

PROG7311 PoE Part 1

Eben Mwema, ST10091324



BCA3

Group 1

VC Waterfall

Declaration

I Eben Nkulu Mwema declare that:

|  |  |
| --- | --- |
|  | SIGN |
| I have read the assessment rules provided in this declaration | E.N.M |
| This assessment is my own work. | E.N.M |
| I have not copied any other student’s work in this assessment. | E.N.M |
| I have not uploaded the assessment question to any website or App offering assessment assistance. | E.N.M |
| I have not downloaded my assessment response from a website. | E.N.M |
| I have not used any AI tool without reviewing, re-writing, and re-working this information, and referencing any AI tools in my work. | E.N.M |
| I have not shared this assessment with any other student. | E.N.M |
| I have not presented the work of published sources as my own work. | E.N.M |
| I have correctly cited all my sources of information. | E.N.M |
| My referencing is technically correct, consistent, and congruent. | E.N.M |
| I have acted in an academically honest way in this assessment. | E.N.M |

Table of Contents

[**1. INTRODUCTION** 2](#_Toc170212943)

[1.1 Background 2](#_Toc170212944)

[1.2 Purpose and Goal 2](#_Toc170212945)

[**2. NON-FUNCTIONAL REQUIREMENTS (NFRS)** 2](#_Toc170212946)

[2.1 Scalability 2](#_Toc170212947)

[2.2 Security 2](#_Toc170212948)

[2.3 Usability 3](#_Toc170212949)

[2.4 Performance 3](#_Toc170212950)

[2.5 Practical Strategies of NFRs 3](#_Toc170212951)

[2.6 Impact of NFRs 4](#_Toc170212952)

[**3. DESIGN AND ARCHITECTURE PATTERNS** 5](#_Toc170212953)

[Importance of design and architecture patterns 5](#_Toc170212954)

[3.1 Integration of Design and Architecture Patterns 6](#_Toc170212955)

[**4. CONCLUSION** 9](#_Toc170212956)

[REFERENCE LIST 10](#_Toc170212957)

Table of Figures

[Figure 1: Agile Model 6](#_Toc170377182)

[Figure 2: Observer Pattern Diagram 7](#_Toc170377183)

[Figure 3: MSA Pattern Diagram (DevTeam.Space, [n.d.]) 9](#_Toc170377184)

# **1. INTRODUCTION**

## 1.1 Background

The IIE (2024) currently, as it stands the agricultural sector and green energy technology providers do not have a unified platform where they can communicate and collaborate on projects. We have the potential to fill this gap in the market by using our personnel and the expertise they possess.

## 1.2 Purpose and Goal

The purpose of this proposal is to get approval to commence with designing, developing, and releasing our goal which is listed below.

The IIE (2024) our goal is to develop an innovative web platform that bridges the gap between the agricultural sector and green energy technology providers by offering them a platform to collaborate, share resources, and innovate in the realms of sustainable agriculture and renewable energy’.

# **2. NON-FUNCTIONAL REQUIREMENTS (NFRS)**

Rome (2023) NFRs are requirements that describe how a system should perform a specific task to meet stakeholder or business needs.

## 2.1 Scalability

* Hayes (2024) Scalability refers to adding more resources to your system/s, either through renting or permanent purchases. The system should be able to handle large spikes of concurrent users during peak times (at least 20,000 users). It is important to have scaling measures in place because if the system receives more requests than it can handle, it may result in the system crashing entirely or long wait times for requests to get fulfilled

## 2.2 Security

* The system should encrypt the financial details such as the card number of the users during the marketplace checkout/payment process.
* The system should use a hashing algorithm such as sha256 and salting method to store user passwords.
* The system should encrypt messages between users using End-to-End encryption (EEE).
* The system should validate and verify that email addresses provided by the user during the sign-up process are real.

## 2.3 Usability

* The system should allow users to seamlessly compare products on the marketplace.
* The system should allow users to seamlessly leave reviews of technologies on the marketplace.
* The system should allow users to seamlessly connect with green tech providers using the green tech providers username.
* The system should allow users to seamlessly navigate from one forum/discuss board to another.

## 2.4 Performance

* The system should be able to retrieve gigabytes of data from various databases within 10 seconds without it affecting the user experience.
* The system should be able to seamlessly handle thousands of data sharing requests per second from a 1000 or more users simultaneously without it affecting the user experience.
* The system should seamlessly allow 2 to 50 farmers and energy experts to collaborate on joint projects.

## 2.5 Practical Strategies of NFRs

Alexandra (2024) the software development lifecycle (SDLC) is a set of guidelines created to help businesses produce high quality products with a realistic budget cost using 6 main stages:

* **Planning** - Alexandra (2024) this stage will focus on us determining how much and how it will take to develop these NFRs as well as what resources we will be required.
* **Designing** - Alexandra (2024) this stage will focus on us building prototypes of these NFRs and getting feedback on these prototypes from our stakeholders.
* **Implementation** - Alexandra (2024) this stage will focus on us actually building these NFRs into our application using code.
* **Testing** - Alexandra (2024) this stage will focus on us performing various testing on these NFRs we implemented into our application to ensure they integrate and function properly on its own and with other components of our application.
* **Deployment** - Alexandra (2024) this stage will focus on us releasing the finished implementation of our NFRs to the public for them to use.
* **Maintenance** – Alexandra (2024) this stage will focus on us fixing issues with our NFRs and implementing the feedback received from our customers about our NFRs.

## 2.6 Impact of NFRs

Alexandra (2024) argues that to implement and release successful and effective NFRs, we need to involve our users and make changes to our NFRs with relative ease. The **Agile** methodology is an approach that allows us to do this. Alexandra (2024) argues that by making use of an agile model, we can:

* Incorporate our **customers** into the planning and development of our NFRs to ensure their satisfaction and get their feedback. This would involve
* We can iterate through designs and implementation of those designs with **relative ease**.
* We can reduce the overall development time of our NFRs by focusing on smaller weekly objectives (**sprints**).
* **Constantly test** the various stages of our SDLC to deliver a high-quality product.
* **Adapt** relatively quickly to customer feedback and emergent technologies.

A diagram of a diagram

Description automatically generated with medium confidence

Figure 1: Agile Model

# **3. DESIGN AND ARCHITECTURE PATTERNS**

It is important to evaluate and select design and architecture patterns that best suit your project before commencing the development process.

## Importance of design and architecture patterns

Mesropyan (2023) argues that making use of design and architectural patterns will:

* Improve **production and efficiency** in the software development process by reusing proven solutions and eliminating redundant code.
* Improve **scalability** by making use of flexible structures that can be adjusted to new emergent user or business requirements.
* Improve **maintainability**. By making code more modular, developers can alter specific code module without having to alter the entire application.
* Establish a common programming language for developers to use which will improve communication and understanding between developers and therefore enable **better collaboration**.

## 3.1 Integration of Design and Architecture Patterns

We will make use of the **Observer design pattern** for the interactive forum/discussion boards. Using this pattern, will enable us to allow users to subscribe to specific forums/discussions boards and get notified when new content has been uploaded to these forums/discussions boards (subscribers). It will also enable us to encapsulate the core components.

A diagram of a diagram

Description automatically generated

Figure 2: Observer Pattern Diagram

Since the Agri-Energy Connect Platform will offer lots of services to its users, we have decided to go with the **Microservices Architecture pattern** (MSA) throughout the application. The MSA pattern is an architectural pattern that focuses on breaking down a complex application into various smaller broken-down services (microservices) that are either independent of each other or weakly dependent on each other.

This will enable the developers to make changes to specific parts of the application without it affecting a few or several parts of the rest of the application (**Maintainability**). If one of our microservices goes down, the rest of the application can still function without issue (**Resiliency**).

If there is an increase in usage for one of our microservices we can easily increase the load our systems can handle for that microservice (**Scalability**) without having to scale our entire application, for example more users signing up and using our education and training resource hub.

We can also for example, break down the 4 key features of our application into microservices. The IIE (2024) the 4 key features consist of the:

* Sustainable Farming Hub
* Green Energy Marketplace
* Educational and training resources
* Project Collaboration and funding opportunities

**A diagram of a computer hardware company

Description automatically generated with medium confidence**

Figure 3: MSA Pattern Diagram (DevTeam.Space, [n.d.])

# **4. CONCLUSION**

With the approval of the biding community and the right budget, we can implement the NFRs listed above and make use of the chosen design and architecture patterns to successfully deliver a satisfactory product to our clients.

# REFERENCE LIST

Alexandra. 2024. What Is SDLC? Understand the Software Development Life Cycle, Stackify, 28 February 2024. [Blog]. Available at: <https://stackify.com/what-is-sdlc/> [Accessed 16 April 2024].

Maguire, J. n.d. Microservices Architecture Diagram Examples, DevTeam.Space, [n.d.]. [Blog]. Available at: <https://www.devteam.space/blog/microservice-architecture-examples-and-diagram/> [Accessed 27 June 2024].

Mesropyan, E. 2023. Introduction to Design Patterns in Software Development, Stackify, 1 August 2023. [Blog]. Available at: <https://stackify.com/introduction-to-design-patterns-in-software-development/> [Accessed 15 April 2024].

Rome, P. 2023. What are Non Functional Requirements — With Examples, Perforce, 24 March 2023. [Blog]. Available at: <https://www.perforce.com/blog/alm/what-are-non-functional-requirements-examples> [Accessed 15 April 2024].

The IIE. 2024. Programming 3A [PROG7311 Module Manual]. The Independent Institute of Education: Unpublished.

The IIE. 2024. Programming 3A [PROG7311 Portfolio of Evidence]. The Independent Institute of Education: Unpublished.

Hayes, A. 2024. Scalability: What a Scalable Company Is and Examples, Investopedia, 22 June 2024. [Blog]. Available at: <https://www.investopedia.com/terms/s/scalability.asp> [Accessed 27 June 2024].