

Proposal

PROG 7311 POE



June 23, 2023

Zànne Smit – ST10082074

ST10082074@vcconnect.edu.za

Contents

[Introduction 2](#_Toc138422658)

[Non-Functional Requirements (edited) 3](#_Toc138422659)

[Design & Architecture Patterns Relevance 5](#_Toc138422660)

[Design Patterns 5](#_Toc138422661)

[Architecture Pattern 6](#_Toc138422662)

[Optimized Performance 8](#_Toc138422663)

[Possible changes or additions 8](#_Toc138422664)

[Overall: 8](#_Toc138422665)

[Farmers portal: 8](#_Toc138422666)

[Employee Portal: 8](#_Toc138422667)

[Guidelines 8](#_Toc138422668)

[Software Development Methodology 10](#_Toc138422669)

[DevOps 11](#_Toc138422670)

[Framework 13](#_Toc138422671)

[Different Frameworks 13](#_Toc138422672)

[ITIL (Information Technology Infrastructure Library): 13](#_Toc138422673)

[Zachman Framework: 13](#_Toc138422674)

[TOGAF (The Open Group Architecture Framework): 13](#_Toc138422675)

[Advised 14](#_Toc138422676)

[References 15](#_Toc138422677)

Below is an Edited proposal for the Farmers Brick and Mortar Website

# Introduction

This document is to enlighten you (the bid committee) on the non-functional requirements that will be of high importance on the website and store interactions between the Farmers and Employees. As well as why certain design patterns and an architecture pattern will be relevant to the planning behind this website that needs to be created. Required ways to optimize performance, software development methodology to ensure throughout the project and what framework the project needs to follow have also been included.

# Non-Functional Requirements (edited)

**What is a** **Non-Functional Requirement?**

Non-Functional Requirements refer to constraints or specifications that define the overall features of a system and are not related to the specific services provided to users. (Sommerville, 2016)

Each Farmer will need a profile that is **organized** and **easy** to navigate

They can have a **Navigation Bar** at the top of the page with the following details:

* About
* Details
* Products
* Add Product

These menu options will direct them to respective pages. This will make it easier for Farmers to view all their details in an organized and easy-to-navigate platform. It will also allow Farmers to see all their products and add new products.

This will be necessary as Farmers will need to establish if they should produce more or less of an item.

In the plan to develop the respective software pages. Each menu option will need to be created and made easy to view.

A Framer will need to **add Products easily** on the website that will automatically update to the Database. The data can be things such as:

1. Name
2. Description
3. Date Added
4. Quantity

Using clearly marked text fields and a button a Farmer can **easily** add this type of data.

This will all need to link to a specific Farmers ID.

The website on the store **employee end** should be **straightforward to operate** and the **updates** on the website need to be **quick**.

If it is not easy to operate by the employees selling stock or recording what comes in, a Farmer will not be able to see all their stock or income.

This could be done by:

* An employee **records new Farmers’ data** which automatically updates to the Database, the data can be things such as:
  1. Code/ID
  2. First Name
  3. Surname
  4. Email
  5. Phone Number
  6. Street Address
  7. City

The plan for the software development will need to be changed by creating these respective forms or pages that the employees can easily operate, and these forms will need to be linked to each respective place on the website that will still allow for swift updates.

The Farmer will want to be the **only one** with **access** to his/her profile. They will want **Security** measures put in place to protect their information.

A Farmer will not want other Farmers or users to have access to their personal data such as their stock, income, personal details, payment details, and other related information. As a competitor could sabotage them.

This can be dealt with by having a **login** for Farmers and they will need a unique **Email** and **password** to enter.

The software development side will need to have a **secure server and database** with safety measures to ensure no one can hack these login details. It will also require an extra menu option and page for Farmers to access on the website.

The website needs to be **reliable** meaning the site does not hang or show a “*site is unavailable*” message.

When Farmers use the website, they will want to do it swiftly and easily and if the site hangs or is inaccessible they will be wasting valuable production time. Some Farmers do not have access to the internet on their farms and must drive somewhere further to access it and if the website is down, they would’ve traveled there in vain.

This could be ensured by possibly having a **generator** for the servers should there be a power outage as well as a **technician on-site** if there should be a malfunction.

In the Software development plan, they will need to ensure a **constant runtime** and ensure the servers are on the **cloud** so that there is no break in connection as well as a **backup server**.

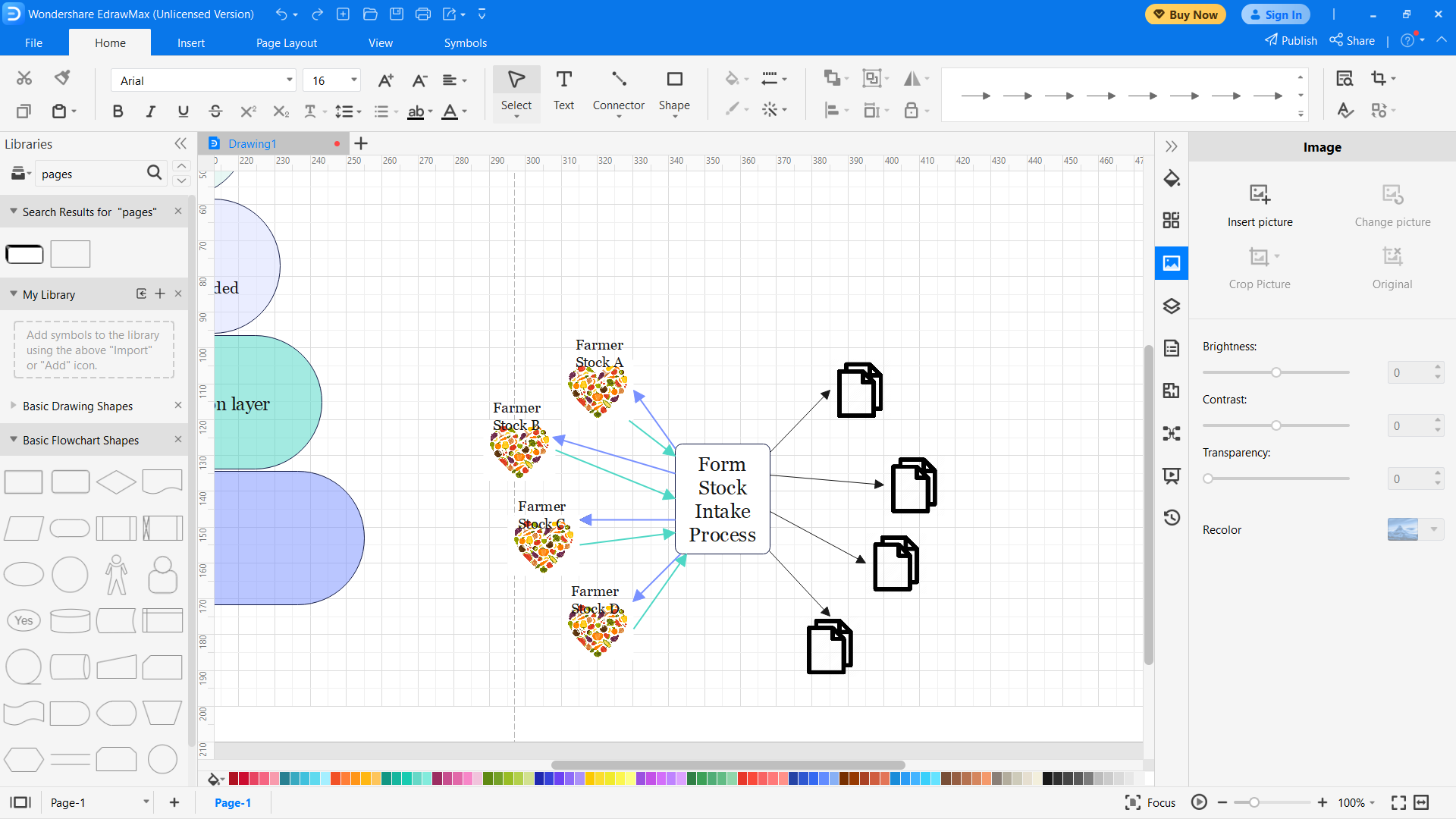
# Design & Architecture Patterns Relevance

## Design Patterns

***(Definition)***Design patterns are a guide or framework for addressing a problem that can be applied in various scenarios, there are many patterns, but they get categorized into 3 main sets **Creational** design patterns, **Structural** patterns, and **Behavioral** patterns. (Malayasreh, 2022)

Design patterns are necessary for the website otherwise there would be chaos within the code and the site might not run or data might pop up in the wrong space. It also makes the **system faster** as the website developer doesn’t need to reinvent new ways to solve tasks if he/she has already created a solution.

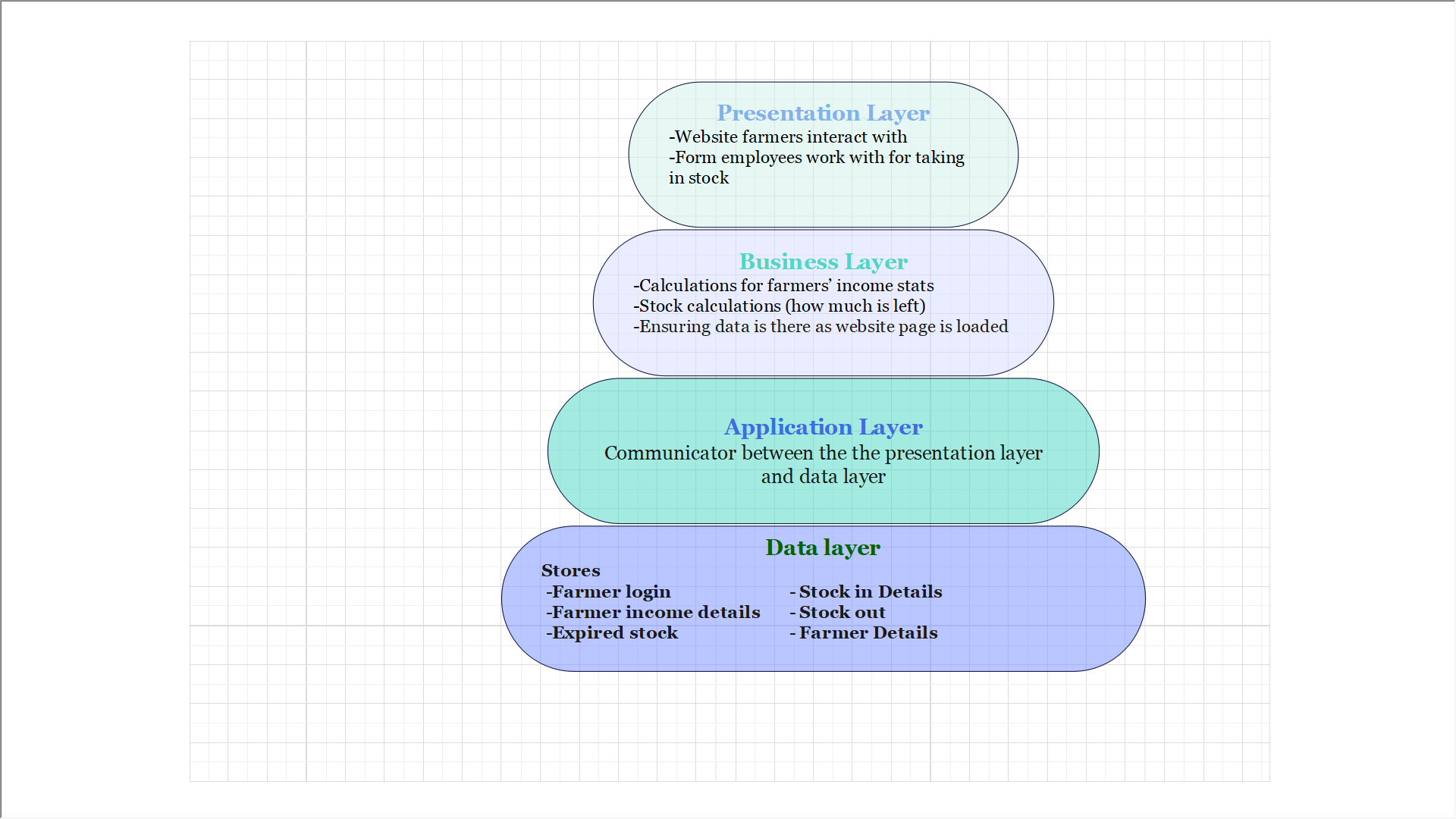
A **creational pattern** that could be used is a singleton pattern which restricts the instantiation of a class to a single instance and offers a way to access it globally. Its advantages are that it can be written quickly and easily, and it can have precise control over when and how it is accessed. (Ionos, 2021) So, this can be applied to this instance when recording a new stock intake as the same process will be used every time new stock comes in.



Design patterns are relevant to this website as they can structure code and make it easily readable when editing is necessary.

## Architecture Pattern

***(Definition)***The architecture pattern of software refers to the complete arrangement of its structure, encompassing the entire software system. Developers make use of various design patterns that align with the specifications and needs of the architecture. (Satyabrata\_Jena, 2021)

On the website the Farmers interact with, a **layered architecture** would be relevant as it will help **encompass all aspects**. In the diagram below you can see what role each layer would have and some of what would be done in the layer.

With a layered architecture, the website can be easily maintained and accessed when there are issues within whichever layer.

For more explanation behind the diagram

-Presentation Layer is what the user interacts with these will be all the pages on the website and the form the employee will use. If there is an issue for eg, The log in age refuses Farmers access, the development team can access just this layer to see what might be wrong.

-Business Layer is where are all the calculations are being done. If eg, the Farmer’s stock level is negative there is an error in the calculation behind it and the development team can fix it on that layer.

-Application Layer is what is communicating between the presentation layer and the data layer. For eg, when the Farmer selects the stock details page, this layer will fetch the data from the data layer and give it to the presentation layer to update the page as it is being entered by the Farmer.

-Data Layer, this is where all the data being stored that the website will use. Details such as the stock intake details, or Farmer income details can be stored here and accessed using the application layer to be used in the presentation layer.

# Optimized Performance

## Possible changes or additions

### Overall:

* Ensure all links to the database are working properly.
* Add more data to the database.
* Give 2 profiles that can be used in demonstration:
  + Employee – Email = [Zannesmit1@gmail.com](mailto:Zannesmit1@gmail.com), Password = admin
  + Farmer– Email = [harry@gmail.com](mailto:harry@gmail.com), Password = Harry
* Ensure there is more error handling when adding Farmers and Products and Editing Farmer details.

### Farmers portal:

* An expiry date could be offered to adding products.
* A page to view what has been sold and for how much, could be offered (this would need another table in the database).
* Ensure the Add Product works (didn’t work in prototype).

### Employee Portal:

* A page that links to a Farmer’s product where an employee can input which of the Farmers products have been sold and for how much (this would need another table in the database).
* Ensure adding Farmer Works (didn’t work in prototype).
* Ensure viewing Farmers products works (Collecting Data from db didn’t work in prototype).

## Guidelines

Follow these guidelines when developing the final software to ensure acceptable performance.

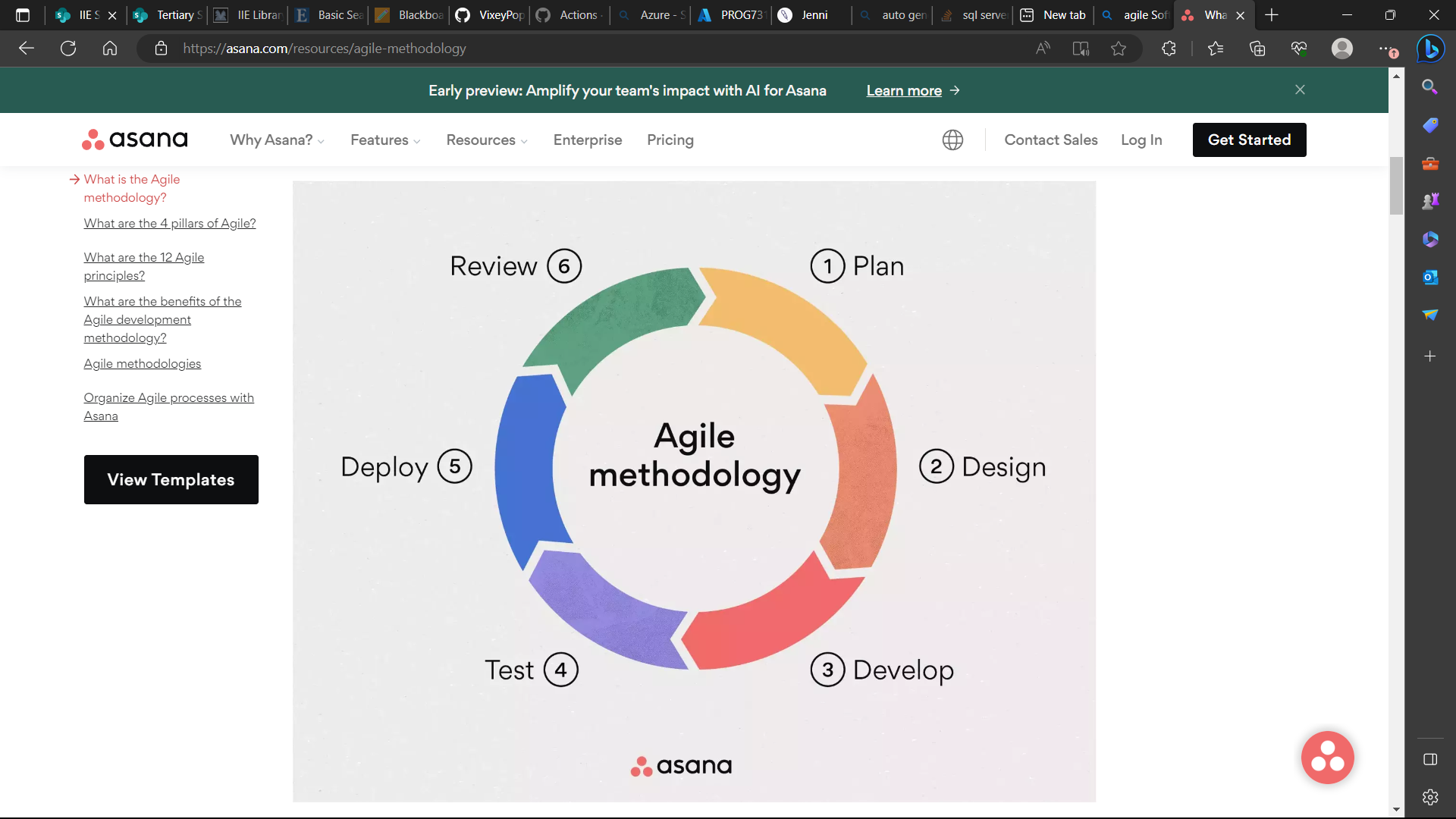
* Define Performance Requirements: Clearly define performance requirements in terms of response times, scalability, and resource utilization. These requirements should align with the expectations and needs of the end-users and stakeholders.
* Conduct Performance Testing: Perform thorough performance testing during the development and testing phases. Use appropriate tools and techniques to simulate real-world scenarios and stress test the software under various loads. This helps identify performance bottlenecks, scalability issues, and any other potential problems early on.
* Database Performance: Optimize database queries, indexes, and data retrieval operations to minimize response times. Ensure appropriate database indexing, data normalization, and query optimization techniques are implemented. Consider implementing database caching and query tuning for better performance.
* Scalability and Load Balancing: Design the software to be scalable, allowing it to handle increased loads and user demand. Implement load balancing techniques to distribute the workload across multiple servers or instances. Consider horizontal scaling by adding more servers or implementing cloud-based solutions to handle increased traffic and workload.
* Security Considerations: Ensure that the software follows security best practices to prevent performance-related vulnerabilities. Implement secure coding practices, authentication, authorization, and data encryption techniques without compromising performance.
* Regular Performance Monitoring: Continuously monitor the performance of the software in a production environment. Utilize monitoring tools and dashboards to track response times, resource utilization, and system health.
* User Feedback and Continuous Improvement: Gather feedback from end-users and stakeholders regarding the software's performance. Regularly evaluate and analyze performance metrics and user feedback to identify areas for improvement. Plan for regular updates and enhancements to address performance-related issues and optimize the software over time.

By following these guidelines, you can ensure that the final software delivers acceptable performance, meets performance requirements, and provides a positive user experience.

# Software Development Methodology

The Agile software development methodology would be a recommended approach for this development effort due to several reasons:

* 1. Iterative and Incremental Development: Agile methodology promotes iterative and incremental development, breaking the project into smaller, manageable iterations called sprints (Laoyan, 2022). In the case of building a website for a Farmer store, this approach allows for continuous feedback and adaptation throughout the development process. It enables the development team to deliver working functionalities at the end of each sprint, ensuring that the website evolves based on real-time feedback from users and stakeholders, such as employees and Farmers.
  2. Flexibility and Adaptability: Agile methodology emphasizes embracing change and responding to evolving requirements (Laoyan, 2022). In this project, where requirements and priorities may change as the business grows or new features are requested, Agile provides a framework that allows for flexibility and adaptability. It enables the development team to respond quickly to changing needs, making adjustments to the features, functionalities, and priorities based on feedback from the users and stakeholders.
  3. Collaboration and Communication: Agile methodology encourages close collaboration and communication between developers, stakeholders, and end-users (Laoyan, 2022). In the context of this project, involving employees and Farmers in the development process through regular interactions, such as sprint reviews and daily stand-up meetings, can lead to a better understanding of their needs and expectations. It ensures that the developed website meets their requirements and provides a user-friendly experience.
  4. Early and Continuous Delivery: Agile methodology promotes delivering working software early and continuously throughout the development process (Laoyan, 2022). This aspect aligns well with the goals of the Farmer store website project, as it allows for the timely deployment of essential functionalities to employees and Farmers. By delivering increments of the website on a regular basis, it enables stakeholders to start utilizing and providing feedback on the system sooner, ensuring that it meets their expectations and can be refined based on their input.
  5. Quality Focus: Agile methodology places emphasis on delivering high-quality software through practices such as continuous integration, automated testing, and regular review sessions (Laoyan, 2022). By incorporating quality assurance processes into the development lifecycle, the website can be thoroughly tested, ensuring that it functions as intended, meets security requirements, and provides a smooth user experience for both employees and Farmers.



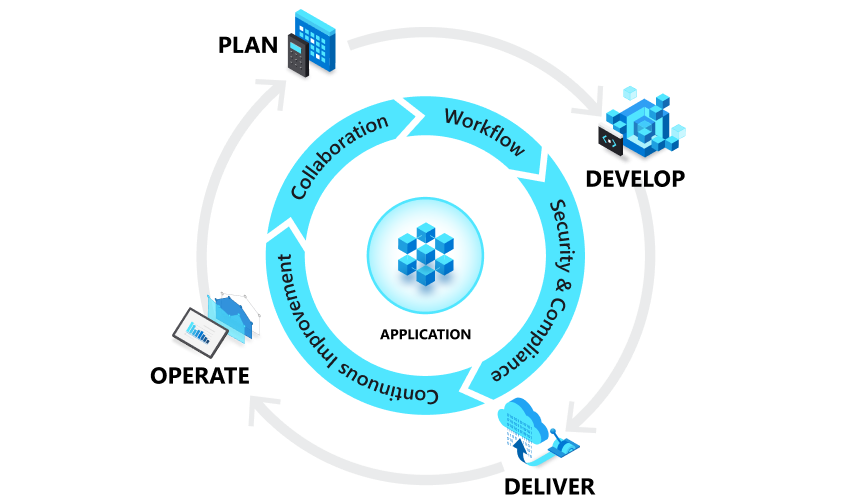
Laoyans (2022) diagram helps show the agile methodology cycle.

In summary, the Agile software development methodology is well-suited for this development effort due to its iterative and incremental approach, flexibility, collaboration, early delivery of working functionalities, and focus on quality. It enables the development team to adapt to changing requirements, involve stakeholders in the process, and deliver a website that meets the evolving needs of the Farmer brick and mortar store.

## DevOps

Implementing DevOps would be recommended for this development effort, it aligns well with the Agile software development methodology. Here's why and how DevOps fits in:

1. Continuous Integration and Deployment: DevOps promotes the practice of continuous integration and deployment, which involves automating the build, testing, and deployment processes (Danielson, et al., 2023). In an Agile development approach, where regular iterations are delivered, DevOps ensures that each increment is seamlessly integrated, tested, and deployed. This allows for faster feedback loops and the ability to deliver new features and updates to the Farmer store website more frequently and reliably. (Danielson, et al., 2023)
2. Collaboration and Communication: DevOps emphasizes collaboration and communication between development, operations, and other stakeholders involved in the software delivery process. It encourages cross-functional teams to work together, share knowledge, and align their goals (Danielson, et al., 2023). In an Agile environment, where close collaboration between developers, employees, and Farmers is crucial, implementing DevOps helps foster a culture of collaboration, enabling smooth communication and coordination between teams. This, in turn, leads to faster resolution of issues, improved feedback loops, and a better overall development experience. (Danielson, et al., 2023)
3. Automated Testing and Continuous Monitoring: DevOps promotes the use of automated testing and continuous monitoring to ensure the quality and stability of the software (Danielson, et al., 2023). In Agile development, where regular iterations are delivered, automated testing becomes even more essential to maintain a high level of quality and catch any regressions introduced by new features. By integrating automated testing into the development process, DevOps helps ensure that each increment of the Farmer store website is thoroughly tested, reducing the risk of introducing bugs and issues. (Danielson, et al., 2023)
4. Continuous Feedback and Improvement: DevOps encourages a culture of continuous feedback and improvement. By closely monitoring the performance and usage of the website, DevOps enables teams to gather valuable data and insights that can be used to enhance the application (Danielson, et al., 2023). In an Agile development approach, this feedback loop is essential to continuously refine and improve the website based on user and stakeholder input. DevOps practices allow for rapid feedback gathering, analysis, and the implementation of necessary improvements. (Danielson, et al., 2023)



Danielson, et al. (2023) diagram clearly shows the phases of the DevOps application lifestyle.

Overall, implementing DevOps in conjunction with the chosen Agile software development methodology brings several benefits. It ensures smooth integration and deployment of new increments, facilitates collaboration and communication between teams, enables automated testing and continuous monitoring, and fosters a culture of continuous improvement. By adopting DevOps practices, the development team can enhance the efficiency, reliability, and quality of the Farmer store website throughout the Agile development lifecycle. (Danielson, et al., 2023)

# Framework

## Different Frameworks

Here are different frameworks and what they are.

### ITIL (Information Technology Infrastructure Library):

ITIL is a widely adopted framework that provides guidelines and best practices for IT service management (ITSM). It offers a set of practices for planning, delivering, and managing IT services to meet business requirements. ITIL focuses on aligning IT services with the needs of the organization and improving the overall efficiency and effectiveness of IT operations.

According to IBM (2021), "ITIL is a framework of best practice approaches intended to facilitate the delivery of high-quality information technology (IT) services." (IBM, 2021)

### Zachman Framework:

The Zachman Framework is an enterprise architecture framework that provides a structured approach for organizing and describing an enterprise's architecture. It defines a set of perspectives and artifacts that help stakeholders understand and communicate various aspects of an organization's architecture. The framework comprises six columns (What, How, Where, Who, When, and Why) and six rows (Planner, Owner, Designer, Builder, Subcontractor, and User) to capture different viewpoints and perspectives.

According to Watts, S. (2019), "The Zachman Framework is an ontology for organizing architectural artifacts, developed by IBM in the 1980s." (Watts, S., 2019)

### TOGAF (The Open Group Architecture Framework):

TOGAF is a comprehensive framework for enterprise architecture (EA) that provides a structured approach for designing, planning, implementing, and governing enterprise information technology architecture. It offers a set of best practices, methods, and tools to support the development and management of enterprise architecture. TOGAF helps organizations align their IT infrastructure with business goals and enables effective decision-making for technology investments.

Visual Paradigm (2022) states that "TOGAF, an Open Group standard, is a proven enterprise architecture methodology and framework used by the world's leading organizations to improve business efficiency." (Visual Paradigm, 2022)

## Advised

A combination of the ITIL and TOGAF frameworks would be most suitable.

ITIL (Information Technology Infrastructure Library):

ITIL focuses on IT service management and provides best practices for planning, delivering, and managing IT services. The employee portal in this project involves managing the database of registered Farmers, adding new Farmers, and viewing products. ITIL can help in defining processes and workflows for managing the Farmer database, ensuring efficient and effective management of data and services. It can also guide in handling employee details, such as authentication and access control to protect sensitive information.

TOGAF (The Open Group Architecture Framework):

TOGAF is an enterprise architecture framework that helps in designing, planning, implementing, and governing enterprise IT architecture. The Farmer portal in this project requires the Farmer to have control over inventory, add new products, and view/edit their profile. TOGAF can provide guidance in designing a scalable and flexible architecture that supports these functionalities. It can assist in defining the structure and components of the system, integrating with the Azure database, and ensuring the security and reliability of the inventory management and profile editing processes.

By combining ITIL and TOGAF, the project can benefit from both frameworks' strengths. ITIL ensures that IT services are well-managed, aligning with the needs of the organization and providing a structured approach for employee portal functionalities. TOGAF helps in creating a robust and well-designed architecture that supports the Farmer portal's requirements, ensuring the scalability and maintainability of the system.

It is important to note that the Zachman Framework is more focused on organizing and describing architectural artifacts rather than providing specific guidance for service management or enterprise architecture development. While it can be useful for documenting and communicating the architecture, it may not directly address the operational and management aspects required.

By leveraging ITIL and TOGAF, the project can benefit from a comprehensive approach to IT service management and enterprise architecture, resulting in an efficient and scalable website for the Farmers brick and mortar store.

# References

Danielson, S. et al., 2023. *What is DevOps?.* [Online]   
Available at: https://learn.microsoft.com/en-us/devops/what-is-devops  
[Accessed 21 June 2023].

IBM, 2021. *What is IT Infrastructure Library (ITIL)?.* [Online]   
Available at: https://www.ibm.com/topics/it-infrastructure-library  
[Accessed 21 June 2023].

Ionos, W., 2021. *The singleton pattern – a class in itself.* [Online]   
Available at: https://www.ionos.com/digitalguide/websites/web-development/singleton-design-pattern/  
[Accessed 16 04 2023].

Laoyan, S., 2022. *What is Agile methodology? (A beginner’s guide).* [Online]   
Available at: https://asana.com/resources/agile-methodology  
[Accessed 21 June 2023].

Malayasreh, 2022. *Difference Between Architectural Style, Architectural Patterns and Design Patterns.* [Online]   
Available at: https://www.geeksforgeeks.org/difference-between-architectural-style-architectural-patterns-and-design-patterns/  
[Accessed 16 04 2023].

Satyabrata\_Jena, 2021. *Types of Software Architecture Patterns.* [Online]   
Available at: https://www.geeksforgeeks.org/types-of-software-architecture-patterns/  
[Accessed 16 04 2023].

Sommerville, I., 2016. *Software Engineering.* 10th ed. Harlow: Pearson Education.

Visual Paradigm (2022) TOGAF 9.1 Framework - A Comprehensive Guide, TOGAF 9.1 framework - A comprehensive guide. Available at: https://www.visual-paradigm.com/guide/togaf/togaf-91-framework/ (Accessed: 23 June 2023).

Watts, S. (2019) Introduction to Zachman framework, BMC Blogs. Available at: https://www.bmc.com/blogs/zachman-framework/ [Accessed: 21 June 2023].