# PROG7311 – POE Part 1

ST10043367
Reece Declan Cunningham

## Table of Contents



## Table of Figures

# Academic Honesty Declaration:

**ACADEMIC HONESTY DECLARATION**

Please complete the Academic Honesty Declaration below.

Please note that your assessment will not be marked, and you will receive 0% if you have not completed ALL aspects of this declaration.

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

*Figure 1 - Academic Honesty Declaration (signed electronically)*

# Part 1: Report about Requirements and Design Patterns

## Introduction

This proposal introduces the Agri-Energy Connect Platform in response to the urgent demand for sustainable agriculture practices and the integration of green energy solutions in South Africa. The goal of this project is to create a digital ecosystem that will enable farmers, specialists in green energy, and enthusiasts to work together to innovate in the areas of renewable energy and sustainable agriculture. I understand the importance of this initiative in tackling real-world issues as an aspiring IT professional. Throughout this report, I will analyse critical non-functional requirements and explore the role of design and architecture patterns in shaping the platform's development. The aim is to provide a clear understanding of our approach and the value it brings to the agricultural and energy sectors.

## 1. Analysis of Non-Functional Requirements

### Scalability:

Implementing scalability is crucial to accommodate potential growth in user base and data volume over time.

- **Modular and Distributed Architecture**: Design the platform using independent microservices to allow for horizontal scaling. This approach ensures that individual components can scale independently based on demand (Couchbase, 2023).
- **Cloud-Based Infrastructure**: Utilize cloud services for dynamic provisioning of computational resources. Cloud platforms like AWS or Azure offer scalable infrastructure, allowing the platform to adapt to changing workloads (Couchbase, 2023).

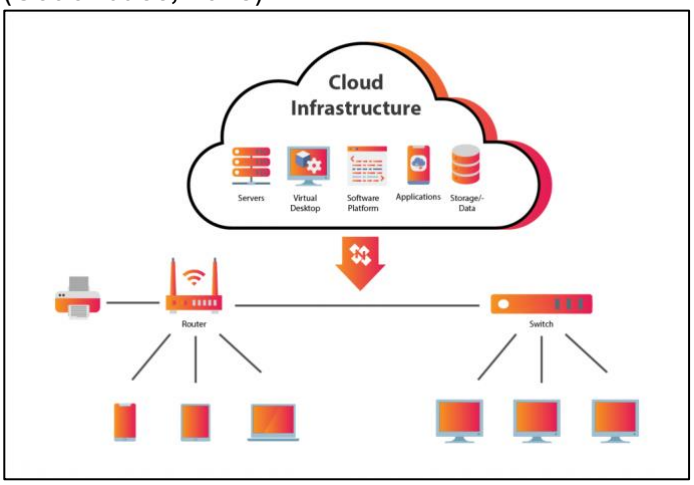


*Figure 2 - Cloud infrastructure (G., 2022)*

- **Load Balancing Mechanisms**: Implement load balancers to evenly distribute incoming traffic across multiple servers. This ensures optimal resource

utilization and prevents performance degradation during peak usage periods (Couchbase, 2023).
- **Scalable Database Solutions**: Employ distributed databases such as MongoDB or Cassandra for horizontal scaling of data storage. This ensures efficient handling of large datasets without compromising performance (Couchbase, 2023).

To implement scalability, we'll design the platform with independent microservices for horizontal scaling. Cloud services like AWS will provide flexible infrastructure to adapt to changing demands. Load balancers will evenly distribute traffic across servers, optimizing resource use. Scalable databases like MongoDB will efficiently handle large datasets without performance issues. This approach ensures the platform can grow seamlessly as user base and data volume increase over time (Couchbase, 2023) (Kazmi, 2023).

Implementing scalability significantly influences how we develop the software. It guides us in designing the architecture and deciding where to allocate resources. We focus on creating a flexible structure with independent parts that can grow as needed. Using cloud services helps us adapt to changes in demand smoothly. We also implement tools to balance the load across servers and manage data efficiently. These steps ensure our platform can handle more users and data over time, making it reliable and scalable (Couchbase, 2023) (Kazmi, 2023).

## Security:

To protect against vulnerabilities, such as unauthorised access and alteration, application security involves developing, integrating, and testing security measures within applications. Strong security protocols are essential for protecting sensitive user data and financial transactions (vmware, 2024).

- **Role-Based Access Control (RBAC)**: Enforce RBAC to restrict access based on user roles. This prevents unauthorized users from accessing sensitive information and functionalities (Gillis, 2023).
- **Encryption**: Apply encryption techniques to data transmission and storage to safeguard against unauthorized access. Implement HTTPS protocol for secure communication and encryption algorithms like AES for data at rest (Sheldon, 2024).
- **Regular Security Audits**: Conduct periodic security audits and penetration testing to identify and remediate vulnerabilities proactively. This ensures compliance with industry standards and regulations (Sheldon, 2024).
- **Data Privacy Compliance**: Ensure compliance with data privacy regulations such as GDPR or HIPAA to protect user privacy and build trust with stakeholders (Paloalto, 2021).


*Figure 3 - Security of financial data (Capital, 2019)*

This project will implement encryption mechanisms for data storage and transmission, as well as enforce role-based access control (RBAC) to guarantee strong security in the Agri-Energy Connect Platform. The implementation of routine security audits and compliance checks is intended to detect vulnerabilities and guarantee compliance with industry standards such as GDPR and HIPAA. By taking these precautions, users and stakeholders will feel more confident and trusting as sensitive user data and financial transactions are protected (Gillis, 2023) (Sheldon, 2024) (Paloalto, 2021).

Implementing comprehensive safety precautions in place guides planning and execution, which has a significant impact on our software development approach. It demands that encryption methods and access control systems be carefully considered from the beginning. The first priority throughout the planning stage is creating an encrypted architecture with role-based access control to prevent unwanted access to private data. To ensure encrypted data transfer and storage, encryption techniques like HTTPS and AES must be implemented during execution. To find and fix vulnerabilities early on, regular security audits and compliance checks are incorporated into the development lifecycle. By assuring adherence to industry standards and data protection rules, these actions promote trust among stakeholders and users. (Gillis, 2023) (Sheldon, 2024).

## Usability:

Enhancing usability is essential to promote user adoption and satisfaction with the platform (Team, 2022).

- **Intuitive User Interfaces**: Design intuitive user interfaces with clear navigation and user-friendly features. Incorporate user-centric design principles to streamline user interactions and improve usability (Team, 2022).
- **User Feedback Mechanisms**: Implement feedback mechanisms such as surveys and usability testing to gather insights into user preferences and pain points. This enables iterative improvements to usability based on user feedback (Team, 2022).
- **Comprehensive Documentation**: Provide comprehensive documentation and user guides to assist users in understanding platform functionalities. Clear

instructions and tutorials help users navigate the platform effectively and maximize its utility (Team, 2022).

- **Accessibility Features**: Ensure accessibility by incorporating features such as screen reader compatibility and adjustable font sizes. This ensures inclusivity and enables users with disabilities to access and use the platform (Team, 2022).

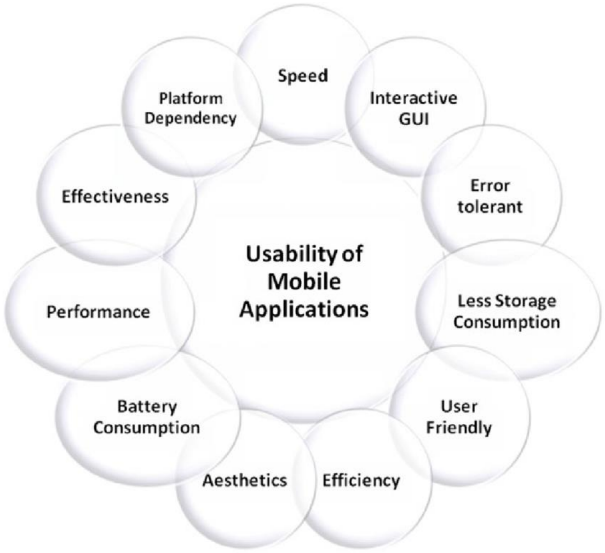


*Figure 4 - Usability of applications (Nagpal, 2018)*

To enhance usability, there will be a focus on creating intuitive interfaces, integrating user feedback, providing comprehensive documentation and incorporating accessibility features. Having thorough documentation will help users grasp the functions of the platform and accessible features guarantee that users with impairments are included. These strategies will increase user satisfaction and adoption, improving the platform's overall usefulness. (Team, 2022).

Prioritizing usability significantly influences our software development approach, shaping both planning and execution stages. During planning, resources are allocated to design intuitive interfaces and incorporate user feedback mechanisms. Execution requires paying close attention to every last detail when implementing comprehensive documentation and accessibility features into effect. The objective is to increase user adoption and overall customer satisfaction by prioritising usability throughout the development lifecycle, which will increase the platform's overall success (Team, 2022).

## Performance:

Optimizing performance is critical to deliver a responsive and efficient user experience (Balachandran, 2013).

- **Caching Mechanisms**: Implement caching mechanisms to reduce response times by serving frequently accessed data from memory. This enhances system responsiveness and improves overall performance (Balachandran, 2013).
- **Asynchronous Processing**: Utilize asynchronous processing techniques such as message queues or event-driven architecture to handle background tasks

efficiently. This prevents performance bottlenecks and ensures smooth system operation (Balachandran, 2013).

- **Performance Monitoring Tools**: Deploy performance monitoring tools like New Relic or Datadog to track system metrics in real-time. This enables proactive identification and resolution of performance issues, maintaining optimal system performance (Balachandran, 2013).
- **Optimized Database Queries**: Optimize database queries through indexing, query optimization, and database schema design. This improves database performance and enhances overall system responsiveness (Balachandran, 2013).

To optimize performance, we'll focus on caching, asynchronous processing, monitoring, and database optimization. Implementing caching reduces response times by serving frequently accessed data from memory. Asynchronous processing prevents bottlenecks by handling background tasks efficiently. Deploying monitoring tools enables real-time tracking of system metrics for proactive issue resolution. Optimizing database queries improves overall system responsiveness. These strategies ensure a responsive and efficient user experience for the Agri-Energy Connect Platform (Balachandran, 2013).

Prioritizing performance optimization shapes our software development approach by guiding planning and execution. During planning, resources are allocated to implement caching, asynchronous processing, and database optimization. Execution involves implementing performance monitoring tools for real-time tracking and proactive issue resolution. By focusing on performance throughout development, we aim to deliver a responsive user experience for the Agri-Energy Connect Platform (Balachandran, 2013).

## 2. Role of Design and Architecture Patterns

### Relevance of Design and Architecture Patterns:

Design patterns and architecture patterns are highly relevant in the context of the Agri-Energy Connect Platform project. These patterns provide proven solutions to common design and architectural challenges. They offer a structured approach to building scalable, maintainable and efficient software systems. In the case of Agri-Energy Connect, where the platform aims to integrate various features and functionalities while ensuring performance, security and usability, design and architecture patterns play a crucial role in achieving these objectives (Geeks, 2021).

### Integration of Design and Architecture Patterns:

For the Agri-Energy Connect Platform, the integration of design and architecture patterns will ensure scalability, maintainability and efficiency. Utilizing the Model-View-Controller (MVC) pattern for modular development, Microservices Architecture for flexibility and resilience and the Repository Pattern for simplified data access will streamline development, enhance code reusability and facilitate seamless integration of features:

1. **Model-View-Controller (MVC) Pattern:** Utilize MVC pattern to separate concerns and facilitate modular development. The model represents the data and business logic, the view handles user interface components and the controller manages user input and application flow. This separation enhances maintainability and scalability (Hernandez, 2021).
2. **Microservices Architecture:** Implement microservices architecture to decompose the platform into smaller, independent services. Each service focuses on a specific function such as sustainable farming hub, green energy marketplace or educational resources. This approach enables flexibility, scalability and resilience (Tech, 2024).
3. **Repository Pattern:** Employ repository pattern to abstract data access and storage logic. Repositories act as intermediaries between the application and data source, providing a consistent interface for data operations. This promotes code reusability and simplifies database interactions (Hernandez, 2021).

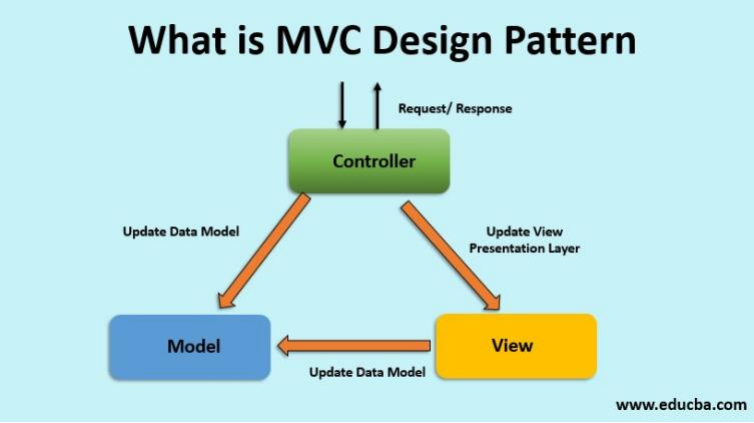


*Figure 5 - Model View Controller (MVC) Pattern (Pedamkar, 2023)*

Therefore, the integration of design and architecture patterns within the Agri-Energy Connect Platform is very important for scalability, maintainability and efficiency. Adopting the MVC pattern ensures modular development while Microservices Architecture offers the flexibility and resilience. Additionally, employing the Repository Pattern simplifies data access, assisting with code reusability and streamlining database interactions. These patterns collectively streamline development processes, facilitating the seamless integration of platform features and contributing to the project's success (Hernandez, 2021) (Tech, 2024).

## Justification of Application:

**The relevance of Design and Architecture Patterns:**
Within the Agri-Energy Connect Platform project, design patterns and architecture patterns are relevant and hold significant importance. These patterns offer proven solutions to common design and architectural challenges and provide a structured approach to building scalable, maintainable and efficient software systems. Given the nature of the platform, where the various features and functionalities need to be

seamlessly integrated while ensuring performance, security and usability, design and architecture patterns are required (Geeks, 2021) (Tech, 2024).

**The value added by Design and Architecture Patterns:**

1. **Modular Development and Scalability (Model-View-Controller Pattern):** By breaking up responsibilities, the Model-View-Controller (MVC) design makes it easier to develop modular applications. As a result of this division, maintainability and scalability are improved as each component may function independently. In the context of the Agri-Energy Connect project, the MVC pattern guarantees that the platform may grow horizontally without sacrificing architectural integrity or performance, which is critical given the possible rise in user base and data volume in the future (Hernandez, 2021).
2. **Flexibility and Resilience (Microservices Architecture):** Flexibility and resilience are improved by the platform's integration of microservices architecture. The platform becomes dynamically flexible by breaking it up into smaller, standalone services, each focusing on specific tasks like the marketplace for green energy or the hub for sustainable farming. Because of its adaptability, the platform can keep up with evolving needs and technological breakthroughs, maintaining its long-term sustainability and relevance in a constantly changing world (Tech, 2024).
3. **Code Reusability and Simplified Data Access (Repository Pattern):** By removing data access and storage logic, the Repository Pattern encourages the reuse of code and improves database interactions. Repositories guarantee effective management of large databases and enable smooth feature integration by offering a consistent interface for handling data. This method not only improves scalability and maintainability but also increases the development processes, enabling the platform to adapt to new changes over time (Hernandez, 2021).

## Conclusion

In summary, the Agri-Energy Connect Platform proposal offers a promising solution to the pressing challenges of sustainable agriculture and green energy integration in South Africa. Through a thorough analysis of key non-functional requirements and the strategic integration of design and architecture patterns, a clear path forward for this innovative digital ecosystem is outlined.

Prioritizing scalability, security, usability and performance aims to create a platform that empowers farmers, green energy experts and enthusiasts to collaborate effectively and drive positive change. With a focus on intuitive interfaces, robust security measures, and efficient performance optimization techniques, confidence lies in the Agri-Energy Connect Platform to provide a valuable resource for advancing sustainable practices and renewable energy solutions in the agricultural sector.

# References

Couchbase. 2023. App Scaling (What It Is and How To Do It) [Online]. Available at: https://www.couchbase.com/blog/app-scaling/ [Accessed 15 April 2024].

Capital, F. 2019. The Role Of Encryption In Safeguarding Sensitive Financial Data [Online]. Available at: https://fastercapital.com/topics/the-role-of-encryption-in-safeguarding-sensitive-financial-data.html [Accessed 16 April 2024].

G., S. 2022. How Does Cloud Infrastructure Work? [Online]. Available at: https://nioyatech.com/beginners-guide-to-cloud-infrastructure-management/ [Accessed 15 April 2024].

Geeks, G. F. 2021. Software Architecture Patterns [Online]. Available at: https://www.geeksforgeeks.org/types-of-software-architecture-patterns/ [Accessed 17 April 2024].

Gillis, A. S. 2023. Role-based access control (RBAC) [Online]. Available at: https://www.techtarget.com/searchsecurity/definition/role-based-access-control-RBAC [Accessed 16 April 2024].

Hernandez, R. D. 2021. The Model View Controller Pattern – MVC Architecture and Frameworks Explained [Online]. Available at: https://www.freecodecamp.org/news/the-model-view-controller-pattern-mvc-architecture-and-frameworks-explained/ [Accessed 17 April 2024].

Kazmi, R. 2023. Scalable Applications: Curious Why Scalability Matters? [Online]. Available at: https://www.koombea.com/blog/why-scalability-matters-for-your-app/ [Accessed 16 April 2024].

Nagpal, R. 2018. Usability factors of mobile applications [Online]. Available at: https://www.researchgate.net/figure/Usability-factors-of-mobile-applications_fig2_320932436 [Accessed 16 April 2024].

Paloalto. 2021. What Is Data Privacy Compliance? [Online]. Available at: https://www.paloaltonetworks.com/cyberpedia/data-privacy-compliance [Accessed 16 April 2024].

Pedamkar, P. 2023. What is MVC Design Pattern? [Online]. Available at: https://www.educba.com/what-is-mvc-design-pattern/ [Accessed 16 April 2024].

Sheldon, R. 2024. Encryption [Online]. Available at: https://www.techtarget.com/searchsecurity/definition/encryption [Accessed 16 April 2024].

Tech, C. O. 2024. 10 microservices design patterns for better architecture [Online]. Available at: https://medium.com/capital-one-tech/10-microservices-design-patterns-

for-better-architecture-befa810ca44e#:~:text=Key%20benefits%20of%20using%20microservices%20design%20patterns&text=Creation%20of%20an%20application%20architecture,out%20incrementally%2C%2 [Accessed 16 April 2024].

Team, B. E. 2022. 10 principles of application usability [Online]. Available at: https://www.softwareone.com/en/blog/articles/2022/04/01/application-usability [Accessed 16 April 2024].

VMware. 2024. What is application security? [Online]. Available at: https://www.vmware.com/topics/glossary/content/application-security.html#:~:text=Application%20security%20is%20the%20process,as%20unauthorized%20access%20and%20modification. [Accessed 16 April 2024].