Project Documentation

for

CZ3003 – Software Systems Analysis and Design

Version 2.0

Done by Iced Lemon Tea

Nanyang Technological University (NTU)

Friday, 16 April 2021

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason for Changes** | **Version** |
| Iced Lemon Tea | 10/04/21 | First Draft. | 1.0 |
| Iced Lemon Tea | 12/04/21 | Updating of document. | 1.1 |
| Iced Lemon Tea | 16/04/21 | Finalize version. | 2.0 |

The document is structure in the following order.

[1. Introduction 5](#_Toc69535877)

[1.1 Purpose 5](#_Toc69535878)

[1.2 Document convention 5](#_Toc69535879)

[2. UML Diagram 6](#_Toc69535880)

[2.1 Use Case model 6](#_Toc69535881)

[3. Program Architecture 7](#_Toc69535882)

[3.1 Candidate Architecture style 7](#_Toc69535883)

[3.2 Candidate Architecture 7](#_Toc69535884)

[3.2.1 Top Level 7](#_Toc69535885)

[3.2.2 Second Level for Godot 8](#_Toc69535886)

[3.2.3 Second Level for React and Database 8](#_Toc69535887)

[3.2.4 Second Level for Web Data analytics 9](#_Toc69535888)

[4. Product Subsystem overview 9](#_Toc69535889)

[4.1 Subsystem Division 9](#_Toc69535890)

[4.2 UML Component Diagram 10](#_Toc69535891)

[5. Subsystem in detail 11](#_Toc69535892)

[5.1 Godot Subsystem 11](#_Toc69535893)

[5.1.1 Why Godot 11](#_Toc69535894)

[5.1.2 Role of this Subsystem 11](#_Toc69535895)

[5.1.3 Communication diagram 11](#_Toc69535896)

[5.1.4 Some annotation 12](#_Toc69535897)

[5.2 React Subsystem 12](#_Toc69535898)

[5.2.1 Role of this Subsystem 12](#_Toc69535899)

[5.2.2 Various API <FEEL FREE TO ADD YOUR OWN> 12](#_Toc69535900)

[5.2.3 Subsystem design 12](#_Toc69535901)

[5.2.4 Database design 12](#_Toc69535902)

[5.2.5 Table Usage 13](#_Toc69535903)

[5.3 Data analytics Subsystem 13](#_Toc69535904)

[5.3.1 Technology chosen. 13](#_Toc69535905)

[5.3.2 Role of this Subsystem 13](#_Toc69535906)

[5.3.3 Communication diagram 13](#_Toc69535907)

[5.4 Communication Diagram 14](#_Toc69535908)

[6. Product Testing 14](#_Toc69535909)

[6.1 Blackbox testing 14](#_Toc69535910)

[6.2 Whitebox testing 17](#_Toc69535911)

[6.2.1 Unit Testing in Godot Subsystem Component 17](#_Toc69535912)

[6.2.2 Unit Testing in React Subsystem Component 18](#_Toc69535913)

[6.2.3 Unit Testing in Data Analytics Component 18](#_Toc69535914)

[6.3 Integration Testing 19](#_Toc69535915)

# Introduction

## Purpose

The purpose of this document is to provide a high-level overview and detail of the architecture of our program.

This document will also define the goals of the architecture, the reason behind it, and how it will be implemented.

To provide some context about our project:

[Abstract from Software Requirement Specification]

This product aims to fulfil the following 5 points:

1. Provide students with an alternate way of learning. It is a gamification application whereby students can learn concepts by playing various mini games.
2. Enhance the learning process by making it fun and interactive, and helping learners pay attention and stay focused on the subject.
3. Allows student to review and compare their progress with their fellow schoolmates.

1. Facilitate professor to track the learning process of the students and give added support to students who need help.
2. Professors can leverage on this by looking at how well their students did and specifically where most of them suffer to understand certain concepts. With this data, professors can in turn develop a better way of teaching or perhaps provide more support for students in need.

## Document convention

Below are some document conventions that will be use throughout the document.

* MySQL -Centralized database
* Godot - Fat client game engine (refer to subsystem design for more detail)
* HTTP -Hypertext Transfer Protocol
* Python - Programming language for Data analytics
* Flask - Web framework
* REST - Representational state transfer a type of API (Application Programming Interface).
* SRS -Software Requirement Specification document.
* Game data - All information about the game.
* JSON - A standard file format for data exchange between subsystems.
* World, Section and level: refer to Module , Chapter and Question difficulties, respectively.

# UML Diagram

## Use Case model

# Program Architecture

## Candidate Architecture style

For our system we implemented using “

Independent components with Client and Server communication”

The reasoning behind this is to support the following two points.

1. Co-Current execution
2. Easy to modifiability and re-use due to the loose coupling.

Point 1 help to ensure that we can finish and deliver our project in time.

Point 2 is critical for our design. Due to in an event one of the tools we decide to use could not meet our requirement we could easily remove and replace it with another tool. Given stated in the SRS none of us have the experience in game development. Another benefit is when there an error occur it won`t cause much of a domino effect.  
  
The next subsection will serve to gave an understanding of our program architecture

## Candidate Architecture

### Top Level

First Level Architecture Diagram

React and Database.

Godot

Web  
Data  
Analytics

### Second Level for Godot



Start-up  
Scene



Play preparation  
Scene



World Selection  
Scene



Typing Game   
 Scene



Level Editor  
Scene



Play Scene



API Call



Maze Game   
 Scene



Play Finish  
 Scene

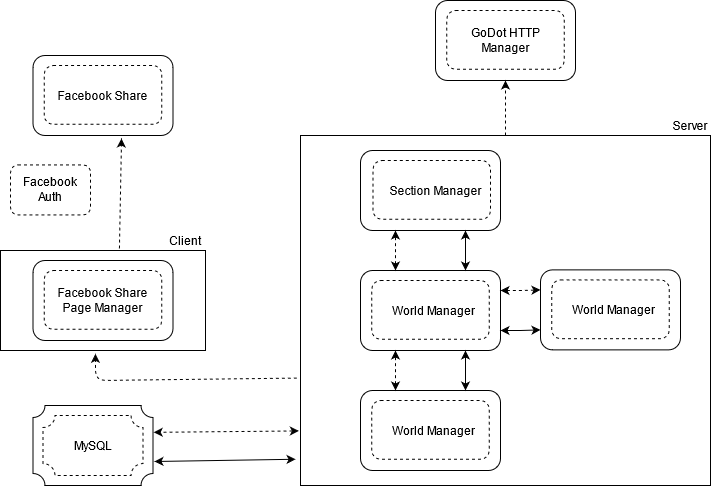


World.json



Section.json

### Second Level for React and Database



### Second Level for Web Data analytics



Webpage ,csv and picture



File System



Path Resolver



Program Function



HTTP Server

# Product Subsystem overview

## Subsystem Division

With reference to our candidate architecture. Our project components are divided and categorize into the following 3 major subsystem.

1. **Godot Subsystem**

It our Fat client where player will interact with our various mini game.

1. **React Subsystem,**

It to supply an API end point to help support Godot Function.

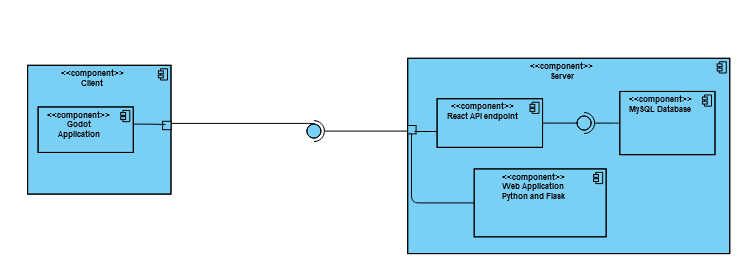
1. **Web Data analytics Subsystem**

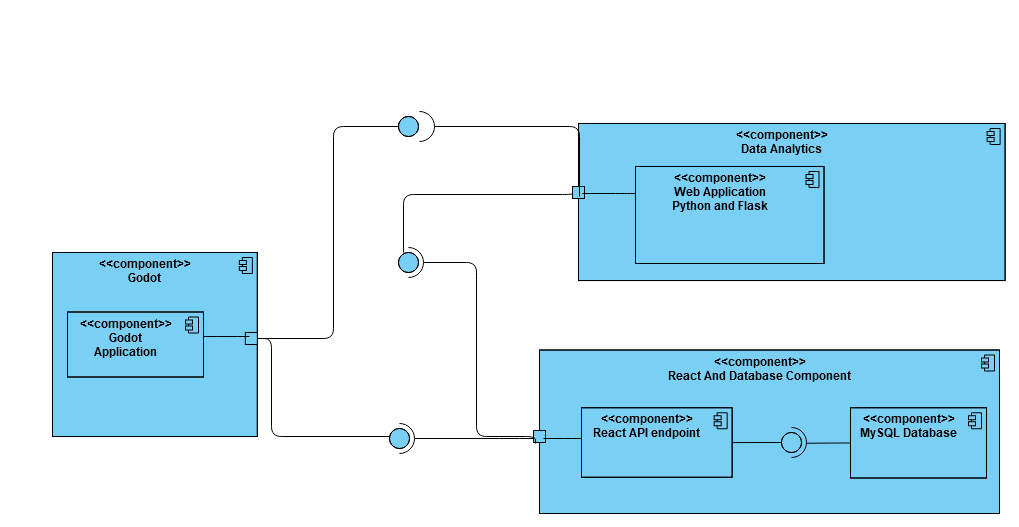
It where certain user with the authorized level to analyse the game data.

The reason for these division is to supply the following three software quality attributes.

1. Portability   
   This would mean certain function our code can be re-use again I.e., retrieving game data from database should we change our game engine Godot. Or we wish to perform another data analytics on a different database but on the same web application while keeping what we have.
2. Maintainability   
   By dividing them into three component it makes maintenance of our code easier due to minimizing the ripple effect.
3. Flexibility   
   If we like to perform certain function that one component could not support, we could easily delegate it to another component via API called. An example will be the “Social media Sharing.”

## UML Component Diagram

Below is our top-level component diagram for our system.

Below is another alternatively if we have another centralized server.

# Subsystem in detail

## Godot Subsystem

### Why Godot

Godot was specially chosen due to it is easy to use with its user Interface ,the depth of its documentation and tutorial available online as compare to other game engine like Unity or Unreal Engine.

### Role of this Subsystem

The aim of this subsystem is to fulfil the following two purpose.

1. Serve as a Fat client that render the game and the creation of Section ,Quiz question and its difficulties, Option available and Explanation.
2. Initiate HTTP (hypertext transfer protocol) request to fulfil our functional requirement in our SRS.  
   1. Communication to React Subsystem is done via HTTP request. Data pass to the React subsystem will always be either in JSON or query string. Data Return from React will always be in JSON.
   2. Communication to the Data analytics is done via the Standard HTTP call. However, no data will be pass between these two subsystems.

### Communication diagram

One communication scenario in this subsystem will be the gameplay and social media sharing afterward.

1.1A:Maze Game

1.1 Load\_mingame()   
Go to either one. At Each time

1: Play Screen

1.1B:Typing Game

1.2 SceneLoader.goto\_scene(XXXXX)

3:Twitter

1.3 OS Shell Open (HTTP)

2: Play Finish

### Some annotation

Each of the component is a scene. One scene can have many nodes and script. Each of the node interact one another to facilitate the game play.

## React Subsystem

### Role of this Subsystem

The purpose of this subsystem is to help perform certain function that our Godot subsystem could not perform due to the limitation of the engine. One notable example will be to connect to a centralized database which it lacks the library to support it.

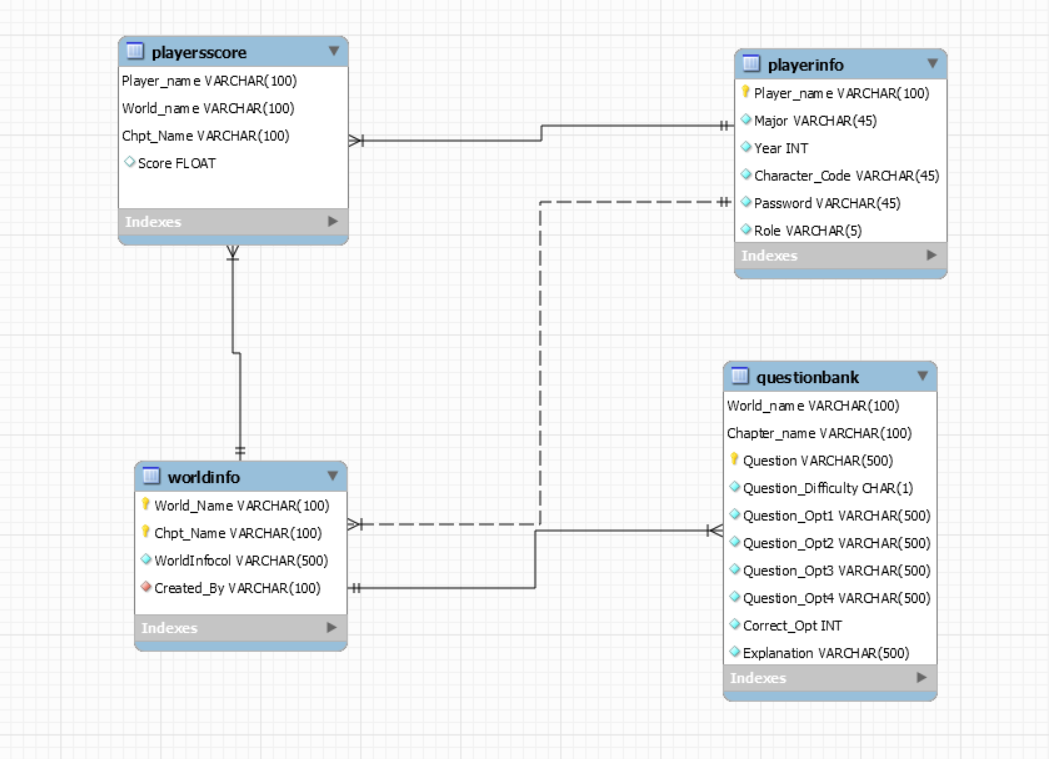
### Various API <FEEL FREE TO ADD YOUR OWN>

### Subsystem design

<**Internal communication within it One scenario>**

### Database design

This section will explain how our database work and support our game.  
Below is our Entity Relationship Diagram (ERD)



### Table Usage

This section will explain what the usage of our tables.

* Playerscore – Will store our player score which will be heavily use by our Data analytics subsystem
* Playerinfo – Will store our player profile as well as it’srole. **It also help function as a mock authentication. Due to our resource limitation**
* Worldinfo—Will store our game world which in our deploy environment is the module and it chapter.
* Questionbank—Will store the respective question and diffcutlies

## Data analytics Subsystem

### Technology chosen.

The technology chosen will be a Python as it a common language to perform data analytics and Flask is chosen for our web framework due to it being lightweight.

### Role of this Subsystem

The purpose of this subsystem is to handle the data analytics requirement stated in our SRS. The reason it is separated is due to the Godot is not designed to supply the necessary tool.

Hence the objectives for this subsystem are:

1. Ensure only Admin can view the analytics graph.
2. Perform Data analysis with the help of visual graph .   
   Some analysis but not limited to are.
   1. Number of submissions by world
   2. Number of submissions by chapter

### Communication diagram

One Interesting scenario for this subsystem will be generating of the graph

1: Main.py

1.Display Analytics()

Path Resolver

1.1 render\_template(analytics.html)

2: Analytic html

## Communication Diagram

The only possible scenario where all 3 subsystems will be involved will be to opening the data analytics from Godot.

1.1 $HTTPRequest.request(xxxxx)

1.1: Login API

1: Login Scene

1.2 ScreenLoader.goto\_scene(xxxx)

2: World Selection scene

2.1 \_on\_LeaderboardButton\_pressed()

2.2 \_on\_LeaderboardButton\_pressed()

3: Data Analytic HomePage

2.A: LeaderboardContainer node

# Product Testing

Testing is critical in our program as it help ensure that our program meet certain qualities. Identify error and ensure that all functional requirements had been met.

However due to the time and technical constraint. We are unable to perform an automated test on all subsystem instead automated testing will be conduct on the React subsystem. While the rest will be conduct manually.

## Blackbox testing

For our Blackbox testing we implement functional testing. To ensure that our program can meet the functional requirement stated in the SRS document.

*<Refer to next page for easy viewing>*

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality to be tested | Expected Result | Actual Result | Verdict |
| Godot Authentication | Credential supplied correctly will be direct to World Selection scene | World Selection scene appear | PASS |
| Credential supplied Incorrectly will be  prompt  Please enter valid username and password | Please enter valid username and password  Prompt on the same scene | PASS |
| Not connect to NTU VPN will be shown  If external device does not connect to NTU VPN | Not connect to NTU VPN will be shown.  Prompt on the same scene | PASS |

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality to be tested | Expected Result | Actual Result | Verdict |
| Creation of Section and Question | Section and its corresponding Question,  Option  Difficulties  Explanation  Insert to database | Section and its corresponding Question,  Option  Difficulties  Explanation  Insert to database | PASS |
| Share the newly create Section And question  To Facebook | A Facebook window will appear letting user to post it to their profile | A Facebook window will appear letting user to post it to their profile | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality to be tested | Expected Result | Actual Result | Verdict |
| Viewing of Leader board | A Leader board will appear with a working link to the Data analytics web application | A Leader board will appear with a working link to the Data analytics web application | Pass |
| Functionality to be tested | Expected Result | Actual Result | Verdict |
| Playing the Game | Mini game will appear  Question is base on the Section I selected  Score will be added accordingly to my input. | Mini game will appear  Question is based on the Section I selected  Score will be added accordingly to my input. | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality to be tested | Expected Result | Actual Result | Verdict |
| Saving Player Score | Once game is ended  The score will be store into the database with it corresponded section and player | Score is saved into the database on the corresponded section and player | Pass |
| Sharing score to twitter | A window will appear.  Allow player to “tweet” their score | A window will appear.  Allow player to “tweet” their score | Pass |

|  |  |  |  |
| --- | --- | --- | --- |
| Functionality to be tested | Expected Result | Actual Result | Verdict |
| Data analytic authentication | Credential supplied correctly will be direct to Profile Page | Redirect to profile page | Pass |
| Credential supplied incorrectly will be prompt.  “Please check your login details and try again”. | On the same page  “Please check your login details and try again”.  Appear below | PASS |
| View Analytics | Non admin will be redirect to history page.  Admin redirect to analytics page with graph being shown | Non admin will be redirected to history page.  While  Admin redirect to analytics page with various graph being shown | PASS |

## Whitebox testing

For our Whitebox testing we perform a manual unit testing where each sub system component system is tested individually during development.

### Unit Testing in Godot Subsystem Component

The test case use for this sub system are as follows:

|  |  |  |
| --- | --- | --- |
| Unit to be tested | Expected Result | Actual Result |
| Godot Authentication | A HTTP post request make to the React API endpoint  With User input pass in a standardize JSON format | A HTTP post request make to the React API end point.  With User input pass in a JSON standardize format |
| Godot Retrieving the Game data | A HTP get request make to the  React API end point | A HTTP get request make to the React API end point |
| Creating the World Section and Question scene | A HTTP post request make to the React API endpoint  With User input pass in a standardize JSON format | A HTTP post request make to the React API endpoint  With User input pass in a JSON standardize format |
| Godot Selection the Section in the corresponding world.  After Which Mini game commence | Game commences  With the correspond question display where user had to play the mini game to answer | Game commences  With the correspond question display where user had to play the mini game to answer |
| Godot Mini game | Game function as needed.  Explanation display when user pick the wrong option.  Appropriate Score had been added in base on player input.  Game ended when timer run out.  Player can select the  option base on the user input  Mini game appears randomly | Game function as needed.  Explanation display when user pick the wrong option.  Appropriate Score had been added in base on player input.  Game ended when timer run out.  Player can select the  option base on the user input.  Mini game appears randomly |

\*Players refer to the student or anyone playing the game.

### Unit Testing in React Subsystem Component

|  |  |  |
| --- | --- | --- |
| Unit to be tested | Expected Result | Actual Result |

### Unit Testing in Data Analytics Component

|  |  |  |
| --- | --- | --- |
| Unit to be tested | Expected Result | Actual Result |
| Authentication | A HTTP post request make to the React API endpoint  With User input pass in a standardize JSON format | A HTTP post request make to the React API end point.  With User input pass in a JSON standardize format |
| Profile Page | Upon successful Login in  It show username and it role | Upon successful Login in  It shows username and it role |
| View Analytics | Non-admin will redirect to history page.  Admin will be direct to Analytic page | Non-admin will redirect to history page.  Admin will be direct to Analytic page |

## Integration Testing

The following manual Integration test had been performed after Integrating. Before making the submission

|  |  |  |  |
| --- | --- | --- | --- |
| Name of Subsystem to Integrated | Functionalities to be Integrated | Expected Result | Expected outcome |
| Godot and  React subsystem | Godot Authentication scene  React Login API | Godot World selection is display if login is successful  or  Message displaying  “Login fail” if login is unsuccessful | Godot World selection is display if login is successful  or  Message displaying  “Login fail” if login is unsuccessful |
| Godot and React subsystem | Creating the World Section and Question scene  Insert API | The respective  World section and its Question is inserted into the database | The respective  World section and its Question is inserted into the database |
| Godot and  React subsystem | Retrieve game data  Select API | The game data is retrieved and store in Json | The game data is retrieved and store in Json |
| Godot and  React subsystem | Share to social media Facebook   React webpage for Sharing | Facebook share had been made | Facebook share had been made |
| Godot and Data analytics subsystems | Godot make an HTTP request to a data analytics  Data analytics authentication | The data analytics web application  will appear prompting user to authentication | The data analytics web application  will appear prompting user to authentication |
| Data analytics  React subsystems | Data analytics function in Web application  React Login API | Data analytics page is displayed  or  Message displaying  “Login fail” if login is unsuccessful | Data analytics page is displayed  or  Message displaying  “Login fail” if login is unsuccessful |