ST10358162

PROG 7311 POE

GROUP 1

Zahra Karan

Contents

[Agri-Energy Connect Prototype Report 2](#_Toc201684614)

[Performance Optimisation of the Prototype 2](#_Toc201684615)

[Recommended Software Development Methodology 4](#_Toc201684616)

[Implementation of DevOps 5](#_Toc201684617)

[Architecture Framework 7](#_Toc201684618)

[Description of Technical Solution 10](#_Toc201684619)

[Feedback 11](#_Toc201684620)

[Resubmission of Part 1 and Part 2 11](#_Toc201684621)

[Part 1 – Original Submission 11](#_Toc201684622)

[Lecturer Feedback - 11](#_Toc201684623)

[Part 2 – Original Submission 11](#_Toc201684624)

[1st correction 11](#_Toc201684625)

[Lecturer Feedback 12](#_Toc201684626)

[2nd correction 13](#_Toc201684627)

[Lecturer Feedback 13](#_Toc201684628)

[3rd correction 13](#_Toc201684629)

[References 15](#_Toc201684630)

YouTube video of application run-through: <https://youtu.be/dsW0cfswFZE>

# Agri-Energy Connect Prototype Report

## Performance Optimisation of the Prototype

Performance optimisation focuses on **making the system run more efficiently** by removing delays, lags, or anything that reduces the system’s performance time. This process **enhances the user experience**, reduces overall costs, and avoids possible disruptions (Camphouse, 2025). Agri-Energy Connect focuses on connecting farmers, their employees and various sustainability experts, and user experience is crucial to encourage users to join and **help the system grow**. The system can be optimised in the following ways.

1. **Asynchronous Programming**

This will ensure that the computer runs **multiple tasks at the same time**, instead of executing one task at a time, improving the overall runtime (The IIE, 2025).

This is done using the “**async**” and “**await**” keywords. The “async” keyword declares the method as **asynchronous**. The “await” keyword indicates the operation can be performed asynchronously and is contained within the “async” method (The IIE, 2025).

Below, is an example of the use of the keywords:

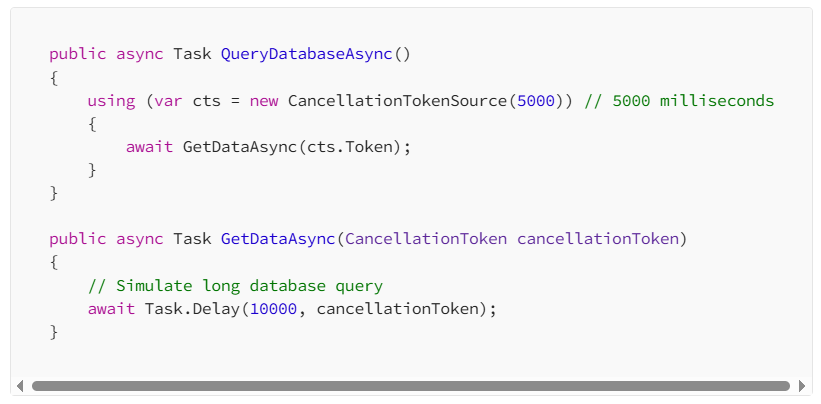


Figure 1

Source: <https://medium.com/codenx/c-threads-asynchronous-programming-79b5d8d787d5>

(Chinta, 2023)

The final prototype should try combining service calls to improve the system’s design and should **only make use** of asynchronous programming where the **code is slow**. Its use is unnecessary if the code is running fast, as the async **adds memory and time costs** (The IIE, 2025).

1. **Cross Apply**

“CROSS APPLY” is an **operator in SQL** Server, or a lateral join in SQL, which allows programmers to run a **mini query** for **each row in another table** and **combines** the **results**. It is most useful when used to fetch the **top or most recent related records** and improves the system’s performance by reducing the need for **multiple separate queries** (The IIE, 2025).

In the context of Agri-Energy Connect, CROSS APPLY could be used to retrieve each farmer’s **top selling product** without needing to run **multiple repeated queries**, or each employee’s **top selling farmer**. This will allow for **quicker access to information** when **displaying dashboards or summaries** for multiple users at once (The IIE, 2025).

1. **Chunky Service Design**

Agri-Energy Connect should use chunky service design because it requires only one call to get access to the data since it **sends fewer requests to the server**. This makes the **app** **work better in areas with slow internet**, like the rural areas the farms are situated on (The IIE, 2025).

1. **Cashing**

Another way to improve the system’s efficiency **offline** is through **cashing.** This will ensure that frequently accessed data gets **temporarily copied** to **fast storage** located **close** to the application, making it accessible when the device is offline (The IIE, 2025).

In Agri-Energy Connect, caching can be used to **store farmer and employee profiles** and recent **crop reports** locally on the device. This allows farmers or employees to still view information without needing a constant internet connection (The IIE, 2025).

1. **SOLID Principles**

To ensure the system is **well organised, understandable, easy to change and maintainable**, the SOLID Principles are essential. SOLID is an acronym for **single-responsibility principle, open-closed principle, Liskov Substitution principle, interface segregation principle, and dependency inversion principle**. Altogether, these principles keep the system at its best by promoting the flexible, clean code (Churchville, 2023).

For Agri-Energy Connect, this is important because the platform **may eventually grow** to include **new features** like **more user types**, **different sources of data**, or even **new** **programmers coding the system**. Using the SOLID Principles will ensure these changes can be made **without disrupting the system**, which helps the app stay **reliable**, **scalable** and **easy to update** (Churchville, 2023).

## Recommended Software Development Methodology

According to ScienceDirect, a development methodology is the **process used to plan, design, develop, test, and deploy software applications** (ScienceDirect, 2015). It is important that the development team responsible to produce the Agri-Energy Connect follows a methodology that will allow for **continuous releases of newer**, **more advanced versions of the prototype** to cater for **scalability**. The methodology should also allow for **flexibility**, in case **new requirements** are added to the application. The **Scrum methodology** best fits this approach, since it is **agile** and allows for **continuous delivery** of the product.

As shown in figure 2, the scrum cycle starts with observing the **product vision**, in this case, the vision is a platform that enables farmers and employees to collaborate and sell their products through an online forum. The team goes onto **planning** the overall process, then they go into the more specific **planning** of executing **deliverable** in the **product backlog**, which they work on in a single sprint, which is a **time-constrained cycle** (Awati and Brunskill, 2024), that usually last **two to six weeks** (The IIE, 2025) making it easier to **release updates in regular intervals** and make **quick responses** to feedback and **new requirements** added to the **product backlog**.

Scrum includes checking in daily through **daily stand-up meetings** which last around **15 minutes**, where each team member **reports on the** **progress** they have made ensuring the team is aligned. It **encourages good communication** between the **developers** and the **people who will use or manage** **the** **system**, which helps make sure the most important features are built first (Awati and Brunskill, 2024).

At the end of a sprint, the team has a **review meeting** to show the **product owner** the system and determine if the criteria are met, and the system gets **deployed** if criteria are met.

A diagram of a scrum

AI-generated content may be incorrect.

Figure 2

(Sergeev, 2020)

Source: <https://hygger.io/blog/what-is-scrum-lifecycle/>

The team should stay cautious of **scope creep**, which occurs when there are **new requirements added without making any formal documentation** of them. This could result in **project delays or going over budget**, (The IIE, 2025).

## Implementation of DevOps

“**DevOps**” is a concatenation of the words, “**development**” and “**operations**”. DevOps is the **collaboration** of the **application development team** and the **operations team**, so that the **same team completes the development and the deployment** of the application (Bigelow, et al., 2024).

The DevOps process aims to help teams:

* **Release software faster** and more often
* It makes it easier to **add new features**
* **Fix** **bugs** quickly
* **Update** the system without much **disruption**
* **Recover** quickly from **failures**

(Bigelow, et al., 2024).

The Agri-Energy Connect platform allows farmers and employees to **connect** from different parts of the world, and using **DevOps** will allow the team to **improve the platform without interrupting services** like the **marketplace** and **collaboration** **forum**.

On the DevOps continuum, there are **three different levels**:

1. **Continuous integration**

This is a development practice where developers are required to **integrate their code** in a **shared repository multiple times per day**. Each bit of code is **automatically tested** and **verified** by an **automated build process**, which ensures any **problems are detected early**. This **avoids future delays** in the project time, the **budget**, and allows the team to deploy the application **faster** (The IIE, 2025).

For Agri-Energy Connect, continuous integration ensures that **new features** like new products listed, updates on the collaboration forum, or security features added, can be **implemented** and **tested** **without** **disrupting** the users on the platform.

1. **Continuous delivery**

This is the ability to **make changes** to the application and **deploy** it **without causing any disruption**, quickly and sustainably. This **includes fixing bugs**, **changing the configuration**, and **adding new features**. In this process, more **automation** and **testing** are **implemented** which ensures the code is **always ready to deploy** when ready (The IIE, 2025).

For Agri-Energy Connect, this means that users across different regions can **continue to interact** with the **platform**, even while it is **being improved in the background**, **without** any **downtime**.

1. **Continuous deployment**

This is the process where the software that is built can be **deployed** to the **live environments**, **without** completing any **manual steps** that would increase the **production** **time** (The IIE, 2025). This means that farmers and employees won’t have to wait long for improvements.

The tools that can be utilised for the DevOps approach are as follows:

* To push the source code to the repository
  + Git
* To build the server
  + Jenkins
* To test the automation
  + Selenium
* To manage the configuration
  + Puppet

Containers

* + Docker
* To orchestrate containers
  + Kubernetes
* To monitor the application
  + Elasticsearch, Logstash, and Kibana (ELK)

(The IIE, 2025) pages: 134-135

## Architecture Framework

According to ScienceDirect, an **architecture framework** is a **way to organise and plan** all the **different parts** of a **system**. It helps people **use the same language** and **follow the same steps** when **designing** how things should work and ensuring that everyone understands **what needs to be done**, who is **responsible**, and what the **priorities** are (ScienceDirect, 2016). The development team for Agri-Energy Connect should use an architecture framework since it brings **clarity, structure and direction** to this complex project, and will **reduce the time spent** on **designing systems** **from** **scratch** (Theseira, 2024).

The **most suitable** architecture frameworks **for Agri-Energy Connect** would be a **combination of ITIL (Information Technology Infrastructure Library) and TOGAF (The Open Group Architecture Framework).**

**ITIL - (Information Technology Infrastructure Library)**

ITIL focuses on **IT service management**. This includes, keeping the system running in the most efficient way possible. It is all about creating a good user experience. It is used to manage the quality of the services, manage downtimes or issues and support the system after deployment, like updates and maintenance (Arun, 2024).

As shown in **Figure 3**, there are **five categories of ITIL Lifecycle** that make up the ITIL framework. Each category **contains a list of specific processes** that a development team uses to plan, create, deliver and improve IT services.

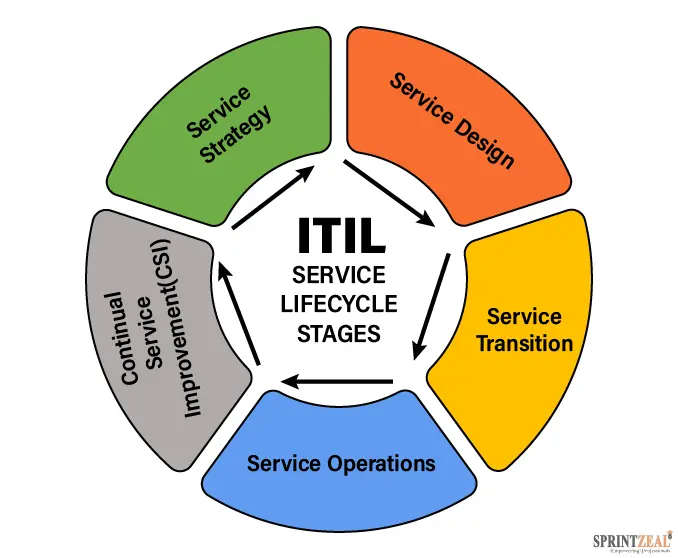


Figure 3

Source: <https://www.sprintzeal.com/blog/itil-processes>

(Yanthan, 2025)

**1. Service Strategy**

In this stage the team **plans which features and services matter most**. The team **creates** **a strategy** (Arun, 2024).

**2. Service Design**

In this stage the team **designs new features** like profile upgrades, personalised dashboards or a better forum layout. ITIL makes sure **all areas (people, tools, processes) are considered** when deciding (Arun, 2024).

**3. Service Transition**

In this stage, the team **tests the new features** before launching. They **use ITIL processes** to handle changes without breaking the live system. For example, smoothly upgrading the payment system on the marketplace without downtime (Arun, 2024).

**4. Service Operation**

In this stage, the team **makes fixes to bugs** quickly, they also **respond to user issues** fast, and keep the platform running 24/7. The ITIL’s **incident management** and **service desk** concepts can be used to help users when **problems occur** (Arun, 2024).

**5. Continual Service Improvement**

In this stage **the team reviews what works and what does not work**. **Feedback** from the **users** and **system** **data** are all used to **keep improving the user experience,** like adding translation or more specific product filters (Arun, 2024).

**TOGAF – The Open Group Architecture Framework**

TOGAF helps the team design and manage the overall architecture and helps the team identify new **business requirements.**

As seen in **Figure 4**, the TOGAF follows an 8-step process. In this example, an additional business requirement to **integrate weather APIs into the Agri-Energy Connect** platform that give farmers **insights and alerts** (Conexiam, 2025). This new **business requirement** is identified in the **Preliminary Phase** (Conexiam, 2025). Once the needs are identified, the high-level details of the system are described in the **Architecture Vision** step (Conexiam, 2025). The next step is the **Business Architecture**, where this phase focuses on how the (requirement) weather API fits into the **business processes** (Conexiam, 2025). Then the **Information Systems Architecture** specifies how the weather API is **technically integrated** into the **existing app** and defines the weather data sources as well as how they will be stored and used (Conexiam, 2025). The next phase is the **Technology Architecture,** which focuses on the **infrastructure and platforms** needed (Conexiam, 2025). Then onto **Opportunities and Solutions phase,** which identifies how to **implement** the new weather API feature. Next is the **Migration Planning**, which plans how to **roll out** the weather API feature into the platform in stages, like testing it with a small group of farmers first (Conexiam, 2025). Then the **Implementation Governance** stage ensures everything **correlates with the design** and **progress is monitored** (Conexiam, 2025). And finally, the **Architecture Change Management** step handles any **new changes, issues, bugs and improvements** to be made (Conexiam, 2025).

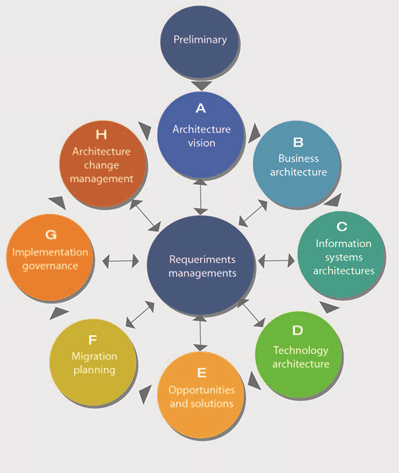


Figure 4

By combining the two frameworks, the **TOGAF helps the team plan and build new features**, and the **ITIL ensures the platform delivers a high-quality user experience**.

## Description of Technical Solution

**Agri-Energy Connect**

Agri-Energy Connect is a **web-based platform** built using **C# and .NET 8.0**. It is designed to connect farmers and agricultural employees.

The system includes core features such as **user registration and login**, a **digital marketplace** where **farmers can upload and sell their products**, a **collaboration forum**, and a **dashboard** that helps users **manage their activity**.

The platform also includes a **contact page** for support and communication.

There are **three main user** **roles**:

* **Farmers:** **can** **list products on the marketplace** and utilise all other core features.
* **Employees**: who can **register farmers** and utilise all other core features.
* **Admins**: who oversee and manage the entire platform.

The application can be accessed through the GitHub link: <https://github.com/VCSTDN2024/prog7311-part-2-zahrakarann.git>

# Feedback

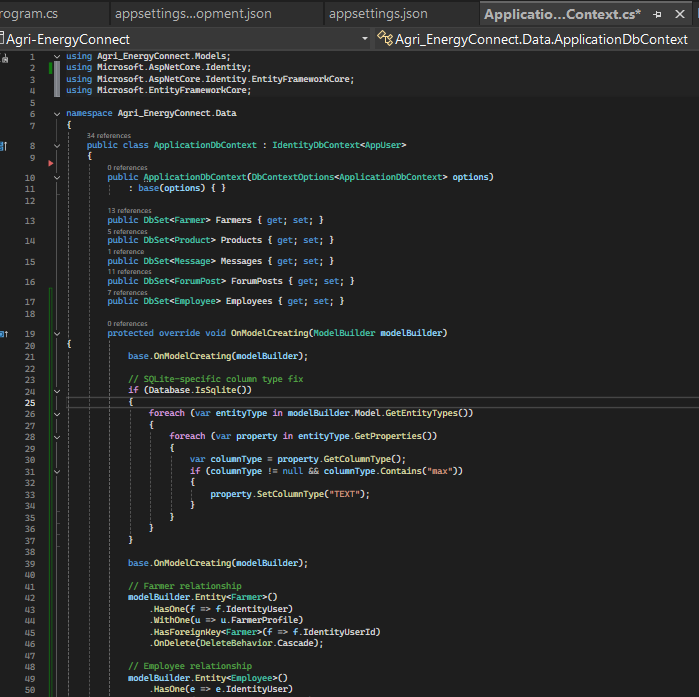
## Resubmission of Part 1 and Part 2

### Part 1 – Original Submission

### Lecturer Feedback -

### Part 2 – Original Submission

1st correction: ApplicationDbContext.cs:



A screen shot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

Lecturer Feedback – “Seed product data also, here you only have login data”

modelBuilder.Entity<Product>().HasData(

new Product

{

Id = 1,

Name = "Organic Compost",

Category = "Fertilizer",

Description = "Eco-friendly compost made from organic waste.",

Price = 120.00m,

ProductionDate = new DateTime(2024, 5, 20),

FarmerId = 1

},

new Product

{

Id = 2,

Name = "Drip Irrigation Kit",

Category = "Irrigation",

Description = "Complete drip irrigation system for small farms.",

Price = 950.00m,

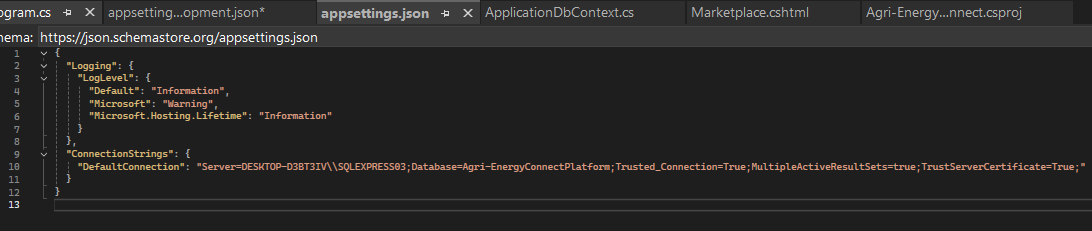
ProductionDate = new DateTime(2024, 6, 10),

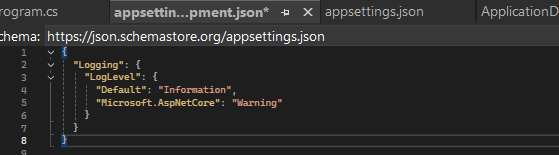
FarmerId = 1

}

);

2nd correction: appsettings.development.json





Lecturer Feedback – “I would definitely suggest to use InMemoryDatabase or Sqlite, specifically for prototype applications. Remember, that this projct is to be experimented with by client, i.e. they may be computer illiterate & you can't expect them to have sql server. Keep the sql server context, but SWAP OUT with sqlite or InMemoryDB. This makes it easier to run the app out the box & to seed data, & it also demonstrates that you understand the solid principles of not having to repeat code, i.e. to easily change code on demand, not by changing the code, but by chaning the behaviour of the code (behavioural pattern + repository opattern) Also, be sure to add product data”

3rd correction: In the appsettings.Development.json:

{

"Logging": {.77

"LogLevel": {

"Default": "Information",

"Microsoft.AspNetCore": "Warning"

}

},

"ConnectionStrings": {

"DefaultConnection": "Data Source=agri-energy.db"

}

}

# References

Arun, R. (2024) What is IT Service Management? Simplilearn. [Online] Available at: <https://www.simplilearn.com/tutorials/itil-tutorial/what-is-it-service-management#itsm_key_concepts> (Accessed on: 23 June 2025).

Awati, R. and Brunskill, V. (2024) What is a sprint (software development)? TechTarget. [Online] Available at: <https://www.techtarget.com/searchsoftwarequality/definition/Scrum-sprint> (Accessed on: 22 June 2025).

Bigelow, S., Courtemanche, M., Gillis, A. (2024) What is DevOps? Meaning, methodology and guide. TechTarget. [Online] Available at: <https://www.techtarget.com/searchitoperations/definition/DevOps> (Acessed on: 23 June 2025).

Camphouse. (2025) Performance Optimization: Techniques for Enhanced Application. Camphouse. [Online] Available at: <https://camphouse.io/blog/performance-optimization> (Accessed on: 22 June 2025).

Chinta, M. (2023) C# Threads & Asynchronous Programming. Medium. [Online] Available at: <https://medium.com/codenx/c-threads-asynchronous-programming-79b5d8d787d5> (Accessed on: 22 June 2025).

CollabNet. (2018) What is DevOps? The Ultimate Guide to DevOps. [Online] Available at: https://resources collab.net/devops-101/what-is-devops (Accessed on: 12 December 2022).

Conexiam. (2025) TOGAF Architecture Development Method Phases Explained. Conexiam. [Online] Available at: https://conexiam.com/togaf-adm-phases-explained/#b (Accessed on: 23 June 2025).

Churchville, F. (2023). The 5 SOLID principles of object-oriented design explained. TechTarget. [Online] Available at: <https://www.techtarget.com/searchapparchitecture/feature/An-intro-to-the-5-SOLID-principles-of-object-oriented-design> (Accessed on: 23 June 2025).

ScienceDirect. (2016) Architecture Framework. ScienceDirect. [Online] Available at: <https://www.sciencedirect.com/topics/computer-science/architecture-framework> (Accessed on: 23 June 2025).

ScienceDirect. (2015) Development Methodology. ScienceDirect. [Online] Available at: <https://www.sciencedirect.com/topics/computer-science/development-methodology> (Accessed on: 23 June 2025).

Sergeev, A. (2020) What is Scrum Process Lifecycle? Hygger. [Online] Available at: <https://hygger.io/blog/what-is-scrum-lifecycle/> (Accessed on: 23 June 2025).

The IIE. 2025. Leadership in education [PROG7311 Module Manual]. The Independent Institute of Education: Unpublished.

Theseira, D. (2024) What Is an Enterprise Architecture Framework? A Full Guide to Types and Process. Ardoq. [Online] Available at: <https://www.ardoq.com/knowledge-hub/enterprise-architecture-framework> (Accessed on: 23 June 2025).

Yanthan, N. (2025) ITIL Processes List – 26 ITIL Processes and 5 ITIL Service Lifecycle Stages. SprintZeal. [Online] Available at: <https://www.sprintzeal.com/blog/itil-processes> (Accessed on: 23 June 2025).