AdventureLearn Game Application
Justification For Candidate Architecture

Version 1.0
24/02/2020

Members:
Aloysius Foo Dun Jiat
Farhan Khalifa Ibrahim
Florowski Kamil
Lee Jiho
Ng Jin Han Benedict
Nguyen Duy Khanh
Nicholas Irving Darmadji
Rickson Hu Hong Rui
Sven Tan Wei Jie
Yuen Kim Hwee

Team: NanyangGuys
CZ3003 Software System Analysis and Design, Tutorial Group SSP5

Table of Contents

1. Introduction

The 3-Tier Architectural Pattern is used as the basis of our Candidate Architecture. This architecture
is chosen because it organises the components into 3 layers (Presentation layer, Business Logic layer, and Data Access layer) thus yielding the following desired advantages for this project:

1. Improves development speed as team members are able to work on different parts of the project concurrently with minimal dependencies on each other
2. Modularity where large components can be decomposed into smaller ones which leads to separation of concerns
3. Changes in a layer will not affect other layers
4. Testability

2. Quality Attributes

Testability: Due to loose coupling between subsystems, the testing of individual subsystems improves, hence improving testability of the system.

Maintainability: Due to separation of concerns in functionalities of every class and each subsystem, changes which are made to any class within a subsystem will not affect other classes in other subsystems.

Reusability: Due to the modularity of components, some of the components can be reused in other projects with little or no changes needed. For example:

- The GamePlay component of AdventureLearn can be reused in other similar games, where players need to answer a certain set of MCQ questions within a set time limit.
- The CreateLevel and EditLevel components can be reused in other applications where users can create MCQ questions.

Scalability: Each layer can be scaled independently depending on needs

3. Integrated Architecture

1. Object-Oriented Model
    a. The Object-Oriented Model's concepts were integrated into our model through the use of distinct objects within the Data Access Object and the various Controllers that are used to interface between subsystems.
2. Client and Server
    a. The Client-and-Server architecture will be integrated into our Layered architecture model to handle multiple clients. By enabling multiple clients, it allows our clients to share data on a single server, allowing the clients to execute key functions such as CustomLevel, Leaderboard and Assignments through a central database.

4. Excluded Architecture

1. **Main program and subroutine**
    a. Small scale which does not fit our project
    b. High complexity to implement
    c. Does not support reusability as much as object oriented pattern
    d. Does not provide significant performance improvement
    e. Tight coupling
2. **Pipe and Filter function**
    a. For our project, client to server API calls are minimal which includes a small amount of data which will be processed in a relatively quick timeframe.
    b. No use for concurrent execution within client and server
    c. High complexity to implement

5. External Frameworks

1. Godot and Mono Framework will be used to create the Presentation Layer. Godot is strongly object oriented as the user interface elements such as Nodes, Scenes which are objects. Godot also supports cross-language scripting which means that we are able to code in both GodotScript and C#. After that, the script will be attached to the relevant nodes.
2. Google Cloud SQL will be used to host our MySQL database as it will allow us to effectively implement the database to interact with the Data Access Objects using Dapper.
3. Dapper will be used as our Object-relational mapping to the Google Cloud SQL database. It is an object mapper for .NET framework which provides the framework for mapping an object-oriented domain model to our MySQL database. It is also available free as an open source program. Some of the benefits include the following:
    a. Reduce lines of code
    b. High speed and performance
    c. Easy handling of SQL query
    d. Object mapper
    e. Supports bulk data insert
    f. Supports multiple query
    g. Operating directly on IDBConnection interface