, donations, events and all records of the organisations efforts. By isolating the database

interactions in their own layer, ensures that any changes made to the database storage/technologies, or optimisations can be done so without concern of negatively impacting the program (GeeksforGeeks, 2021).

The layered architecture pattern offers several benefits including the loose coupling as previously mentioned. It improves on flexibility, scalability, and maintainability, leading to the point that if changes are made to any one of the four layers the remaining layers will remain unchanged (Sommerville, 2021). Thus, allowing the business to evolve the capabilities of the software as needed in isolation. This is especially important for TPF, as it may need to adapt quickly to changing requirements or implementing new technologies (Sommerville, 2021).

Because TPF may need to react swiftly to changes like growing its donor base, launching new event formats, or adhering to updated legal requirements like South Africa's Protection of Personal Information Act (POPI), flexibility is very crucial for the organization. For instance, a new feature that generates receipts for donations for all donors together and for each contributor separately can be implemented by the organization without requiring a complete system upgrade; only the business layer needs to be altered (GeeksforGeeks, 2021)

For TPF, scalability is similarly crucial. If the system must handle growing amounts of data to manage activities like donations, events, and user interactions, this could result in a number of problems. But using this structure, you may upgrade certain parts as needed, such adding additional storage or putting in place more effective business rules, without affecting the rest of the system (Sommerville, 2021).

In reference to the independence of each layer, TPF can benefit by enabling modifications without disrupting the overall system. For example, the organisation wants to make changes to the user interface by adding more statistics to progress reports or adjusting the donation amounts. The presentation layer can be updated to include these changes without comprising the system, while the underlying business logic and database remain unchanged. Similarly, developers can implement advanced analytics or machine learning models for donor behaviour analysis in the business layer

without affecting the user interface. This modularity reduces development time and costs, because TPF is operating a resource-constrained environment (Jaiswal, 2024).

In continuation, the ability to make quick and efficient modifications to the system is a major advantage for developers. For example, if the organisation wanted to implement a new feature such as donation tracking. Which could provide the donors with personalised insights into their environmental impact, made by their contributions, which can be achieved by expanding on the business logic and persistence layers (Jaiswal, 2024) (Sommerville, 2021).

In conclusion, the layered architecture pattern provides a well-rounded, sustainable solution for TPF by strengthening scalability, flexibility, and maintainability over the long term. The architecture ensures that each layer can be developed and maintained independently, reducing the risk of disruptions to the system. This modularity is particularly beneficial as the foundation grows and its needs evolve. For example, TPF can seamlessly introduce new features such as improved donation portals, multilingual support, or real-time updates on tree-planting progress, ensuring the software continues to meet the organization’s operational and strategic objectives. By adopting this architecture, TPF positions itself to efficiently manage donations, events, and user interactions, thereby supporting its mission of environmental sustainability and community engagement. (GeeksforGeeks, 2021) (Sommerville, 2021).

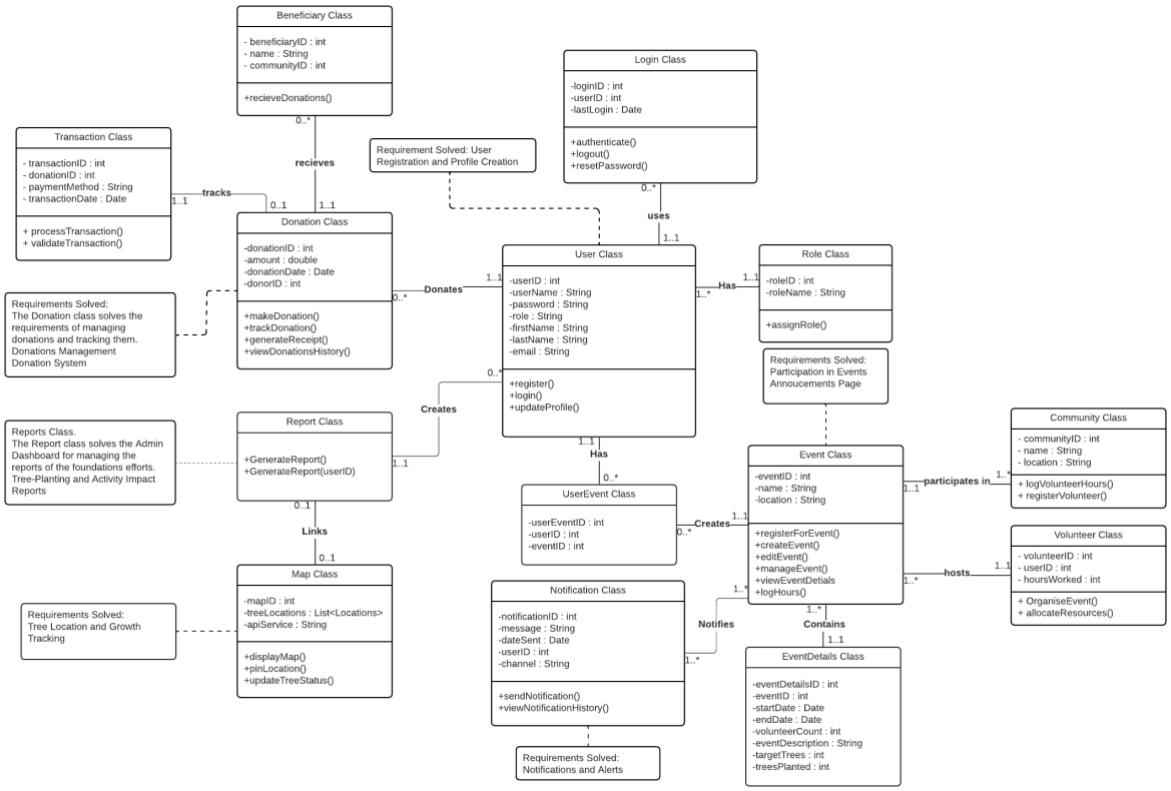


Figure 1: UML Class Diagram of Tree-Planting Foundation Software Development

(Sommerville, 2021) (Microsoft, 2024) (Lucid Software, 2024) (Visual Paradigm, 2024)

UML Diagram Description:

The above UML class diagram showcases several classes linked together that perform the intended functions of the Tree-Planting Foundation (TREE-PLANTING FOUNDATION) program. The User class acts as the central class, holding all user information and operations for creating a user. This class is linked to the Role class, which assigns the role of each user (volunteer, staff, admin). Following that is the Donations class, which holds all information and operations regarding donations. The newly added Transaction class complements this by tracking donation transactions and payment methods. Additionally, the Report class is used to generate reports on the foundation's efforts and statistics. The Events class manages all events, while the Notification class creates alerts and notifies users about these events. The UserEvents class supports users who wish to register and sign up for events, and the added Volunteer, Beneficiary, and Community classes enhance the system's ability to represent the foundation's

programs and participants effectively (Microsoft, 2024) (Sommerville, 2021) (Visual Paradigm, 2024).

Appendix B

This appendix outlines the changes made to the report on the chosen architecture pattern for the software solution and the UML class diagram.

1. Changes to the report

Enhanced motivation for architecture choice:

The initial submission provided a good foundation of the architecture patter however, it was lacking in terms of the motivation and aligning with the foundations goals. In response to the feedback, examples were added to showcase how the system and architecture pattern will align with the organisations goals. This was done by bringing in flexibility, scalability and modularity by building on the existing report.

2. Changes to the UML Class Diagram

The UML class was updated to incorporate the missing entities and improve cohesion in the system.

- Transaction class: To validate and process donation related payments.
- Beneficiary class: This class holds the entity benefiting from the donations reflecting the foundations impact.
- Volunteer class: To manage the hours worked by the volunteers at the events, and manage the resources better.
- Community class: To capture community details, allowing for better tracking of the foundations resources to better equip themselves and their events.

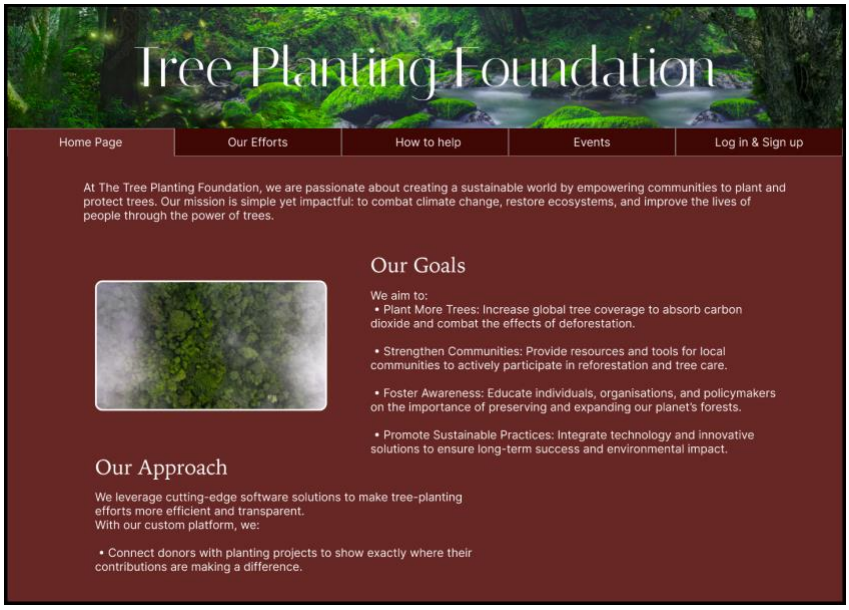
Additionally, by going through the system, I have improved on the relationships between the different classes.

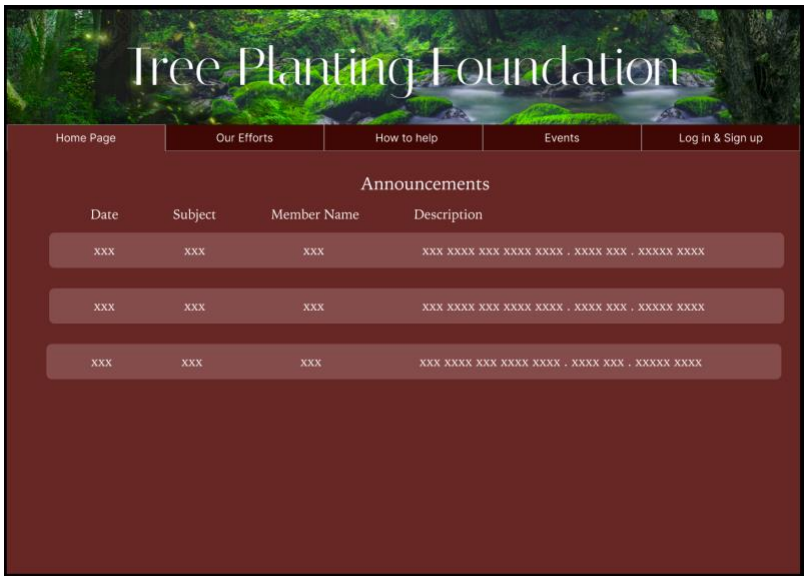
These updates enhance the system's alignment with the Tree-Planting Foundation's operations, creating a more robust and comprehensive framework to effectively meet its requirements.

Question 2

Landing Page

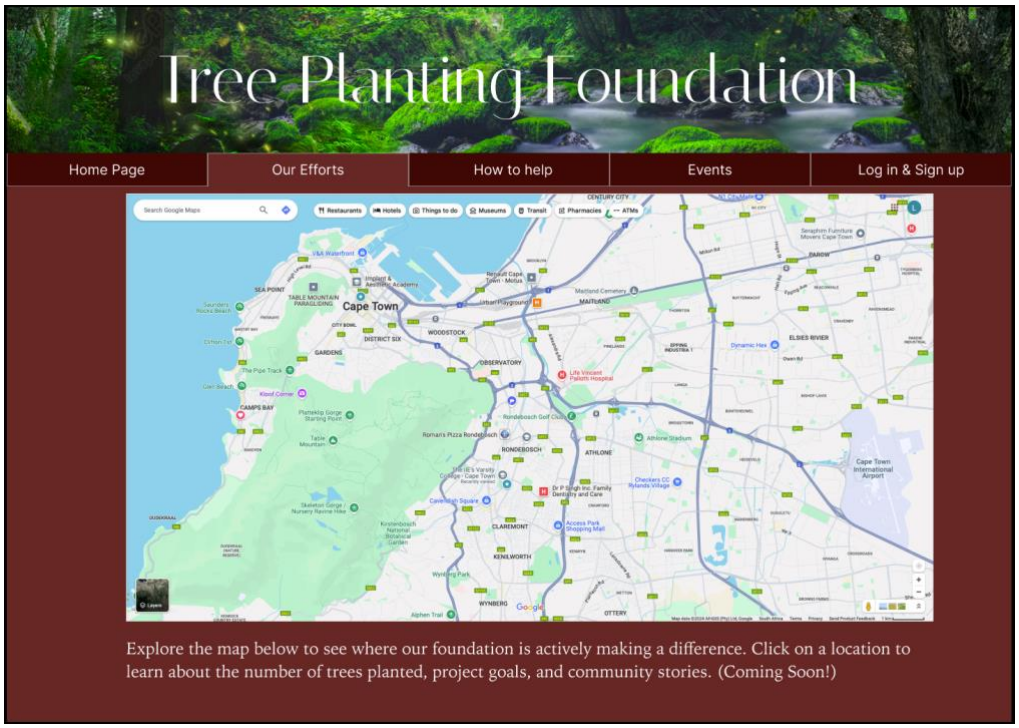
This is the landing page, this page is dedicated to volunteers or donators. Providing the most basic information about the cause with a scroll feature. Beneath the first image lies an announcement page which any type of user will have access to view. These announcements will be managed by admins, to keep their members up to date. This page solves the functional requirement of Announcements Page (Sommerville, 2021).

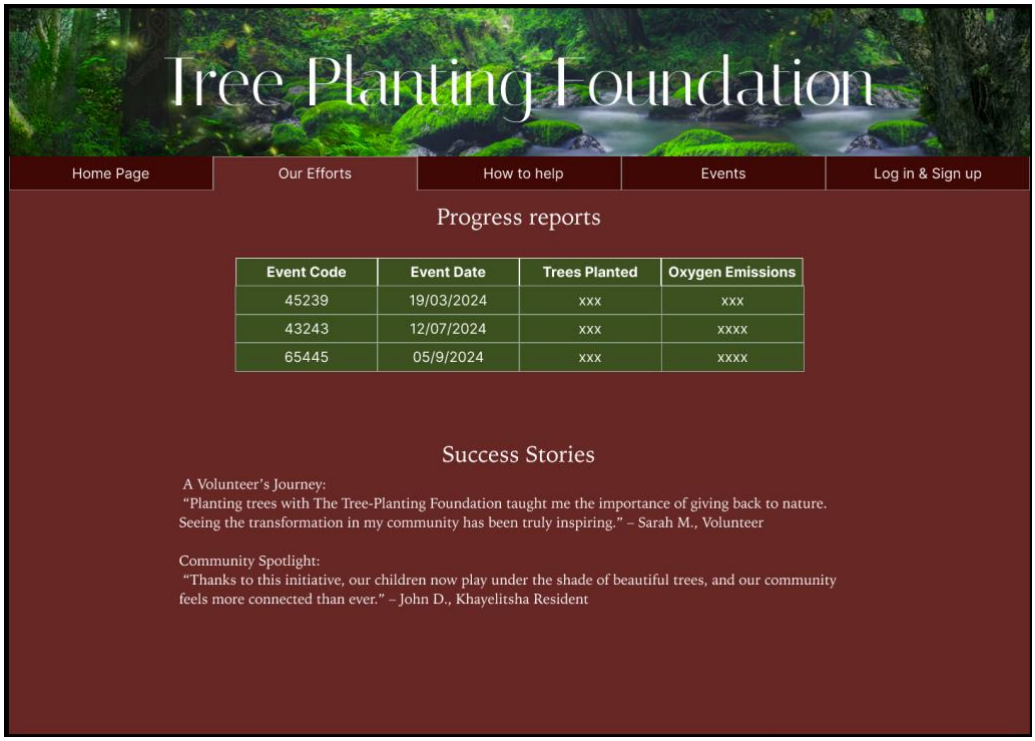


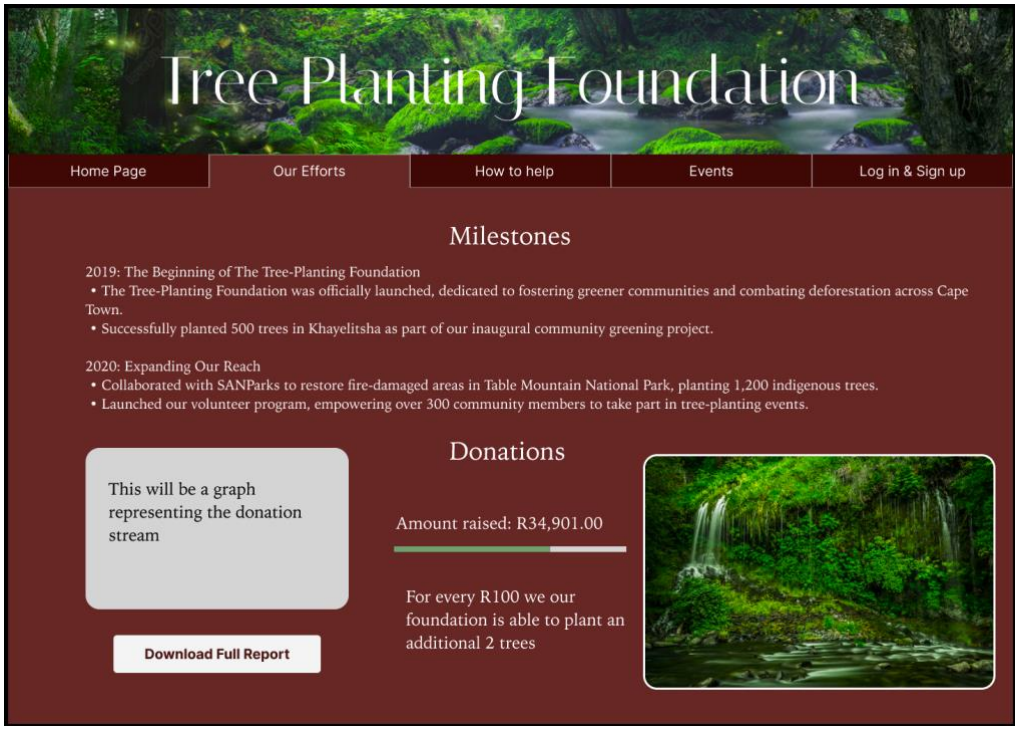


Our Efforts Page

This following pages are designed for volunteers or donators. The first image is an interactive map, which would highlight all the locations the NGO has planted trees. Scrolling down, the website displays a progress report at a basic level for the user to quickly read and understand. Additionally, there is a download report option this is there for the users whom are wanting to view a detailed report incorporating a significantly more advanced report. Along with a section on success stories from members and their experiences. At the bottom is a section about the milestones the foundation has reached and what they are planning on(just an example). The purpose of this page is to inspire other members to donate or participate in current and future programs. The user can see a list of the recent donations and the total amount raised. This page solves the following requirements, tree location and growth tracking, tree-planting and activity reports and, event reports. These pages resolve the following functional requirements: Tree-Planting and Activity Impact Reports, Tree Location and Growth Tracking and Event Reports (Sommerville, 2021).

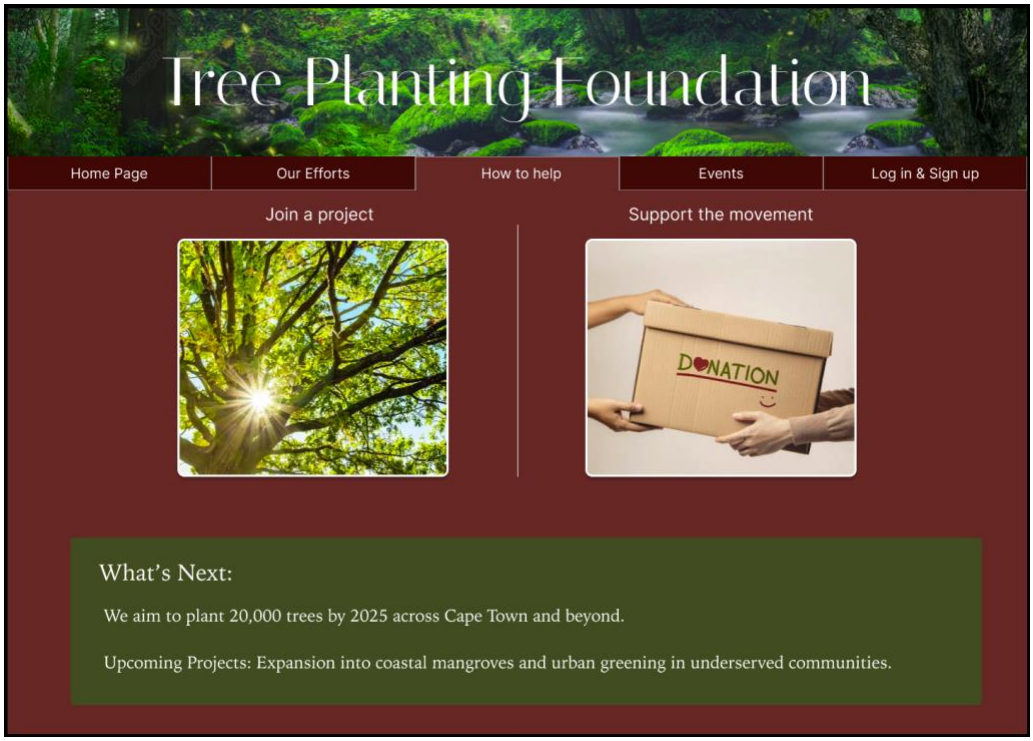



Tree Planting Foundation
Home Page
Our Efforts
How to help
Events
Log in & Sign up
Progress reports
Event Code
Event Date
Trees Planted
Oxygen Emissions
45239
19/03/2024
xxx
xxx
43243
12/07/2024
xxx
xxxx
65445
05/9/2024
xxx
xxxx
Success Stories
A Volunteer's Journey:
"Planting trees with The Tree-Planting Foundation taught me the importance of giving back to nature. Seeing the transformation in my community has been truly inspiring." – Sarah M., Volunteer
Community Spotlight:
"Thanks to this initiative, our children now play under the shade of beautiful trees, and our community feels more connected than ever." – John D., Khayelitsha Resident


Tree Planting Foundation
Home Page
Our Efforts
How to help
Events
Log in & Sign up
Milestones
2019: The Beginning of The Tree-Planting Foundation
• The Tree-Planting Foundation was officially launched, dedicated to fostering greener communities and combating deforestation across Cape Town.
• Successfully planted 500 trees in Khayelitsha as part of our inaugural community greening project.
2020: Expanding Our Reach
• Collaborated with SANParks to restore fire-damaged areas in Table Mountain National Park, planting 1,200 indigenous trees.
• Launched our volunteer program, empowering over 300 community members to take part in tree-planting events.
This will be a graph representing the donation stream
Download Full Report
Donations
Amount raised: R34,901.00
For every R100 we our foundation is able to plant an additional 2 trees

How to help Page

The following pages are designed for volunteers or donators. This web page provides information about participating and registering for the events and how to donate to the foundation. On the first screenshot below there are 2 images with the headings “Join a project” and “Support the movement” the purpose of this is to allow the users to reach their desired location. Following that the next page provides a list of the most recent donations to the foundation, to inspire users to donate. Additionally, users can view the existing programs. On the last image, is the form where users can donate via their desired payment method. Users are urged to consider opting for a recurring payment option. These pages solve the following requirements donation system. These pages solve the following functional requirements: Donation System (Sommerville, 2021).

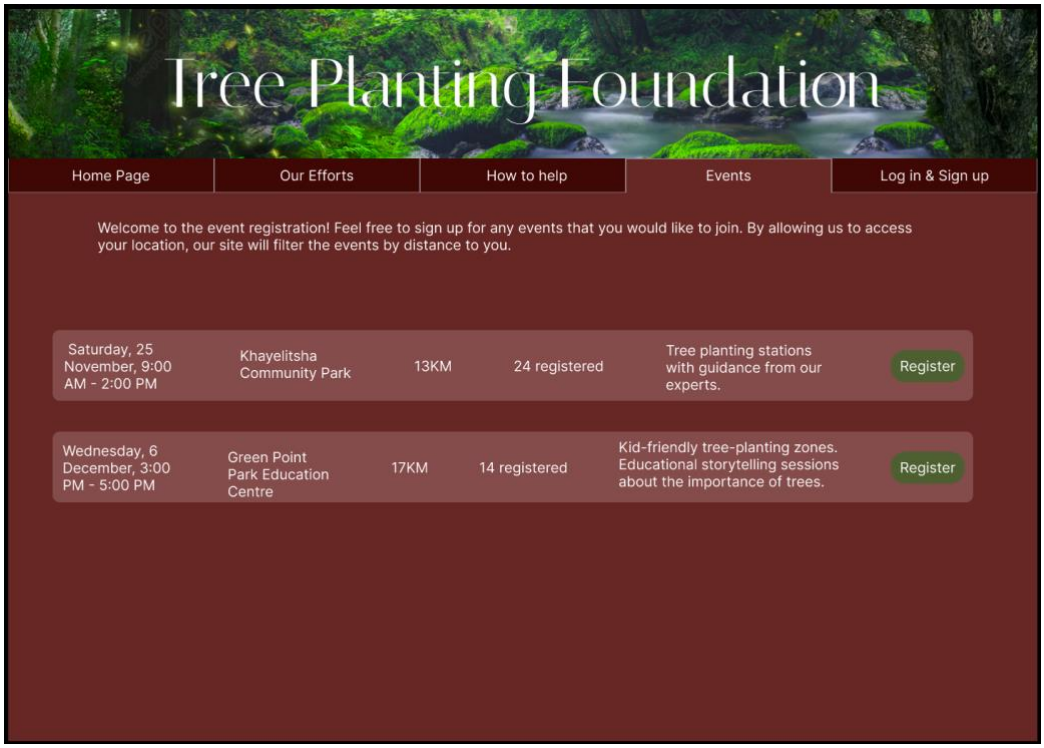



Tree Planting Foundation
Home Page
Our Efforts
How to help
Events
Log in & Sign up
Start Funding
Donation Activity
John Parker has just donation R325
Malcom Marks has just donation R1250
Cheslin Kolbe has just donation R600
Cheslin Kolbe has just donation R600
Existing Programs
Mitchells Plain
Athlone
Khayelitsha Township


Tree Planting Foundation
Home Page
Our Efforts
How to help
Events
Log in & Sign up
Every Donation Grows a Forest
Please indicate the amount
R50
R100
R200
R500
Enter a custom amount
Make this a monthly contribution.
SnapScan
VISA
Thank you for your contribution!

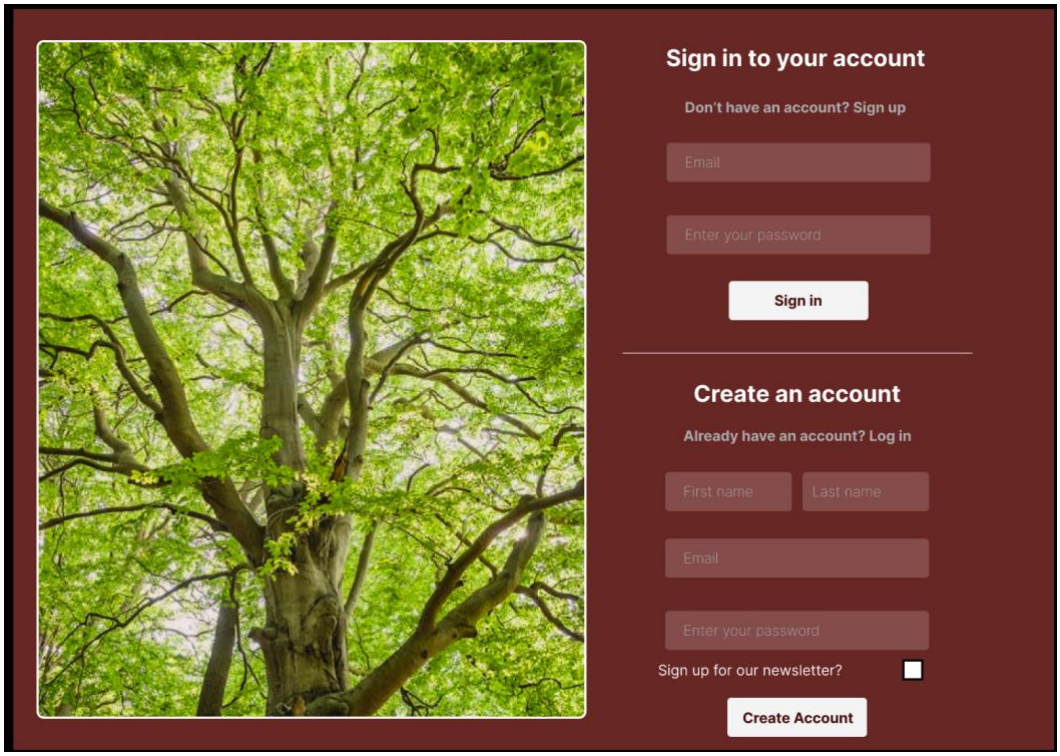
Events Page

The events page is designed for users who such as volunteers or donators. This page displays all scheduled events that users can register to participate in. Users are required to have signed in to view be able to register for the event. Additionally, users can view relevant information about the event on this page. This page was designed to solve participation in events functional requirement. Once a user has registered this and partaken in this event, their contributions will be recorded and available to download on the “Our Efforts” page. This page solves the functional requirement of Participation in Events (Sommerville, 2021).



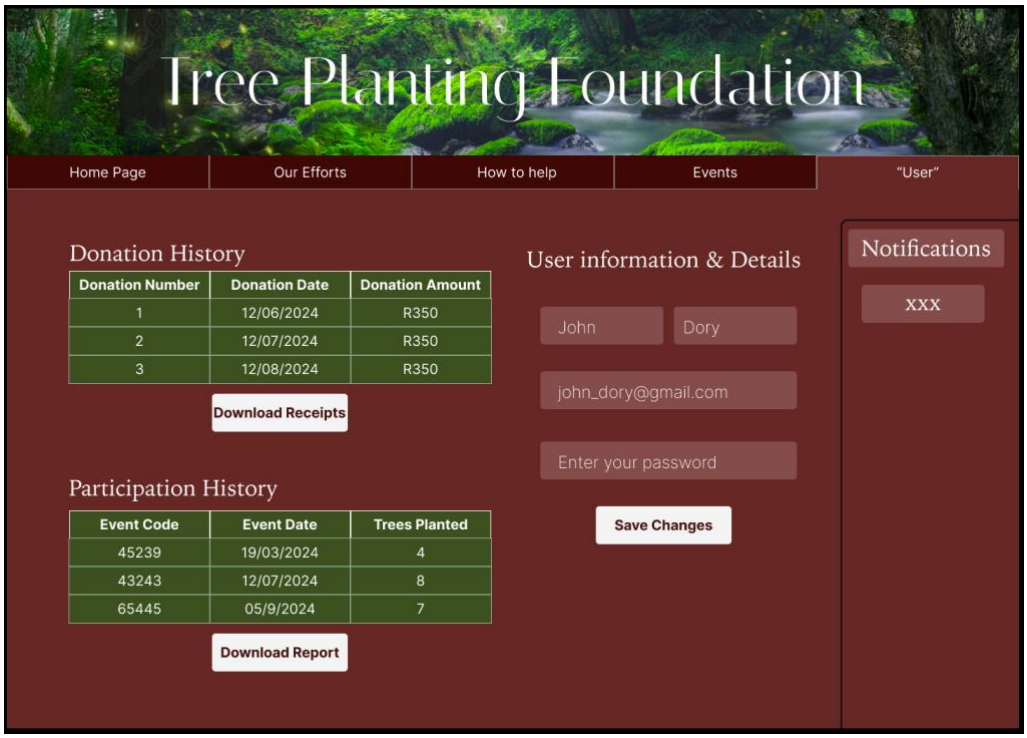
Sign In and Register Page

The following page was designed to allow users to register and create an account to make use of the website. Allowing new members to register and existing users to sign in. Upon a user register they are given the option to opt for the newsletter. This solves the functional requirement of notifications and alerts, with the foundation being able to directly inform its members externally. Beyond that by providing profile registration solves the functional requirement for user registration. The design maintains a user friendly appearance with an easily understood layout (Sommerville, 2021).



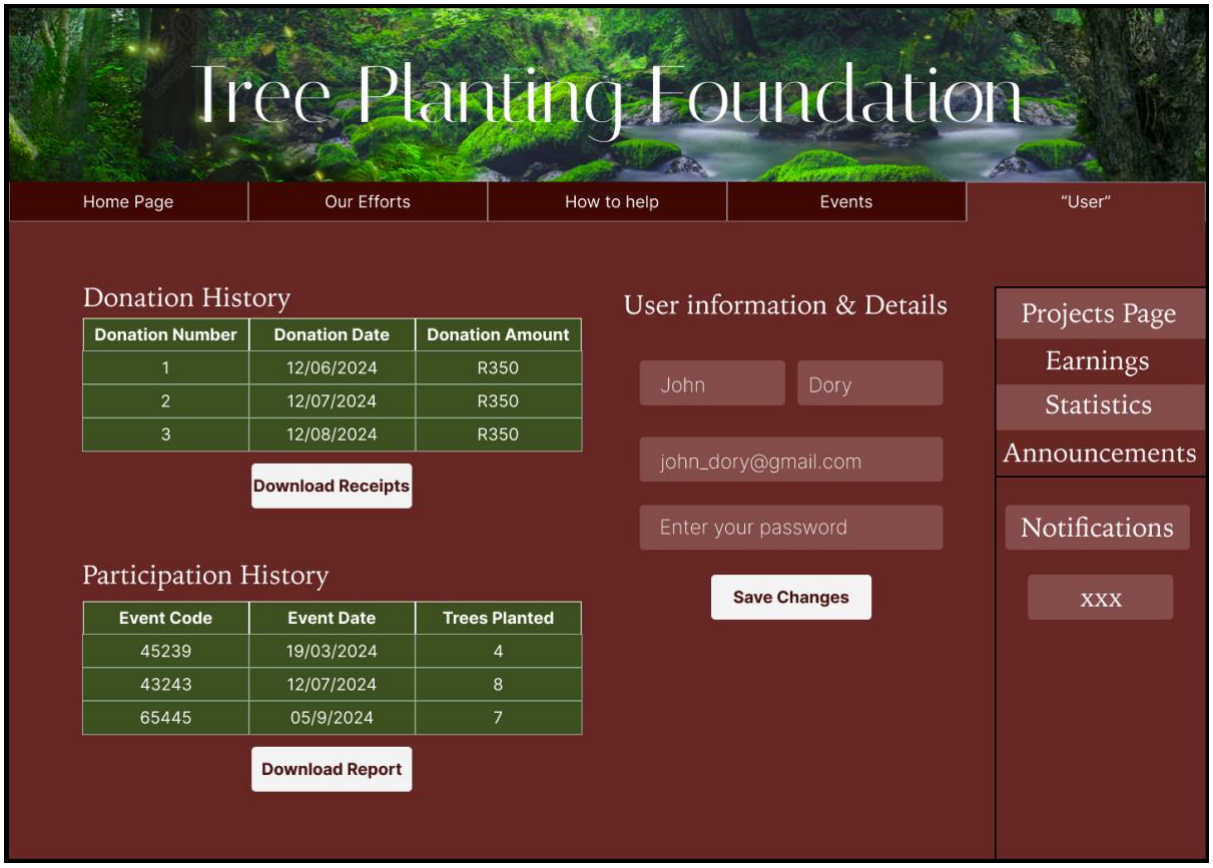
User Information Page

The following page is designed for both all types of users with a slight change for staff members and administrators. The page show the users history in event participation and donations with detailed reports and receipts available for download. This was done to prevent clutter and overly complicated designs which may result in a less friendly user experience. Additionally, this page allows for users to edit their personal information and view notifications sent by the staff members of the foundation. This page helps solve the donation system functional requirement by providing a user's individual contributions and information. The reason for incorporating an additional notification view is in the event a user has not opted to receive newsletters and email. Then those notifications would display here. Ensuring a well-developed and consistent experience. This page solves the functional requirement of User Registration and Profile Creation and Notification and Alerts (Sommerville, 2021).

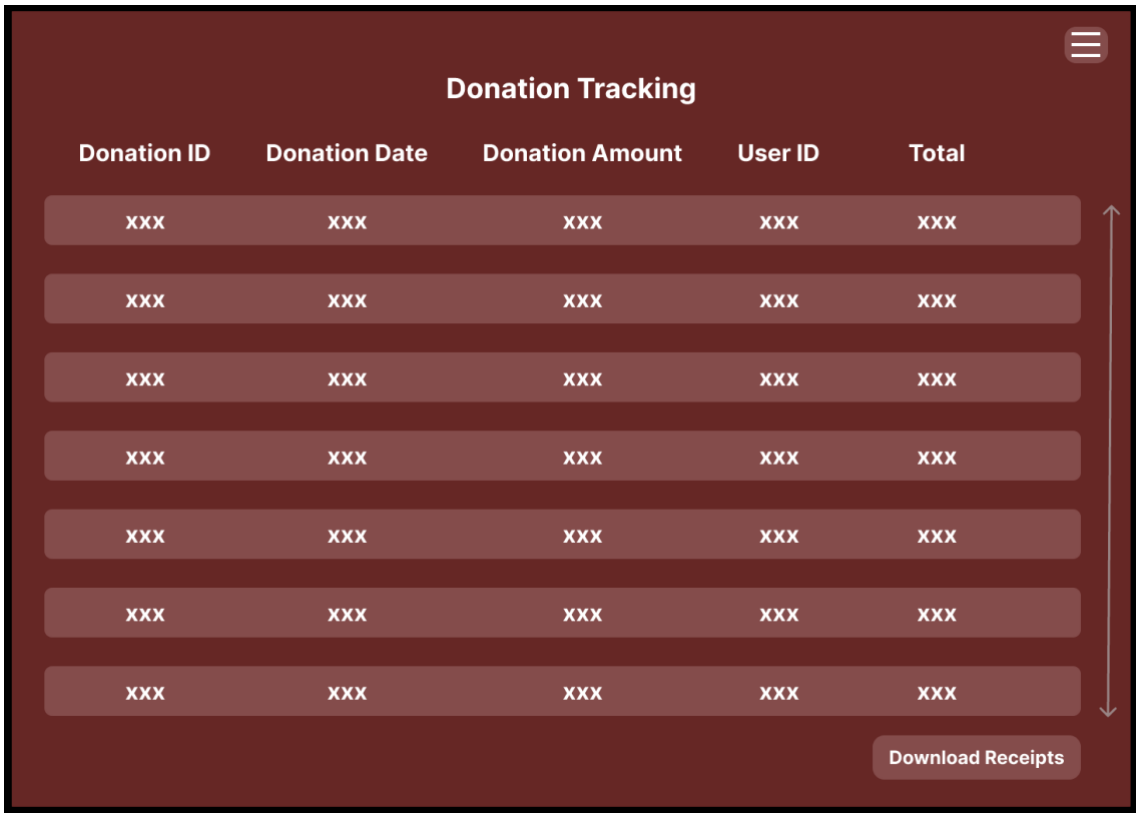


Admin Pages

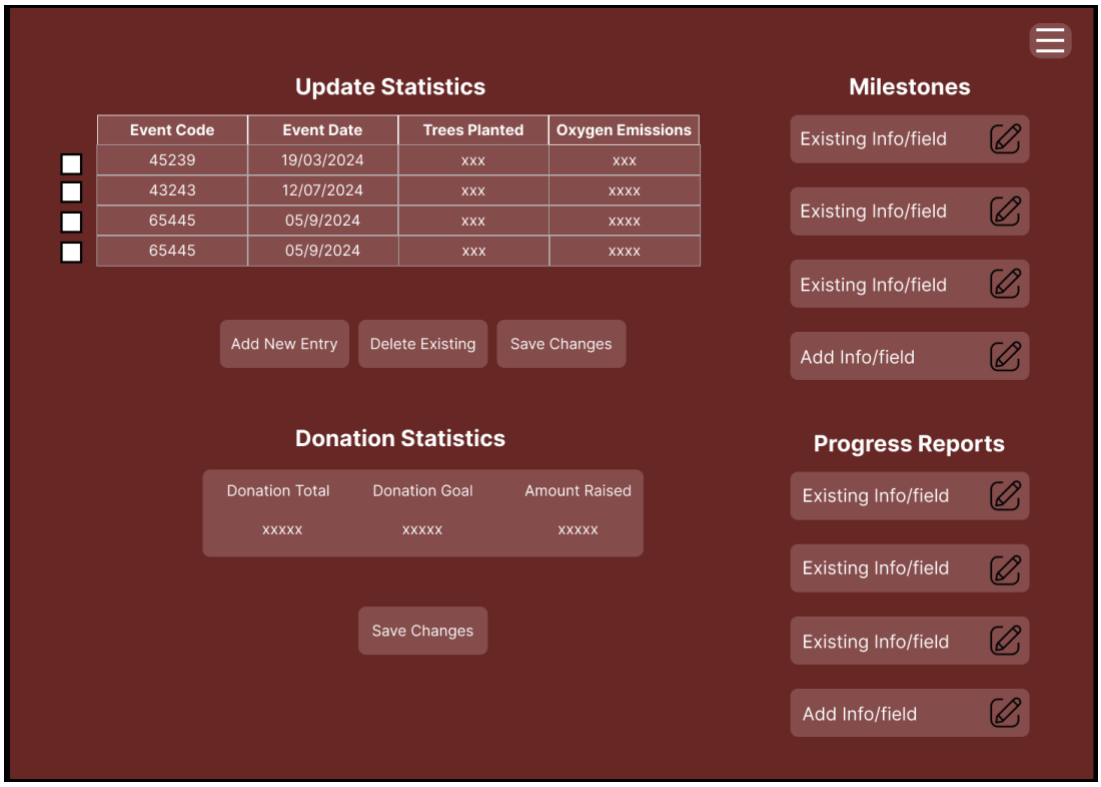
The following page is an adjusted version of the existing user information page. This page is designed for the administrators of the foundation. Only users who account have been granted a higher level of authorisation are able to reach these pages. On the right hand column is a navigation bar that leads to several other admin restricted pages. The purpose of these pages is to solve the functional requirement of an admin dashboard. Which requires admins to be able to edit information such as, progress tracking, events, programs, notify users and track donations (Sommerville, 2021).



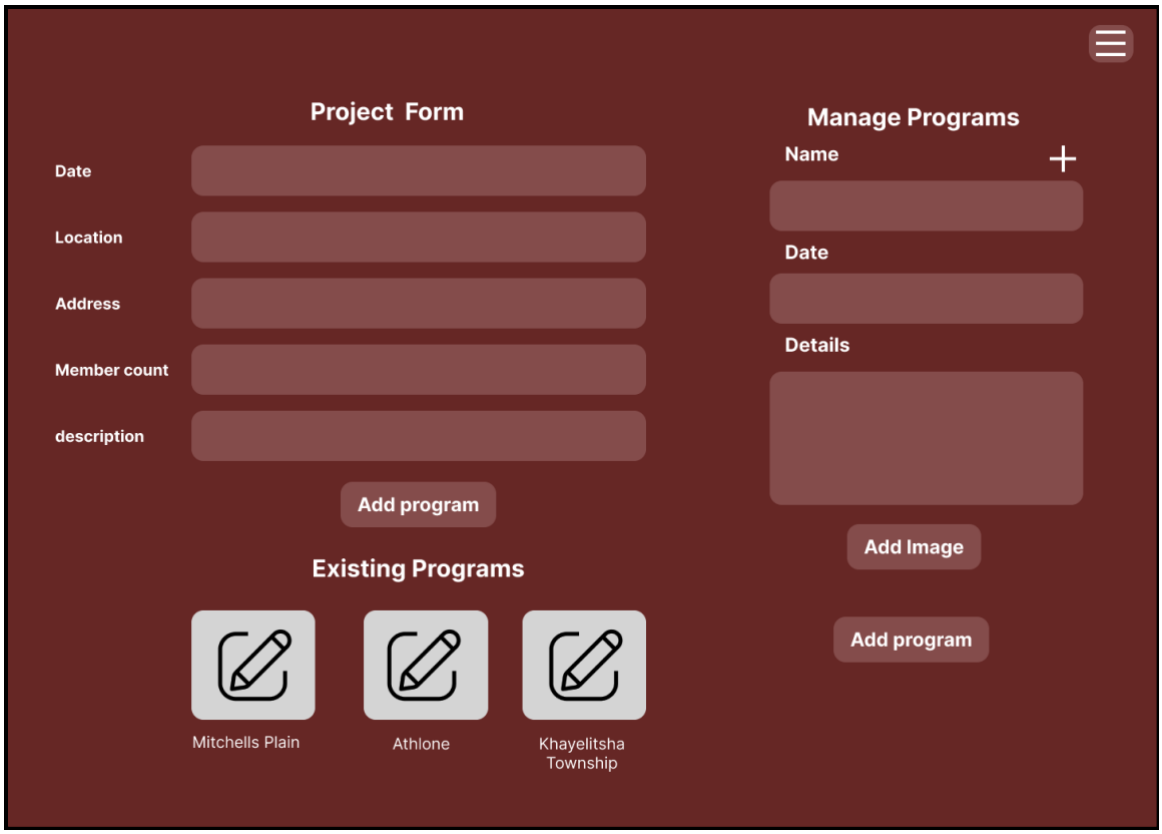
The page below was designed for administrators to access the donation records. The design allows for admins to download the data into a csv for auditing purposes. Therefore, allowing the admins to retrieve all financial records of donations. Additionally, with the UI design on the “Our Efforts” web page. The graph will automatically update along with the progress bar in accordance with the percentage gained upon the donated amount. The purpose of this functionality is to improve efficiency and solve the functional requirement of the donation management. Which is allows the admins to view all donor details (Sommerville, 2021).



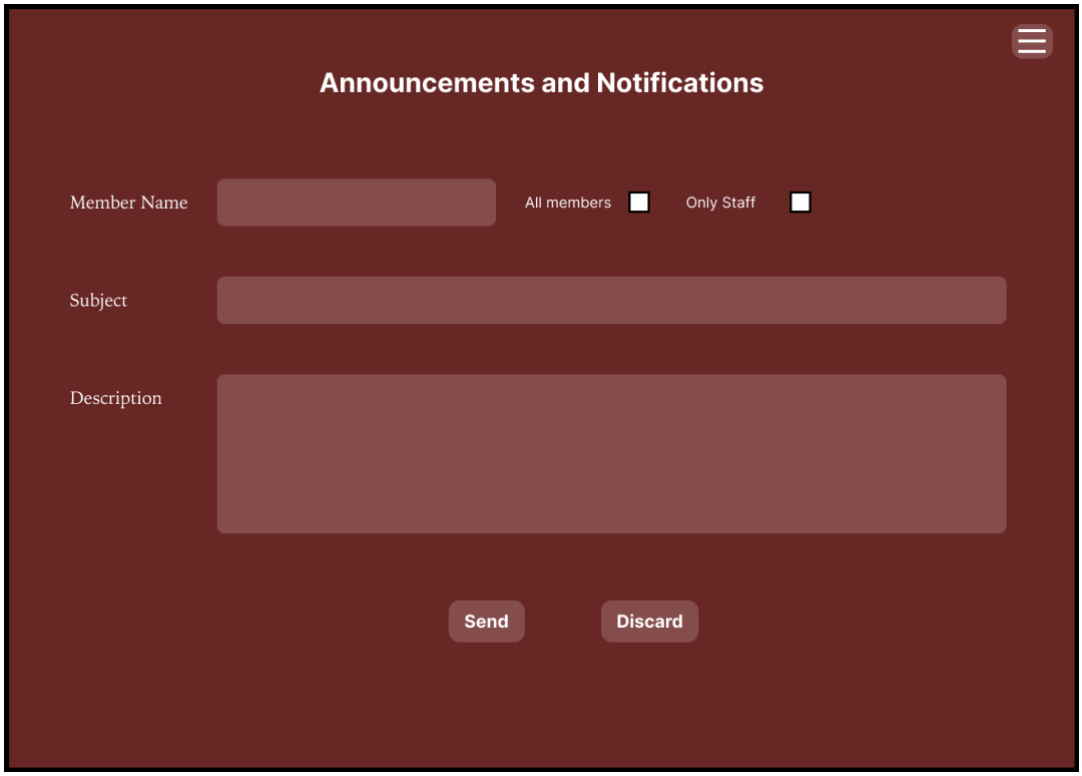
The following page is to allow the admins to alter the information on the other web pages. The top table is to enter the stats available on the “Our Efforts” page. Granting admins the ability to add new records or delete them as a whole, solving the functional requirement of the Admin Dashboard. Following that, the columns on the right side manage the milestones, and progress reports which is just an entry of a string which is set to have predefined formatting to match its destination in the “Our Efforts” Page. Lastly, the Donation statistics is used to manage to progress bar and graph to display their level of donations to their members in terms of their goals (Sommerville, 2021).



The following page is designed for administrators to use to alter information regarding projects or programs for the foundation. For example, if an admin wanted to create a new project(event) for the community, that individual would be able to by making use of the form on the left side. The right hand column and fields are used to create a new program, which encompasses more information as one program can have many events but a project can either be linked to a program or independent. When a user creates or alters a project the information displayed on the “Events” page will be updated in real time. The process is the same with the programs, when an admin makes any changes to the programs the information displayed on the “How to help” page will be updated (Sommerville, 2021).



The following page is designed for the administrators. This page allows admins the create announcements and send notification to all or any member with a registered account. This page has a basic email layout, purely for functionality over appearance. This plays a role in the admin dashboard functional requirement and Notification and Alerts (Sommerville, 2021).



Question 3

I have learned a lot from this assignment about how important colour schemes are when creating an interface that is easy to use. I discovered that choosing the appropriate colours has an impact on accessibility and usability in addition to improving the system's aesthetic appeal. By directing attention, producing contrast, and facilitating natural navigation, colour selections can enhance the user experience. Creating a whole system from the ground up was one of the biggest hurdles I encountered. In order to ensure a seamless user experience and a unified design, it was necessary to carefully evaluate how each component of the system would interact. Every element had to make sense and perform properly inside the bigger structure, thus it was a complicated process. Furthermore, developing my understanding of design concepts like alignment, hierarchy, and consistency has been crucial to producing a system that is both aesthetically pleasing and simple to use. I created a functional system that is easy to use for administrators and users alike by putting these ideas into practice. All things considered, this assignment has improved my comprehension of the value of careful design in developing a system that is both practical and user focused (Sommerville, 2021).

POE SUBTOTAL				/100
RUBRIC POE — Self-Evaluation	Levels of Achievement			Score
	Excellent	Good	Developing	
	Score Ranges Per Level (½ marks possible)			
Criteria (3)	3 • Bravo! You have done exceptionally well!	2 You are on the right track, but you can do better!	0 to 1 You have learned something – but you are not proving it!	2 1/2
Reflection (3)	3 • Reflection shows thorough thoughtfulness. • Reflection has several supporting details and examples. • All parts of the reflection are complete and done.	2 Reflection shows little thoughtfulness. Reflection has few details or examples. Most parts of the reflection are incomplete.	0 to 1 • Reflection shows no thoughtfulness. • Reflection has no details. • Reflection is incomplete.	2 1/2
Demonstration of Learning (4)	4 • Clearly explains what was learned. • Reflection is beyond simple description of event/experience to an analysis of how it contributed to learning and understanding.	2 to 3 The reflection demonstrates student's attempt to analyse the event/experience but fails to demonstrate depth of analysis.	0 to 1 • Reflection does not move beyond description of the event/experience.	3
POE – Self Evaluation TOTAL				/ 10

6

Bibliography

Burak, A., 2023. *Your guide to writing a Software Requirements Specification (SRS) document*. [Online]
Available at: https://relevant.software/blog/software-requirements-specification-srs-document/
[Accessed 16 September 2024].

GeeksforGeeks, 2021. *Types of Software Architecture Plans*. [Online]
Available at: https://www.geeksforgeeks.org/types-of-software-architecture-patterns/
[Accessed 21 October 2024].

Indeed Editorial Team, 2024. *Functional vs. Nonfunctional Requirements (With Exmples)*. [Online]
Available at: https://www.indeed.com/career-advice/career-development/common-functional-and-non-functional-requirements
[Accessed 19 September 2024].

Jaiswal, S., 2024. *Understanding Layered Architecture: A Comprehensive Guide*. [Online]
Available at: https://medium.com/@satyendra.jaiswal/understanding-layered-architecture-a-comprehensive-guide-4c2eee374d18
[Accessed 21 October 2024].

Lucid Software, 2024. *UML Class Diagram Tutorial*. [Online]
Available at: https://www.lucidchart.com/pages/uml-class-diagram
[Accessed 22 October 2024].

Microsoft, 2024. *Claims-based authorization in ASP.NET Core*. [Online]
Available at: https://learn.microsoft.com/en-us/aspnet/core/security/authorization/claims?view=aspnetcore-8.0
[Accessed 23 October 2024].

Sommerville, I., 2021. *Software Engineering*. 10th Edition ed. Harlow: Pearson.

Visual Paradigm, 2024. *UML CLass Diagram Tutorial*. [Online]
Available at: https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/
[Accessed 22 October 2024].