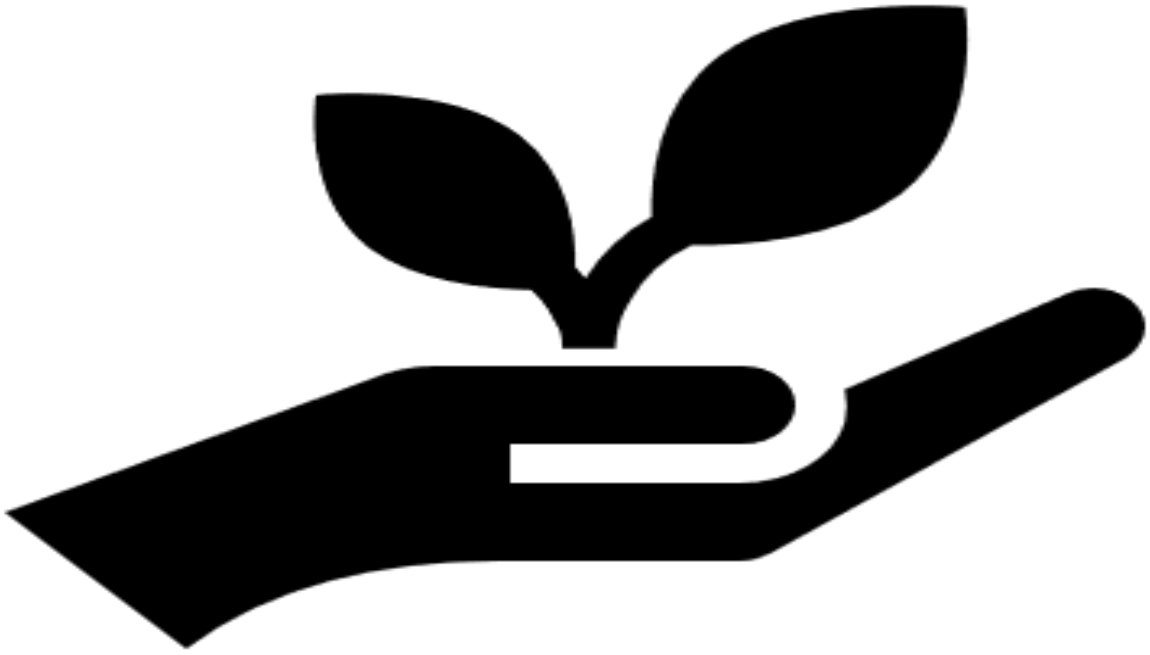
PROG7311 POE PART 1

ST10083262- Kaitlyn Abigail Naidoo

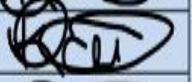
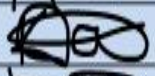


.

ACADEMIC HONESTY DECLARATION

Please complete the Academic Honesty Declaration below.

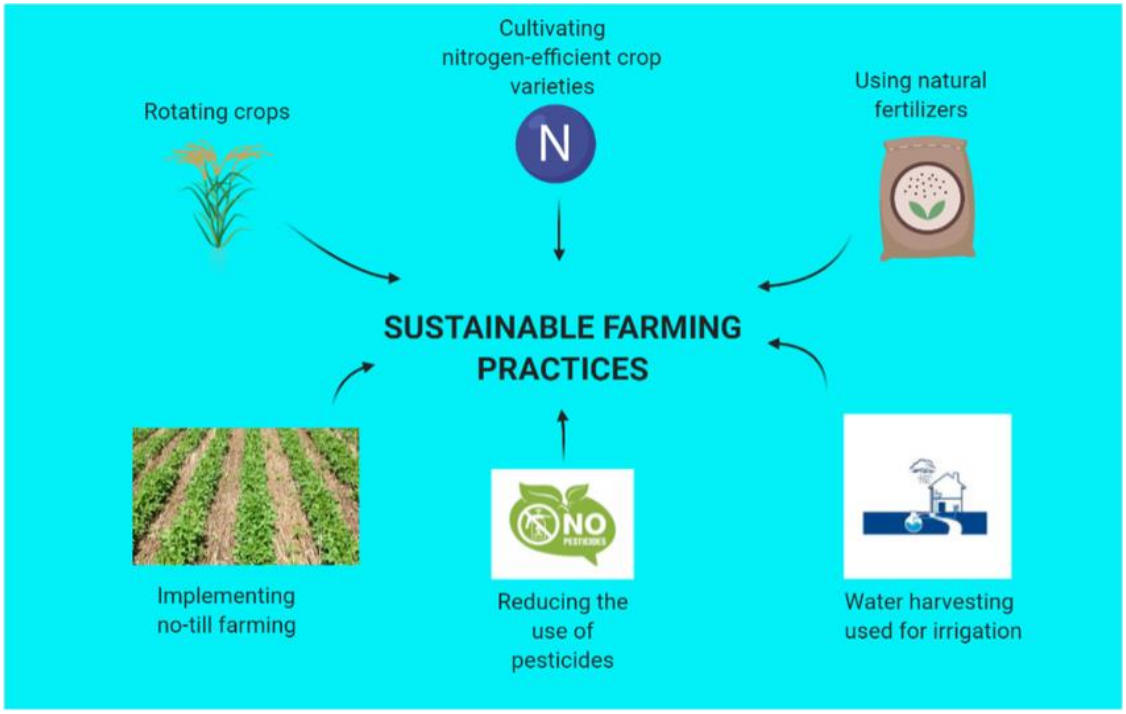
Please note that your assessment will not be marked, and you will receive 0% if you have not completed ALL aspects of this declaration.

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

Introduction:

The Agri-Energy Connect platform is an innovative combination of cutting-edge green energy technologies and sustainable agriculture practices. The purpose of this digital ecosystem is to facilitate the effective collaboration of farmers, energy experts, and enthusiasts, hence promoting the adoption and integration of renewable energy resources in the agricultural sector. With the increasing need for environmentally conscious farming practices due to global environmental issues, the Agri-Energy Connect platform hopes to be a key instrument in advancing ecological sustainability and energy efficiency. In order to guarantee that the platform is reliable, scalable, and easy to use, this proposal describes the essential non-functional requirements, design tenets, and architectural patterns that will guide the platform's development.

(Regen,n.d)



https://www.researchgate.net/publication/368361187/figure/fig3/AS:11431281128442741@1679367595193/The-different-practices-tools-for-sustainable-and-eco-friendly-farming-strategies.png

(altexsoft, 2023)

Analysis of Non-Functional Requirements:

- *Scalability*: It is critical that the platform can handle growing loads without experiencing performance issues. Elastic cloud computing resources will be used, which dynamically scale based on data volume and user demand. This strategy accommodates peak periods such as harvest seasons or financing deadlines while maintaining operational efficiency and cost-effectiveness.

  (Indeed, 2022)
- *Security:* Given the sensitive nature of the data—which includes private user information and proprietary information for agricultural techniques—security is of the utmost importance. To protect the platform, this strategy includes putting in place strong access controls, thorough data encryption, and frequent security assessments. Users can feel confident in their interactions on the platform since we will abide by local data privacy laws and worldwide best practices and compliance requirements like the General Data privacy Regulation.

  (Indeed, 2022)
- *Usability:* We will provide an intuitive interface with user-friendly navigation to guarantee that the platform is usable by users with different levels of technical knowledge. A wider range of access and engagement will be made possible by the responsive design, which guarantees compatibility across platforms and devices. Accessibility requirements will receive extra consideration in order to accommodate all users, including those with disabilities.

- *Performance:* There will be less downtime and fast loading times thanks to the platform's excellent performance optimization. To guarantee a seamless and responsive user experience, strategies including effective backend coding, optimized database queries, and the usage of content delivery networks will be employed.

  (Sage, n/d)
  Performance Metrics
  The platform aims to achieve regarding performance:

  - **Response Time**: Target the system to respond to user inputs within 2 seconds under normal conditions and 4 seconds under peak traffic.

  (stackify, 2024)

  Load Testing

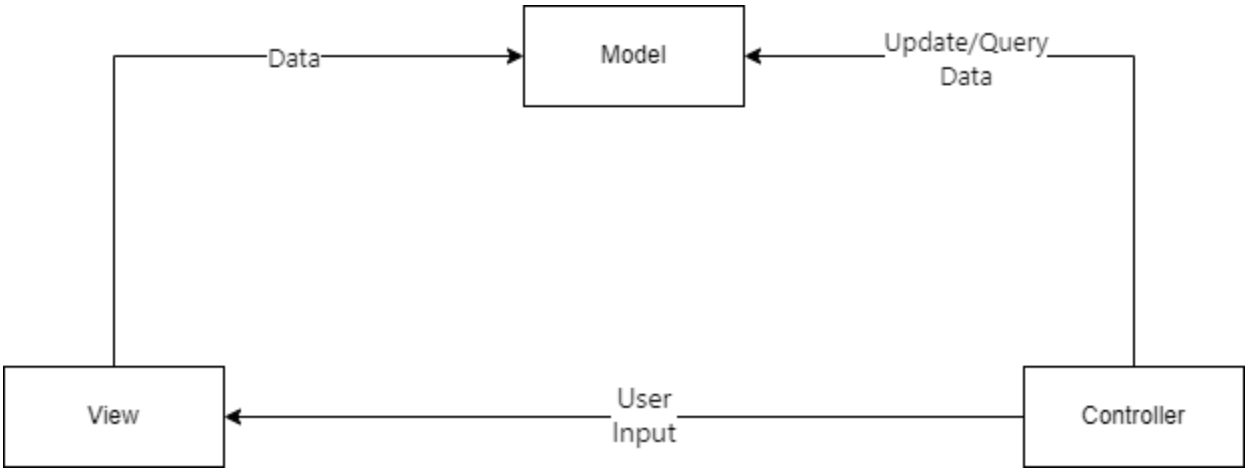
  - **Tools and Technologies**: Use performance testing tools like Apache JMeter or LoadRunner to simulate both user interactions and API traffic to test the robustness of backend services**.**

- *Impact on Development:* Addressing these non-functional requirements calls for a robust preparation and implementation plan, as well as the adoption of agile techniques to promote adaptability and iterative testing throughout the development process.

(Linkedin, 2024)

<u>Role of Design and Architecture Patterns:</u>

- *Model-View-Controller*: By organizing the program into three interconnected components, this architectural pattern will improve scalability and maintainability. By isolating the functionality, user interface, and input control, it streamlines the platform and makes independent development and testing easier.



Model: Manages data, business logic, and rules

View: Displays model data, receives user input

Controller: Processes user input, updates model, selects view

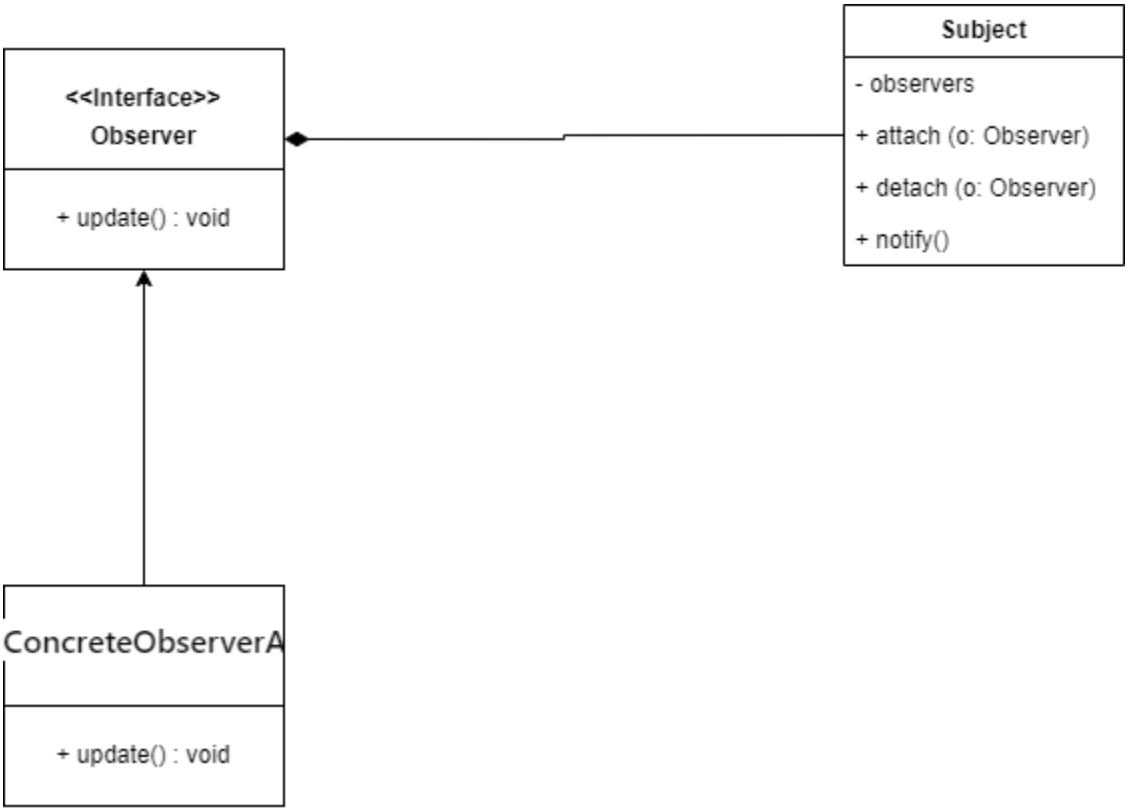
- *Observer Pattern*: This pattern is necessary for real-time updates since it will allow components to effectively monitor and respond to state changes. Features like real-time market price notifications or community forum updates are especially beneficial with this.

(Refactoring, n/d)

The **Observer Pattern** is a design pattern in which a **subject** (or object) keeps track of its dependents (or **observers**) and automatically tells them (typically by calling one of their methods) of any changes to the subject's state. Implementing distributed event-handling systems is its main usage. The design pattern encourages loose coupling between interacting elements, which is necessary to create applications that are scalable and maintainable.

(Smartdraw, n/d)

I will provide a UML diagram and description to allow for explanation:



(Geeksforgeeks, 2024)

This UML diagram illustrates the Observer design pattern, which is used to establish a one-to-many dependency between objects so that when one object changes state, all its dependents are notified and updated automatically.

- **Observer Interface**: This is an abstract class or interface that defines the **update** method. This method is called to update the observer about changes in the subject.
- **ConcreteObserver Classes**: These are specific implementations of the Observer interface (**ConcreteObserverA**, **ConcreteObserverB**, etc.). Each observer implements the **update** method according to its specific reaction to the subject's state change.
- **Subject Class**: This class maintains a list of its dependents, the observers. It provides methods for attaching (**attach()**) and detaching (**detach()**) observers to and from the list. Whenever a significant event occurs or its state changes, the **notify()** method is called, which in turn calls the **update()** method on each observer in the list.

  (Geeksforgeeks, 2024)
- **Relationships**:
  - **Implementation**: The **ConcreteObserver** classes implement the **Observer** interface, indicating that they must define the **update** method.
  - **Aggregation**: The aggregation relationship between the Subject and Observer shows that the Subject contains references to one or more Observers, but the lifecycle of the Observers is not managed by the Subject.

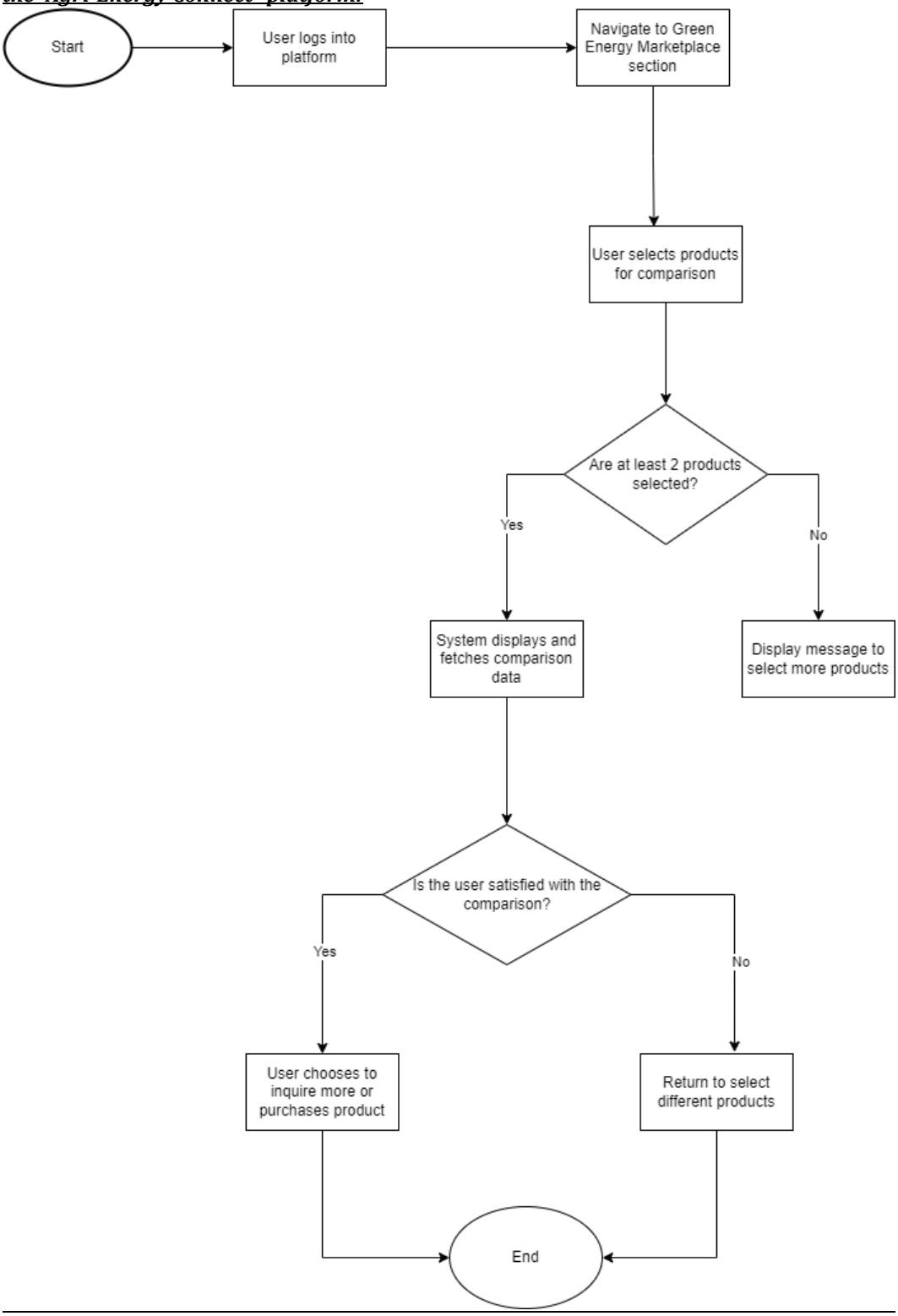
(Pentalogy, 2023)

- *Singleton Pattern*: Essential for handling shared resources and configurations, the Singleton pattern makes sure that some classes have only one instance across the lifespan of the program, avoiding platform-wide inconsistencies.

  (Spiceworks, 2022)
- *Microservices Architecture*: We improve the scalability and ease of maintenance of the application by breaking it up into smaller, loosely linked services. By delegating a specialized task to each service such as product listings, user management, or transaction processing the platform will be more resilient and adaptable.

  (corporatefinanceinstitute, n/d)
- *Value Addition:* The Agri-Energy Connect platform's operational requirements and strategic objectives are addressed by the design and architecture patterns selected. They add to a strong base that facilitates scalability, resilience, and ease of upkeep.

Flowchart depicting a user interaction process: a user comparing green technology products on the "Agri-Energy Connect" platform:



(Indeed, 2023)

Methodologies:

(Teamwork,n/d)

1. *Project Management Approach*

- Agile Development: To enable iterative and incremental development processes, apply an agile software development methodology. This methodology facilitates adaptability in addressing evolving demands from interested parties or when fresh perspectives surface via continuous enhancement.
- Scrum Framework: Using specified sprints and ongoing feedback loops, Scrum enables teams to be arranged around the idea of regular and continuous production increments.

2. *Design and Development Techniques*

- Using methods including user interviews, personas, user journey maps, and usability testing, user-centered design focuses on comprehending user needs, behaviors, and environments.

- Prototyping: Create preliminary models to experiment with various layouts and features. These models will be improved via several iterations in response to input from stakeholders and users.

Findings:

(Linkedin, 2023)

1. *User Preferences and Needs*

The results show that there is a significant need for a centralized hub where farmers can obtain information and goods about green energy and sustainable farming. Users stated that they needed a platform that was simple to use, offered collaborative tools, and real-time assistance.

(Aha,n/d)

2. *Analysis of Technology Gaps*

Significant gaps exist in current platforms regarding the integration of green technology solutions with agricultural needs, as revealed by the investigation. Many platforms focus on one area but struggle to smoothly combine the two.

(Dynatrace,n/d)

3*. Scalability and Performance*

The suggested technological stack performs well in handling simulated user loads, according to preliminary testing. Iterative testing and constant monitoring, however, will be necessary to make sure the platform can grow with the number of users.

Conclusion:

The Agri-Energy Connect platform offers a strong, scalable, and user-friendly digital environment and is intended to be a cornerstone in the integration of green energy and sustainable agriculture. Through the utilization of sophisticated design and architecture patterns, together with an emphasis on essential non-functional requirements, this platform will enable significant partnerships, stimulate creativity, and propel the implementation of sustainable practices throughout the agriculture industry. Our plan presents a progressive strategy that not only attends to the community's immediate needs but also creates the groundwork for expansion and adaptability in the future.

Referencing:

- altexsoft.2023. Nonfunctional Requirements in Software Engineering: Examples, Types, Best Practices, 30 December 2023. [Online].Available at: https://www.altexsoft.com/blog/non-functional-requirements/ [Accessed 14 April 2024].

- Regenz.n/d. Farming Sustainably in South Africa | Sustainable Farming Guide. [Online]. Available at: https://regenz.co.za/resources/farming-sustainably-in-south-africa/ [Accessed 14 April 2024].

- Indeed.2022. 9 Nonfunctional Requirements Examples, 25 June 2023. [Online]. Available at: https://www.indeed.com/career-advice/career-development/non-functional-requirements-examples [Accessed 14 April 2024].

- Sage. n/d. What are performance metrics? [Online]. Available at: https://www.sage.com/en-us/blog/glossary/what-are-performance-metrics/#:~:text=Performance%20metrics%20are%20used%20to,achievement%20of%20overall%20business%20goals. [Accessed 15 April 2024].

- Stackify. 2024. What is Load Testing? How It Works, Tools, Tutorials, and More, 29 February 2024. [Online]. Available at: https://stackify.com/what-is-load-testing/#:~:text=Load%20testing%20examines%20how%20the,software%20development%20project%20nears%20completion. [Accessed 15 April 2024].

- Linkedin, 2024. Building Digital Cities: The Roles of Architecture & Design Patterns in Software Engineering, 5 March 2024. [Online]. Available at: https://www.linkedin.com/pulse/building-digital-cities-roles-architecture-design-patterns-behfard-jlbnf/ . [Accessed 15 April 2024].

- Refactoring, n/d. Observer. [Online]. Available at: https://refactoring.guru/design-patterns/observer. [Accessed 16 April 2024].

- Smartdraw, n/d. UML Diagram Examples. [Online]. Available at: https://www.smartdraw.com/uml-diagram/examples/ [Accessed 16 April 2024].

- Geeksforgeeks. 2024. Observer Design Pattern, 6 February 2024. Observer Design Pattern. [Online]. Available at: https://www.geeksforgeeks.org/observer-pattern-set-1-introduction/ [Accessed 16 April 2024].

- Pentalogy. 2023. The Singleton Design Pattern, 27 June 2023. [Online]. Available at: https://www.pentalog.com/blog/design-patterns/singleton-design-pattern/#:~:text=Singleton%20is%20a%20creational%20design,from%20anywhere%20in%20the%20codebase. [Accessed 16 April 2024].

- Spiceworks, 2022. What Are Microservices? Definition, Examples, Architecture, and Best Practices for 2022, 5 April 2022. [Online]. Available at: https://www.spiceworks.com/tech/devops/articles/what-are-microservices/ [Accessed 16 April 2024].

- Corporatefinanceinstitute, n/d. Value Added. [Online]. Available at: https://corporatefinanceinstitute.com/resources/valuation/what-is-value-added/#:~:text=Value%20added%20is%20the%20extra,sale%20to%20the%20end%20customer. [Accessed 17 April 2024].

- Indeed.2023. What are methodologies? Definition, explanation and example, 10 January 2023.[Online]. Available at: https://uk.indeed.com/career-advice/career-development/what-are-methodologies [Accessed 17 April 2024].

- Teamwork.n/d. Project Management Methodologies. [Online]. Available at: https://www.teamwork.com/project-management-guide/project-management-methodologies/ [Accessed 17 April 2024].

- Linkedin.2023. Understanding Users and User Needs, 4 November 2024. [Online]. Available at: https://www.linkedin.com/pulse/understanding-users-user-needs-machinemindstechnologies-yy6if/ [Accessed 17 April 2024].

- Aha. n/d. What is a technology gap analysis?. [Online]. Available at: https://www.aha.io/roadmapping/guide/it-strategy/technology-gap-analysis [Accessed 17 April 2024].

- Dynatrace. n/d. Differentiating Performance from Scalability. [Online]. Available at : https://www.dynatrace.com/resources/ebooks/javabook/performance-and-scalability/ [Accessed 17 April 2024].