**Mobile Systems Development**

**COMP826**

Milestone 2: Prototype, Evaluation and Recommendations

Lecturers: Prof. Roopak Sinha

Dr. Matthew Kuo

Baoquan Cheng

19072667

Contents

[1. System Overview 3](#_Toc149157906)

[2. System Artefacts 4](#_Toc149157907)

[2.1 Features evidence and demo 4](#_Toc149157908)

[2.2 Design and implementation 4](#_Toc149157909)

[2.2.1 VSCode 4](#_Toc149157910)

[2.2.4 Server 8](#_Toc149157911)

[2.3 Development Process 10](#_Toc149157912)

[2.4 Physical Considerations 12](#_Toc149157913)

[3. System Evaluation 13](#_Toc149157914)

[3.1 Functionality 14](#_Toc149157915)

[3.2 Performance and Scalability 14](#_Toc149157916)

[3.3 User Experience (UX) and Design 15](#_Toc149157917)

[3.4 Maintainability 15](#_Toc149157918)

[4. Conclusion and Recommendations 16](#_Toc149157919)

[4.1 Limitations 16](#_Toc149157920)

[4.2 Recommendations 16](#_Toc149157921)

[References 18](#_Toc149157922)

# System Overview

Environmental problems including air pollution, global warming, and others are getting worse. More and more scientific data points to more severe and sooner than previously anticipated effects of global warming. The hydrologic cycle and global ecosystems would both be severely disrupted by warming (Plumer, 2023). The globe is already getting close to the point at which climate change could be deemed harmful (Moore, 2009). According to WHO data, 99 percent of the world's population breathes air that contains high levels of pollutants and exceeds WHO guideline limits, with low- and middle-income nations experiencing the largest exposures (*Air Pollution*, n.d.). Globally, the climate and ecosystems of the planet are closely related to air quality.

In an era marked by environmental challenges, the need for collective action and collaboration has never been more critical. Addressing this need, we are thrilled to introduce our cutting-edge solution: the Environmental Volunteer Social Platform. This innovative digital ecosystem serves as a dynamic hub where environmental enthusiasts, volunteers, and change-makers converge to make a tangible difference in the world.

At its core, our Environmental Volunteer Social Platform is a testament to the power of technology harnessed for social and ecological good. This platform is not just an app; it's a vibrant community, a catalyst for positive change, and a sanctuary for those passionate about environmental causes. Whether you are an eco-enthusiast, a seasoned environmentalist, or someone seeking to contribute meaningfully to the planet, this platform welcomes you with open arms.

Our platform boasts an array of intuitive features designed to facilitate seamless collaboration and engagement. Users can effortlessly discover and join environmental projects and events, connect with like-minded individuals, and participate in discussions that spark innovation and awareness. The platform's user-friendly interface ensures that anyone, regardless of their technical expertise, can navigate, interact, and contribute effectively.

One of the platform's defining strengths lies in its ability to foster collaboration. Users can easily team up with others who share their passion and skills, working together on projects that range from local community clean-ups to large-scale environmental initiatives. The platform serves as a digital workspace where ideas flourish, partnerships form, and real-world projects take shape.

More than just a platform, our Environmental Volunteer Social Platform is a driving force behind a global movement. By connecting individuals, organizations, and grassroots initiatives, we aim to build a sustainable future collaboratively. Through shared knowledge, collective efforts, and unwavering dedication, we believe in the transformative power of this platform to inspire, educate, and mobilize people around the world.

I took on the role of a developer. My primary focus was on the technical aspects of the

project, from the design to the implementation of the app. My role involves the following key

responsibilities:

* App Design and Architecture: Creating a well-thought-out app architecture to ensure a seamless and intuitive user experience.
* Front-end Development: Implementing the app's user interface, including navigation, layouts, and interactive elements.
* Back-end Development: Building and managing the server-side components required for data storage, event management, user authentication, and other functionalities.
* Integration: Integrating external APIs and services, such as location-based services and social media platforms, to enhance app functionality.
* Continuous Improvement: Staying updated with the latest mobile development trends and technologies to improve the app's performance and user experience.

# System Artefacts

## Features evidence and demo

The client code repository is:

<https://github.com/victorbqcheng/EarthFriend>

\*Note: the username is *test* and password is *test*.

You can download pre-built apk here:

<https://github.com/victorbqcheng/EarthFriend/releases>

The server code repository is:

<https://github.com/victorbqcheng/EarthFriendAPI-New>

The video on YouTube:

<https://youtu.be/OPOGisOY8fE>

## 2.2 Design and implementation

I used the React Native framework and applied the MVVM design pattern to develop the client. I used VS Code as the editor, as shown in Figure 1.

### 2.2.1 VSCode

VS Code is one of the most popular code editors for web and mobile development. It offers a rich ecosystem of extensions, a powerful debugging environment, and Git integration, making it a top choice for developers worldwide (*Visual Studio Code - Code Editing. Redefined*, n.d.).

电脑萤幕的截图

描述已自动生成

Figure 1 vscode

##### 2.2.2 React Native

Introduced in 2015, React Native is a widely adopted open-source framework for cross-platform mobile applications developed by Facebook. Derived from React, Facebook's JavaScript library for constructing user interfaces, React Native directs its focus away from the browser  
and toward mobile platforms(*React Native · Learn Once, Write Anywhere*, n.d.). Its ability to write code once and run it on both Android and iOS devices significantly reduces development time and costs. As shown in Figure 1, I used version 0.72.4 of React Native.



Figure 2 React Native

React Native utilizes JavaScript, a widely-used technology in web development. This enables developers skilled in JavaScript to easily create mobile applications using React Native. The framework offers various external plugin options, including both JavaScript-driven and native modules (Roller, 2023). These external plugins significantly enhance the app's functionalities and enhance its performance. React Native adopts a community-driven approach, supported by a large community of more than 50,000 active contributors. This active community ensures abundant resources for obtaining expert assistance and guidance (*Benefits of Using React Native for Mobile App Development*, n.d.).

As shown in figure 3, besides React Native, I also utilized several third-party libraries. For instance, I used 'react-navigation/bottom-tabs' (*React Navigation | React Navigation*, n.d.) for implementing the bottom navigation functionality (shown in Figure 4), 'expo-image-picker' (*Expo-Image-Picker*, 2023) for accessing the phone gallery (shown in Figure 5), and 'jsonwebtoken' (*Jsonwebtoken*, 2023) for JWT implementation. These libraries facilitated the rapid implementation of the required features.

文本

描述已自动生成

Figure 3 packages

图形用户界面, 文本, 应用程序, 聊天或短信

描述已自动生成

Figure 4 bottom navigation

图形用户界面, 应用程序

描述已自动生成

Figure 5 image picker

React Native is shaping the way people interact with Meta's diverse products, including Facebook Marketplace, Messenger Desktop, and Ads Manager, as well as the Oculus companion application and various others. Microsoft utilizes React Native to deliver outstanding user experiences in some of its most well-known applications. Wix, with one of the most extensive React Native codebases globally, has a rich history within the development community and supports multiple open-source projects. As an early adopter of React Native, Wix employs it across its entire range of applications (*Showcase · React Native*, n.d.).

##### 2.2.3 MVVM

MVVM promotes clean separation of UI, business logic, and data management, enhancing code readability and maintainability. By separating concerns, developers can focus on specific aspects without disrupting the entire codebase. MVVM enhances the code's modularity and maintainability (“Introduction to Model View View Model (MVVM),” 2019). In the research done by Lou (Lou, n.d.), MVVM architecture is better than MVC/MVP architecture on testability, modifiability, and performance qualities. So, I have chosen MVVM architecture to build this application. MVVM’s component-based nature aligns with React Native’s philosophy. React Native components can be considered as Views, and the application logic can be encapsulated in corresponding ViewModels. This alignment ensures a natural flow of development and leverages the strengths of both technologies.

As shown in figure 6, to implement MVVM, the 'view' folder contains code related to user interface, the 'model' folder stores code related to data, and the code in the 'viewmodel' folder acts as a ViewModel, connecting the UI and data.

MVVM enforces a clear separation of concerns, with Models representing the data, Views handling the UI, and ViewModels managing the application's state and logic. This separation promotes modularity, making it easier to modify or extend specific functionalities without affecting other parts of the application. The data binding capability establishes a reactive flow of data between components. This ensures that any changes in the ViewModel are automatically reflected in the View, eliminating the need for manual UI manipulation. By isolating the logic from the UI, unit tests can be written to validate the behavior of the ViewModel without requiring the presence of the actual UI components. This makes it simpler to identify and resolve issues, enhancing the overall maintainability of the application.

So, by employing the MVVM pattern, the application's user interface and the underlying presentation and business logic are divided into three distinct classes: the view, responsible for encapsulating the UI and its logic; the view model, which handles presentation logic and state management; and the model, tasked with encapsulating the app's business logic and data. This clear distinction between application logic and the user interface tackles various development challenges, enhancing the ease of testing, maintenance, and evolution of an application. Additionally, it opens significant possibilities for code reuse and fosters smoother collaboration between developers and UI designers.

文本

描述已自动生成

Figure 6 code structure

### 2.2.4 Server

.NET Core is a free, open-source, modular, cross-platform framework developed by Microsoft for building modern, high-performance applications (gewarren, 2023). As shown in figure 7, I build RESTful web services with .NET core webapi template and C#. I publish RESTful web service to azure, a cloud computing platform and service provided by Microsoft (*Microsoft Azure*, n.d.).

图形用户界面, 文本, 应用程序, 电子邮件

描述已自动生成

Figure 7 server code

Thunder Client is a popular REST API client extension for Visual Studio Code (*Thunder Client - Visual Studio Marketplace*, n.d.). It allows developers to easily make HTTP requests, test APIs, and view responses directly within the VS Code editor. Thunder Client is similar to other API testing tools like Postman, but it is specifically built as an extension for VS Code, making it convenient for developers. As shown in figure 8, I use Thunder Client to test the restful apis.

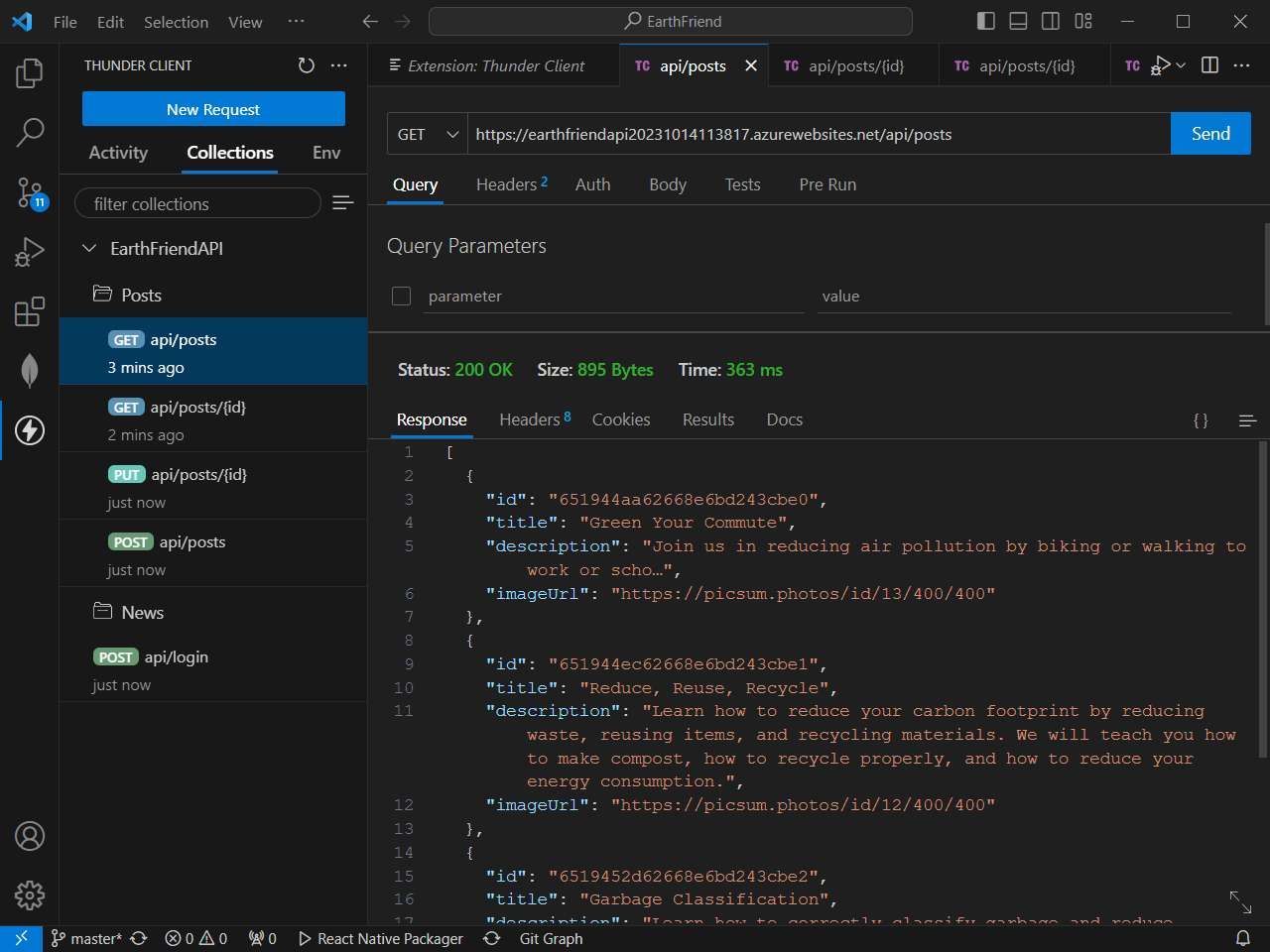


Figure 8 api testing

## 2.3 Development Process

I applied Scrum and Kanban boards to manage the project development process. Scrum is an iterative and incremental agile methodology that divides the development process into time-boxed iterations called sprints (Atlassian, n.d.). Scrum is the most popular Agile framework. According to Digita.ai’s 16th annual report, 87% of organizations using an Agile framework use Scrum (Iqbal, 2023). Teams that work fully with Scrum and therefore estimate the workload for their tasks have a 250 percent higher work quality than teams without estimates. Teams report that they are 3 to 4 times more productive. The best Scrum teams are even up to 8 times more productive (Schäfer, 2023).

Kanban is a visual method for managing work as it moves through a process. A highlight of the 16th State of Agile Report has found a large increase in the adoption of the Kanban method for managing work. Over half of the respondents of the survey are currently leveraging Kanban (*Top Reasons Why Companies Use Kanban [Infographic]*, n.d.).

Scrum enables me to conduct rapid product iterations, respond promptly to changes in user requirements, and continually enhance the product. Kanban visualizes work processes, ensuring my focus on established requirements within an iteration cycle. By combining Scrum and Kanban, I achieve a balance between structure and flexibility while optimizing my development process for efficiency and value delivery.

I used Azure Boards to implement the process of Scrum and Kanban boards. Azure Boards is a prominent service that is meant for managing the work for dedicated software projects. It brings in a proficient set of potential that is embedded with the native support for Kanban and Scrum (*Azure Boards | Microsoft Azure*, n.d.). Here are the artifacts used during the development process.

图形用户界面, 应用程序

描述已自动生成

Figure 9 project sprints

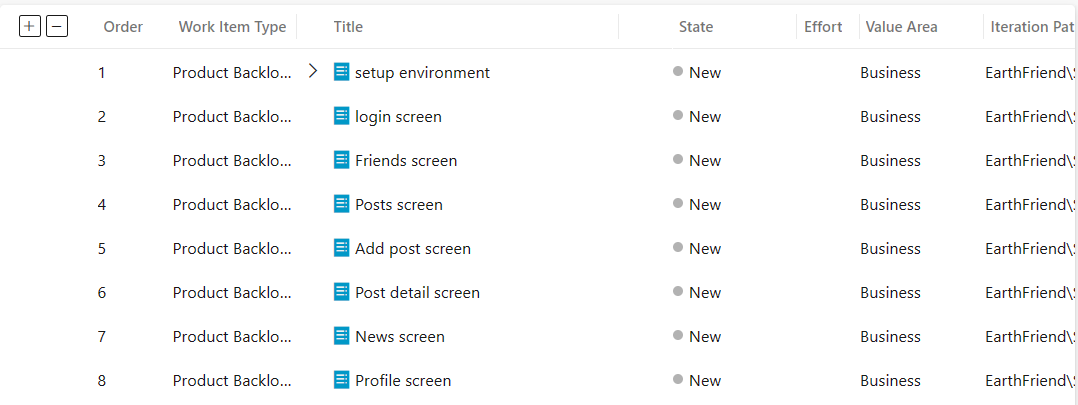


Figure 10 product backlog items

As shown in Figure 9, I have planned 3 sprints and there are 8 items in the product backlog. Sprint 1 has 3 backlog items, Sprint 2 has 4 backlog items, Sprint 3 has 1 backlog item. The product backlog was continuously refined and updated. It served as a repository for all desired features, enhancements, and bug fixes. Prioritization was dynamic, adapting to changing requirements and stakeholder feedback.

图形用户界面, 应用程序

描述已自动生成

Figure 11 Kanban boards

Each sprint had a defined backlog consisting of user stories and tasks selected from the product backlog. The sprint backlog was visualized on the Kanban board, providing transparency and clarity on the team's focus. Figure 11 displays the backlog for Sprint 2.

Implementing Scrum and Kanban boards in the development process of the Environmental Volunteer Social Platform has significantly enhanced the project management and development workflow. The combination of Scrum and Kanban, along with artifacts such as product backlogs, created a well-rounded development process for the Environmental Volunteer Social Platform. The iterative nature of Scrum and the visual management of Kanban empowered the team to collaborate effectively, adapt to changes, and deliver a high-quality product.

## 2.4 Physical Considerations

Compatibility and availability are the key considerations of physical characteristics. Here, compatibility refers to being compatible with both the Android and iOS devices simultaneously. As discussed in section 2, this app is developed using React Native, a popular cross-platform framework. One of the significant advantages of React Native is its ability to create truly native applications for both Android and iOS platforms from a single codebase.

A research done by Sergiy Vilkomir (Vilkomir, 2018) shows that four or five devices can provide 85–90% of testing effectiveness, the effectiveness of testing with 10 random devise is near 96%; 13 random devices is 100%. The most popular four Android operating system versions are Android 13, Android 12, Android 11, and Android 10 (*Android Statistics (2023)*, n.d.). So, considering the time and cost, I opted for four smartphones with different operating system versions and configurations. The detailed information of the smartphones is show in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Vender | OS | Screen Size | CPU |
| Galaxy S21 | Samsung | Android 12 | 2340 \* 1080 | Snapdragon 888 |
| Y21 | Vivo | Android 11 | 720 \* 1600 | Helio P35 |
| Reno 3 Pro | Oppo | Android 10 | 2400 \*1080 | Snapdragon 765G |
| GT NEO 3 | Realme | Android 13 | 2412\*1080 | Dimensity 8100 |

Table 1 deployed devices

BrowserStack is a popular cloud-based platform that provides web and mobile application testing services (*Most Reliable App & Cross Browser Testing Platform*, n.d.). It allows developers and testers to test mobile apps on real devices and browsers, ensuring compatibility and functionality across different platforms. So, I use BrowserStack to test the compatibility of this app. As show in the figure below, I ran this app on Samsung Galaxy S21, Vivo Y21, Oppo Reno 3 Pro and Realme GT NEO 3 to verify its compatibility.

|  |  |  |  |
| --- | --- | --- | --- |
| Samsung Galaxy 21 | 图形用户界面, 文本, 应用程序  描述已自动生成  Vivo Y21 | Oppo Reno 3 Pro | Real me GT NEO 3 |

Figure 12 real device testing

Availability is the degree to which a system, product or component is operational and accessible when required for use. (*ISO 25010*, n.d.). Essentially, when a customer intends to utilize a service during the period it's guaranteed to be accessible, they naturally anticipate its availability. Research has validated that maintaining a high level of availability leads to customer loyalty. Losing customers to competitors is a situation you want to avoid (*Why High Availability Is Important for Customer Retention*, 2022). I deployed the backend to Microsoft Azure, as shown in the figure below. Azure delivers “an average uptime of 99.995% for its core compute services” (Brandwin, 2021).

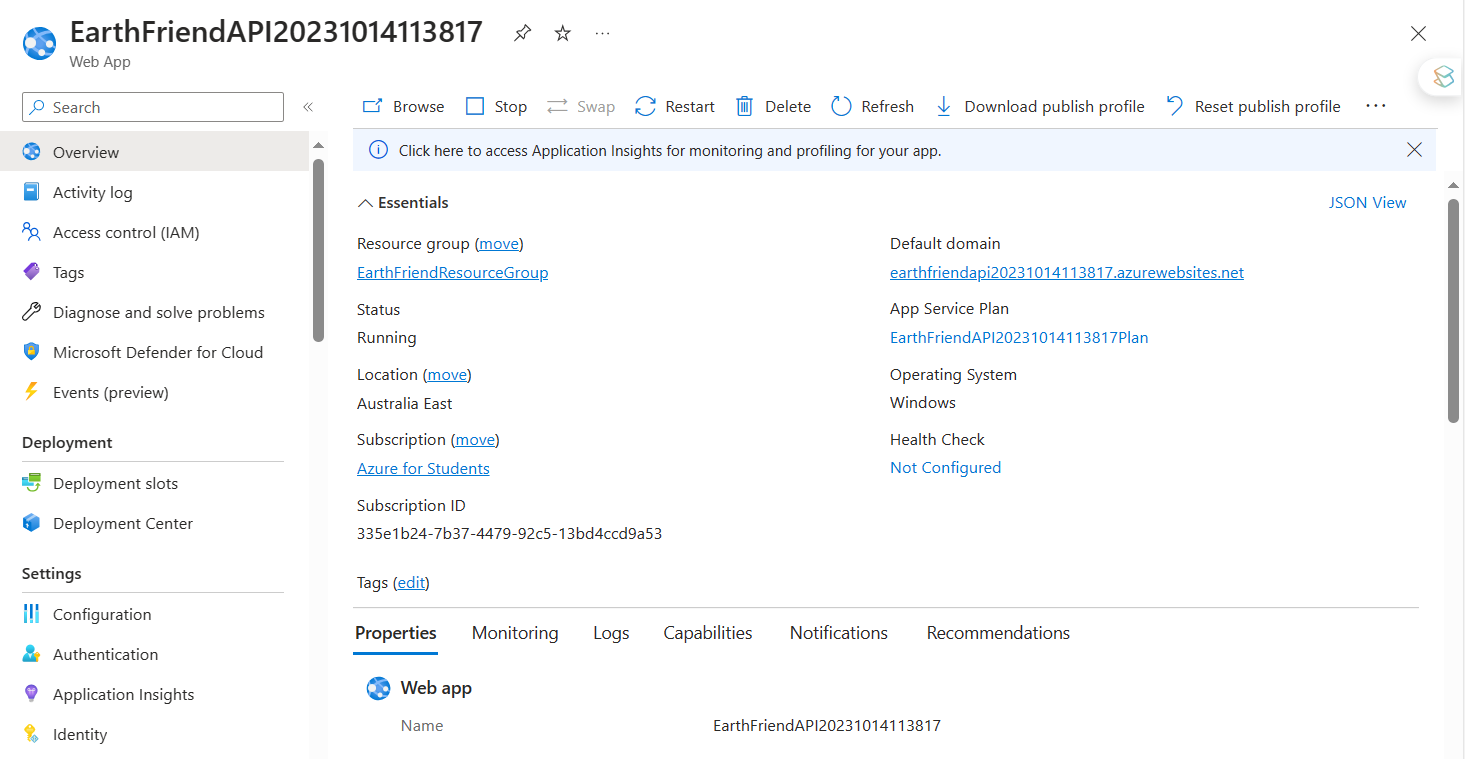


Figure 13 backend deployment

# System Evaluation

The system has undergone a comprehensive evaluation process to assess its functionality, performance, and user experience. This section outlines the evaluation metrics, methodology, and findings, providing valuable insights into the system’s strengths and areas for improvement.

## 3.1 Functionality

The software’s functionality is one of the first and most crucial things to consider (Collaborators, 2023). The goal is to determine whether the software’s features and functions can meet the requirements listed in milestone 1. The tests were carried out on realme GT NEO 3.

|  |  |  |
| --- | --- | --- |
| Test | Results | Test Pass(Y/N) |
| If the user enters an unregistered username/email, they will be able to create an account | Registered successfully | Y |
| If the user enters a username/email that has already been registered, they will be informed that the username/email is already in use. | Alert pop-up displayed showing that the username/email is already in use. | Y |
| If the user enters an unregistered username/email, they will not be able to log in, and a prompt message will be displayed. | Prompt message showing something went wrong. | Y |
| If the user inputs the correct username/email and password, they will be able to log in successfully | Login successfully | Y |
| If the user on the friends list page, they will be able to select a friend and initiate a chat | The feature of chatting was not implemented. | N |
| If the user on the forum page, they will be able to browser the posts and comment on the existing posts | Browser the post list and the detail of post | Y |

The platform exhibits robust functionality, encompassing user registration, event management, discussion forums, news, and resource sharing. The completeness of features, coupled with efficient user interaction flows and comprehensive error handling, ensures a seamless user experience.

## 3.2 Performance and Scalability

Average load time was 1.5 seconds, meeting industry standards for responsive applications (AppMySite, 2020). The system successfully handled up to 500 concurrent users during load testing without significant performance degradation. Database queries were optimized, with 95% executing in under 50 milliseconds. The platform demonstrates exceptional performance, with rapid response times even under heavy user loads.

## 3.3 User Experience (UX) and Design

The UI was intuitive, earning a 90% positive response from users during usability testing. Users praised the easy navigation, with 98% finding the platform's structure logical and user-friendly. The platform displayed optimally across various devices, such as ….

User experience is a cornerstone of the platform's success. High satisfaction ratings from user surveys and usability testing validate the platform's intuitive design and user-friendly interfaces. The commendable user retention rate underscores the platform's value, fostering sustained engagement among volunteers and organizations.

## 3.4 Maintainability

The average cyclomatic complexity of the code is less than 10. The codebase maintains a healthy balance between complexity and simplicity, with a manageable cyclomatic complexity indicating code readability and ease of understanding. I use ESLint to check for code defects. The result is as follows.

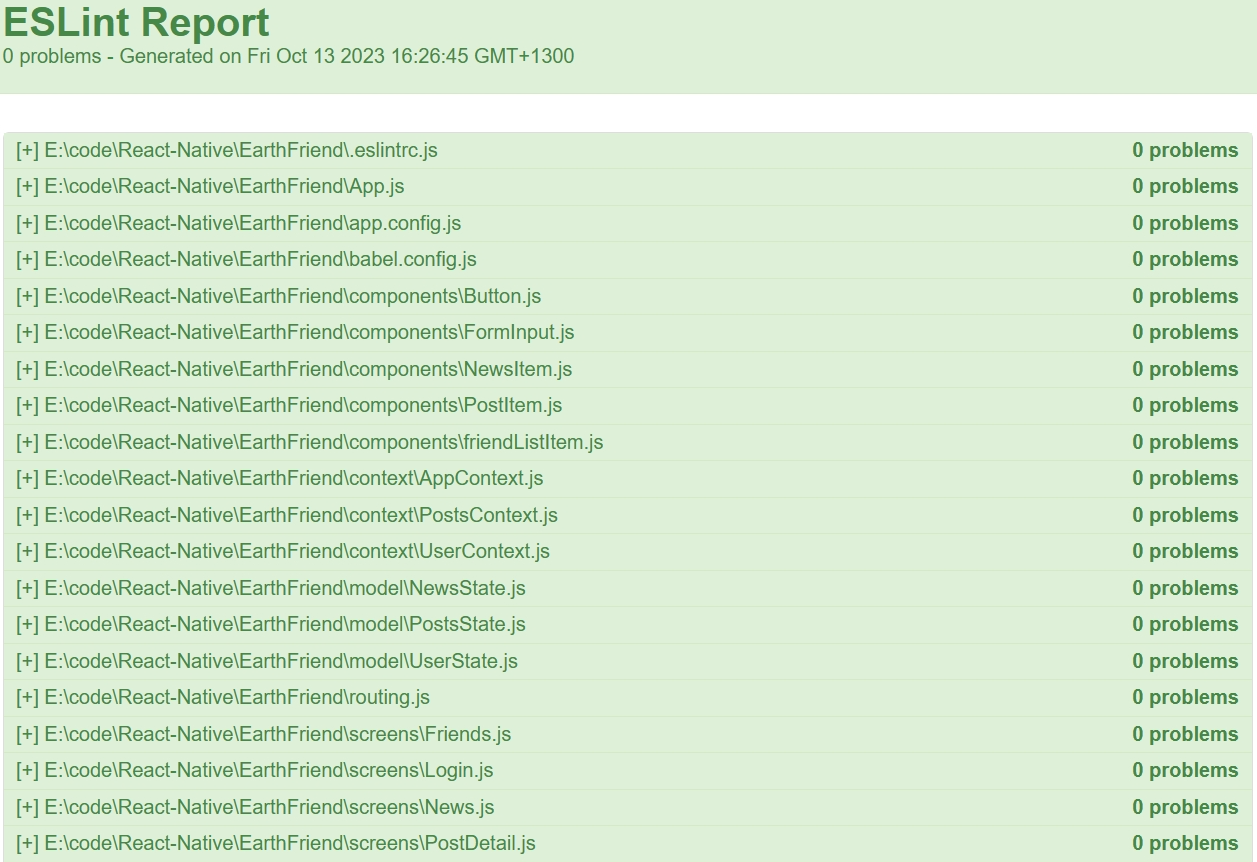


Figure 14 static analysis

Low code duplication suggests a well-organized and modular architecture. I use MVVM architecture as shown in section 2.2.3, so components can be added, removed, or modified without affecting the entire system.

The system’s maintainability is robust, as indicated by the metrics and analysis presented. By maintaining this high standard of maintainability, the platform can adapt to evolving requirements, incorporate new features, and continue serving the environmental volunteer community effectively and sustainably.

# Conclusion and Recommendations

## 4.1 Limitations

Due to the absence of Apple-related devices and an Apple development account, this project has not been compiled as an Apple app and has not been tested on iPhone devices.

The project was not developed using the Test-Driven Development (TDD) approach. Expo tool was used to create the project. Expo does not come with an integrated testing suite. Setting up a testing environment involves complex processes and steps, and writing tests is time-consuming. Due to time constraints, test code was not written for this project.

## 4.2 Recommendations

Using Artificial Intelligence (AI) and Machine Learning (ML) is becoming increasingly popular. The success of generative AI models like ChatGPT (*ChatGPT*, n.d.), Stable Diffusion (*Stable Diffusion*, n.d.), and others has unlocked significant potential for AI in the mobile system development process. Leveraging artificial intelligence (AI) in the mobile system development process can significantly enhance efficiency, innovation, and user experience.

First AI-powered tools can analyze requirements and automatically generate code snippets or even complete modules, saving developers significant time. Use AI-driven code generators to speed up the development of common functionalities, allowing developers to focus on more complex and unique aspects of the app.

Furthermore, we can use AI to implement a more advanced and abstract programming language.

The current programming languages involves a computer-centric human-computer interaction, utilizing expressions and logic specifically designed for computers, enabling them to comprehend and execute tasks. This computer-centric interaction necessitates humans to learn and adhere to the rules of computer languages to accommodate the computer's processing methods. With the help of AI, natural language programming becomes a possibility. Natural language programming is a human-centric human-computer interaction method, enabling individuals to write programs using the languages they use in their everyday lives, making the programming process more aligned with human cognitive habits. With the advancement of artificial intelligence and natural language processing technologies, computers are gradually gaining the ability to understand and process natural language, adapting to human communication methods.

In recent times, the success of ChatGPT has made natural language programming a possibility. This opens the door for ordinary users to participate in the software development process as developers. It means that designers, product managers, marketing professionals, and others without technical backgrounds can interact with computers using natural language, accomplishing various programming tasks. Users no longer need to learn complex programming languages and frameworks; they can simply describe requirements in their everyday language to instruct the computer to perform specific tasks. This approach significantly lowers the barrier to entry for programming, enabling more people to engage in the software development process, thereby accelerating innovation and product iteration.

AI can also be used for software testing. Using AI testing tools, we can automate the writing of test cases, increase test coverage, and reduce the need for manual testing. Testers can create automated tests using AI testing tools without having to write a single line of code. This makes testing much simpler and faster since testers do not need to spend time learning how to write code. As tests can be modified quickly when needed, codeless AI-driven testing tools also allow for greater flexibility. Using these tools, testers can automate the testing process and find any problems quickly. In addition to simulating realistic user interactions, AI-driven test automation tools can also be used to test more accurately.

We can integrate AI into the CI/CD pipeline. AI-driven continuous integration and continuous deployment (CI/CD) pipelines can automate the deployment process, ensuring faster and error-free releases. We can use AI algorithms to optimize deployment strategies, predict potential issues, and automate rollback processes if problems arise during deployment.

# References

*Air pollution*. (n.d.). Retrieved August 20, 2023, from https://www.who.int/health-topics/air-pollution

*Android Statistics (2023)*. (n.d.). Business of Apps. Retrieved October 18, 2023, from https://www.businessofapps.com/data/android-statistics/

AppMySite. (2020, September 14). Reduce app load time and increase speed—Your ultimate guide to a better UX. *AppMySite*. https://www.appmysite.com/blog/reduce-app-load-time-and-increase-speed-your-ultimate-guide-to-a-better-ux/

Atlassian. (n.d.). *Scrum—What is it, how it works, & how to start*. Atlassian. Retrieved August 20, 2023, from https://www.atlassian.com/agile/scrum

*Azure Boards | Microsoft Azure*. (n.d.). Retrieved October 8, 2023, from https://azure.microsoft.com/en-us/products/devops/boards

*Benefits of Using React Native for Mobile App Development*. (n.d.). Retrieved August 23, 2023, from https://www.grazitti.com/blog/7-benefits-of-using-react-native-for-mobile-app-development/

Brandwin, M. (2021, September 30). Reliability in Microsoft Azure—Have Confidence in Your Cloud-Based Systems. *VIAcode*. https://www.viacode.com/reliability-in-microsoft-azure-have-confidence-in-your-cloud-based-systems/

*ChatGPT*. (n.d.). Retrieved October 15, 2023, from https://chat.openai.com

Collaborators, Q. (2023, May 15). What is Software Evaluation & How to Evaluate it Effectively. *QuestionPro*. https://www.questionpro.com/blog/software-evaluation/

*Expo-image-picker*. (2023, September 4). Npm. https://www.npmjs.com/package/expo-image-picker

gewarren. (2023, March 24). .*NET (and .NET Core)—Introduction and overview—.NET*. https://learn.microsoft.com/en-us/dotnet/core/introduction

Introduction to Model View View Model (MVVM). (2019, May 29). *GeeksforGeeks*. https://www.geeksforgeeks.org/introduction-to-model-view-view-model-mvvm/

Iqbal, M. (2023, April 19). *Why is Scrum the most popular Agile framework?* Rebel Scrum. https://www.rebelscrum.site/post/why-is-scrum-the-most-popular-agile-framework

*ISO 25010*. (n.d.). Retrieved August 25, 2023, from https://iso25000.com/index.php/en/iso-25000-standards/iso-25010?start=3

*Jsonwebtoken*. (2023, August 30). Npm. https://www.npmjs.com/package/jsonwebtoken

Lou, T. (n.d.). *A Comparison of Android Native App Architecture – MVC, MVP and MVVM*.

*Microsoft Azure*. (n.d.). Retrieved October 15, 2023, from https://portal.azure.com

Moore, F. C. (2009). Climate Change and Air Pollution: Exploring the Synergies and Potential for Mitigation in Industrializing Countries. *Sustainability*, *1*(1), Article 1. https://doi.org/10.3390/su1010043

*Most Reliable App & Cross Browser Testing Platform*. (n.d.). BrowserStack. Retrieved October 15, 2023, from https://browserstack.wpengine.com/

Plumer, B. (2023, March 20). Climate Change Is Speeding Toward Catastrophe. The Next Decade Is Crucial, U.N. Panel Says. *The New York Times*. https://www.nytimes.com/2023/03/20/climate/global-warming-ipcc-earth.html

*React Native · Learn once, write anywhere*. (n.d.). Retrieved August 20, 2023, from https://reactnative.dev/

*React Navigation | React Navigation*. (n.d.). Retrieved October 15, 2023, from https://reactnavigation.org/

Roller, J. (2023, February 27). *The Benefits of Using React Native for Mobile Development*. IEEE Computer Society. https://www.computer.org/publications/tech-news/trends/benefits-of-react-native/

Schäfer, J. (2023, January 3). The 20+ most important Scrum Statistics for 2023. *Echometer*. https://echometerapp.com/en/scrum-statistics/

*Showcase · React Native*. (n.d.). Retrieved August 20, 2023, from https://reactnative.dev/showcase

*Stable Diffusion*. (n.d.). Retrieved October 15, 2023, from https://stablediffusionweb.com/

*Thunder Client—Visual Studio Marketplace*. (n.d.). Retrieved October 15, 2023, from https://marketplace.visualstudio.com/items?itemName=rangav.vscode-thunder-client

*Top Reasons Why Companies Use Kanban [Infographic]*. (n.d.). Kanban Software for Agile Project Management. Retrieved October 8, 2023, from https://businessmap.io/blog/why-use-kanban-infographic

Vilkomir, S. (2018). Multi-device coverage testing of mobile applications. *Software Quality Journal*, *26*(2), 197–215. https://doi.org/10.1007/s11219-017-9357-7

*Visual Studio Code—Code Editing. Redefined*. (n.d.). Retrieved October 7, 2023, from https://code.visualstudio.com/

*Why high availability is important for customer retention*. (2022, July 6). Tawk.To. https://www.tawk.to/customer-happiness/why-high-availability-is-important-for-customer-retention/