High Level Architecture Design (Layered Architecture)

Shaikha Al-Marri

201507150

Table of Contents

[Overview of Layered Architecture 1](#_Toc73607308)

[Advantages of Layered Architecture 1](#_Toc73607309)

[Non-Functional Requirements 2](#_Toc73607310)

# Overview of Layered Architecture

Layered architecture components are organized horizontally, each layer in this architecture has a specific role and responsibility within the system. For instance, the presentation layer is responsible in handling all user interface and browsing communication logic. While the business logic layer has responsibility to execute specific business rules associated with the request. Every layer in the architecture in order to satisfy a particular business request, forms an abstraction around the work which needs to be done. As an example, it is not necessarily for the presentation layer to know about how to get data; it only displays the information on a screen in specific format. Likewise, it is not necessarily for the business logic layer to be concerned about how to format data for display on a screen or even where the data is coming from; it only needs to get the data from the persistence layer, perform the business logic, and pass the data up to the presentation layer. Figure 1 shows the high-level architectural design.

# Advantages of Layered Architecture

Layered architecture has some features like the separation of concerns between components. The components within a certain layer deal only with logic that relates to that layer. For instance, components in the presentation layer deal only with logic of presentation, while the components in the business logic layer deal only with logic of business. Another feature is that the interaction between the layers is limited, every layer at most will communicate with the below or above layer, thus it will be simple to change each layer independently. This will fulfil the system’s modifiability requirement which will make adding new features and modifying existing ones without effecting other layers easier. Moreover, this architecture meets the scalability requirement of the system as servers can be duplicated in each layer. In the layered architecture, every layer is selected to be a set of related services; therefore, provides high degree of cohesion within the layer and hides it complexity from other layers. This can make each layer runs on its own machine, and therefore achieves portability across computing platforms. The reliability in this architecture is highly met as if any error occurs in a specific layer, other layers will not be affected and if the system under consideration requires high reliability, then this architecture has the capability to provide that.

# Non-Functional Requirements

|  |  |
| --- | --- |
| Non-Functional Requirements | Description |
| Modifiability | More functionalities need to be added to the system such as enter the records directly from the webpage. Also, some existing functions might be modified or deleted in order to enhance capabilities of the system such as modifying the interface of the system. |
| Scalability | The system can grow further to adjust huge number of recruitment employees. Such as, expanding it to be used by over a thousand of recruitment employees from different recruitment offices. |
| Usability | The interface of the system is user friendly and is implemented in such a way that it can be used easily by recruitment employees and ADLSA employees. |
| Reliability | The system will handle all the user’s errors. |
| Availability | The system will be accessible all the time 24/7. Any updates that occur during the sessions, will be reflected on the page without page refreshment. |

A picture containing application

Description automatically generated

Figure 1. High Level Architecture Diagram