

SOEN ICE TASK 4

Software Engineering



September 18, 2023

Varsity college

[Company address]

**Layered Architecture Pattern:**

The layered Architecture pattern working on building the foundation of bottom layer and as its pattern grows or builds it means it builds the layer above layer. The higher layer more the functionality and more complexity, bottom layer is a basic functionality and complexity.

**Client-Server Architecture pattern:**

The Client Server Architecture patterns is a design pattern for application, and it contains two parts in an application which is Client, Server. The client and server work together like hand-in-hand where client is like a customer where it makes request to provide the specific result on the other hand where server is part of responding in application to respond the request from client.

**Pipe-Filter Pattern Architecture**

The Pipe-filter works like a production or process of an application divided into filters. Each filters works specific operation of an application process and passes through the pipe to execute everything.

**Repo Pattern Architecture**

The Repo pattern architecture is like a librarian of an application. It knows exactly where to fetch your data and where to use it. It can be able to get specific data from an application instead of going through whole data to find the specific value. This makes application faster, more efficient.

Based on the case study I will be choosing; the **Layered Architecture Pattern** seems to be the most suitable for implementing this software solution. Here’s why:

* Scalability: The Layered Architecture Pattern allows for high scalability, which is crucial for a non-profit organization that may need to expand its operations as it grows.
* Separation of Concerns: This pattern separates the system into distinct layers such as presentation, business, and data access layers. This would allow the non-profit to manage tree-planting efforts, communicate with supporters, and handle donations separately yet cohesively.
* Maintainability: Changes in one layer of the system (like adding a new communication channel with supporters) would not affect other layers. This makes the system easier to maintain and update.
* Reusability: Components within each layer are reusable, which can speed up development and reduce costs.

While other patterns like Client-Server could also be used (for example, to handle communication between the organization and its supporters), the Layered Architecture Pattern provides a comprehensive structure that addresses all the organization’s needs.