Software Requirements Specification

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 | 27/01/21 | Initial start of SRS. | 1.0 |
| Iced Lemon Tea | 16/04/21 | Editing to standardize final version. | 2.0 |

Table of contents

[1. Introduction 1](#_Toc69519989)

[1.1 Purpose 1](#_Toc69519990)

[1.2 Document Conventions 1](#_Toc69519991)

[1.3 Intended Audience and Reading Suggestions 2](#_Toc69519992)

[1.4 Product Scope 2](#_Toc69519993)

[1.5 References 2](#_Toc69519994)

[2. Overall Description 3](#_Toc69519995)

[2.1 Product Perspective 3](#_Toc69519996)

[2.2 Product Functions 3](#_Toc69519997)

[2.3 User Classes and Characteristics 3](#_Toc69519998)

[2.4 Operating Environment 4](#_Toc69519999)

[2.5 Design and Implementation Constraints 4](#_Toc69520000)

[2.6 User Documentation. 4](#_Toc69520001)

[2.7 Assumptions and Dependencies 4](#_Toc69520002)

[3. External Interface Requirements 4](#_Toc69520003)

[3.1 User Interfaces 5](#_Toc69520004)

[3.2 Hardware Interfaces 9](#_Toc69520005)

[3.3 Software Interfaces 9](#_Toc69520006)

[3.4 Communications Interfaces 9](#_Toc69520007)

[4. System Features 9](#_Toc69520008)

[4.1 Functional Requirement 10](#_Toc69520009)

[4.1.1 Godot authentication 10](#_Toc69520010)

[4.1.2 Creation of the World ,Section and Level 10](#_Toc69520011)

[4.1.3 Selection of World, Section 10](#_Toc69520012)

[4.1.4 Playing the Game 10](#_Toc69520013)

[4.1.5 Saving Game data 10](#_Toc69520014)

[4.1.6 View Leader board 11](#_Toc69520015)

[4.1.7 Data Analytics 11](#_Toc69520016)

[5. Other Non-functional Requirements 11](#_Toc69520017)

[5.1 Flexibility 11](#_Toc69520018)

[5.2 Performance Requirements 12](#_Toc69520019)

[6. Future Work 12](#_Toc69520020)

# Introduction

## Purpose

The purpose of this SRS is to describe the purpose and features of our game product and its expected behaviour along with any constraints faced while designing the system.

If you wish to look at how our project is implemented such as Entity Relationship Diagram , Component , Subsystem or use case model. **Please refer to our “Program document”**  instead

## Document Conventions

The following document convention is used in our Software Requirement Specification

* Admin  
   Refer to the Professor, lecturer or any account that is authorized to perform data analytics.
* Player  
  Refer to student or anyone playing the game without the authorization to data analytics feature.
* User  
  Refer to both Admin and Player.
* World   
  Refer to the course module.
* Section   
  Refer to the individual module chapter.
* Level   
  Refer to various Quiz question difficulties.
* HTTP  
  Refer to the Hypertext Transfer Protocol that is use by our Godot Client to make request.
* Godot  
  Refer to our “fat” Game client engine, that is used to develop the game.
* React Application Programming Interface (API) end point   
  Refer to the API endpoint that help support certain function that our client could not support.
* JSON  
  Refer to the open standard file format use for data interchange.
* Game data  
  Refer to all information regard to the game example are but not limited to “User score”.

## Intended Audience and Reading Suggestions

The intended audience of this document is for the Professor and Student of NTU, Developer of this system, tester, and documentation writer. This document will have information on our Project in a product perspective, it functional and non-functional requirement.

The reader can follow the document in the numbering system shown in the guide. beginning with the overview sections and proceeding through the sections that are most pertinent to each reader type.

For detail about System architecture and its subsystem implementation, refer to the Project documentation instead.

## Product Scope

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.

## References

To help in our project development the following document is being reference.

* Godot documentation   
  <https://docs.godotengine.org/en/stable/index.html>
* React.js   
  <https://reactjs.org/docs/getting-started.html>
* Flask   
  https://flask.palletsprojects.com/en/1.1.x/
* IEEE Software Requirement Specification Template

# Overall Description

## Product Perspective

This product is a new, self-contained product designed to gamify and socialize teaching and learning of software engineering courses in NTU. It is also aimed at helping teachers better know the needs of the students.

## Product Functions

Our product function can be summarized and categorise as follow(s).

Users

1. The game support authentication for its users.
2. The game allows the user to share their new score onto a social media.

Player

1. The game allows players to design their own levels and create their own section.
2. The game allow player to answer questions on various level set by the professors and students in an interactive way.
3. The game allows players to view the leader board for the world.

Admin

1. The game allows analysing of player’s playing history.
2. Data analysis of overall course completion and competency

You may refer to our “Functional requirement” at later section for more detail information.

## User Classes and Characteristics

The product will have two main user classes:

1. Players -Students or anybody playing the game without the authorization level of the Admin
2. Admin – Professor or any teaching staff that had access to the data analytics feature and being able to create a new world.

## Operating Environment

The minimum specifications needed to run the program include:

1. Windows 10 OS.
2. 4 GB RAM.
3. Intel Core2 Duo E8300 @ 2.83GHz or AMD A10 PRO-7800B APU CPU and above.
4. Internet connectivity.
5. Optional: NVIDIA GeForce 6200 or AMD RADEON 9550 GPU and above.

## Design and Implementation Constraints

The product is designed with the following constraints:

1. Inexperience in game development
2. Game is only tested on the latest version of Windows 10.
3. Game development is limited to 8 weeks.
4. Lack of Mono/C# scripts support in deploying to WebGL in Godot.
5. Poor performance of the Light2D node in Godot.
6. Insufficient Godot documentations in Window PC social media sharing and connection to database
7. Lack of comprehensive tutorials/ resources on building a full game with Godot that meet all our functional requirement.
8. Provided with one centralized server.

## User Documentation.

There won`t be any user documentation given to the end user. Because we design our Game UI (User Interface) to be simple.

However, there a project documentation for the developer team/

## Assumptions and Dependencies

The following will be assumed for the development and usage of the system:

1. User is running on a supported version of windows 10.
2. Users of the system shall have a stable internet connection of 100mbps upload/download.
3. User Operating system have a modern web browser pre-installed.
4. The system will not be able to use the NTU login API for account creation/validation.

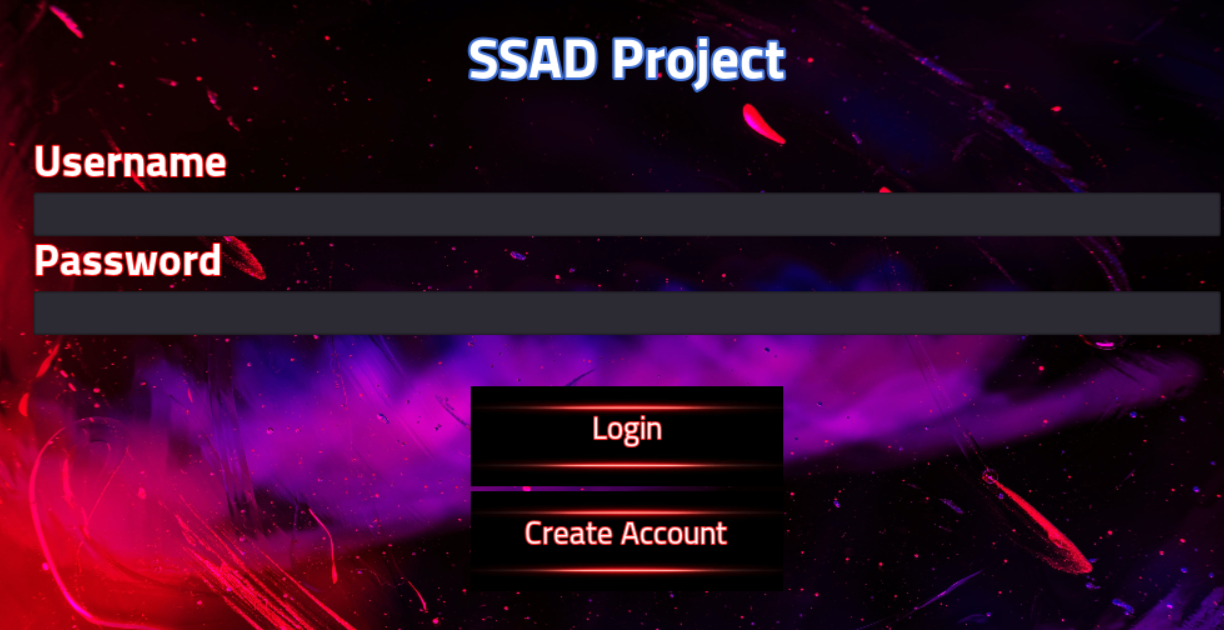
# External Interface Requirements

This section of the SRS describes the interface requirements for the system. Requirements for user, hardware, software, and communication interfaces are defined.

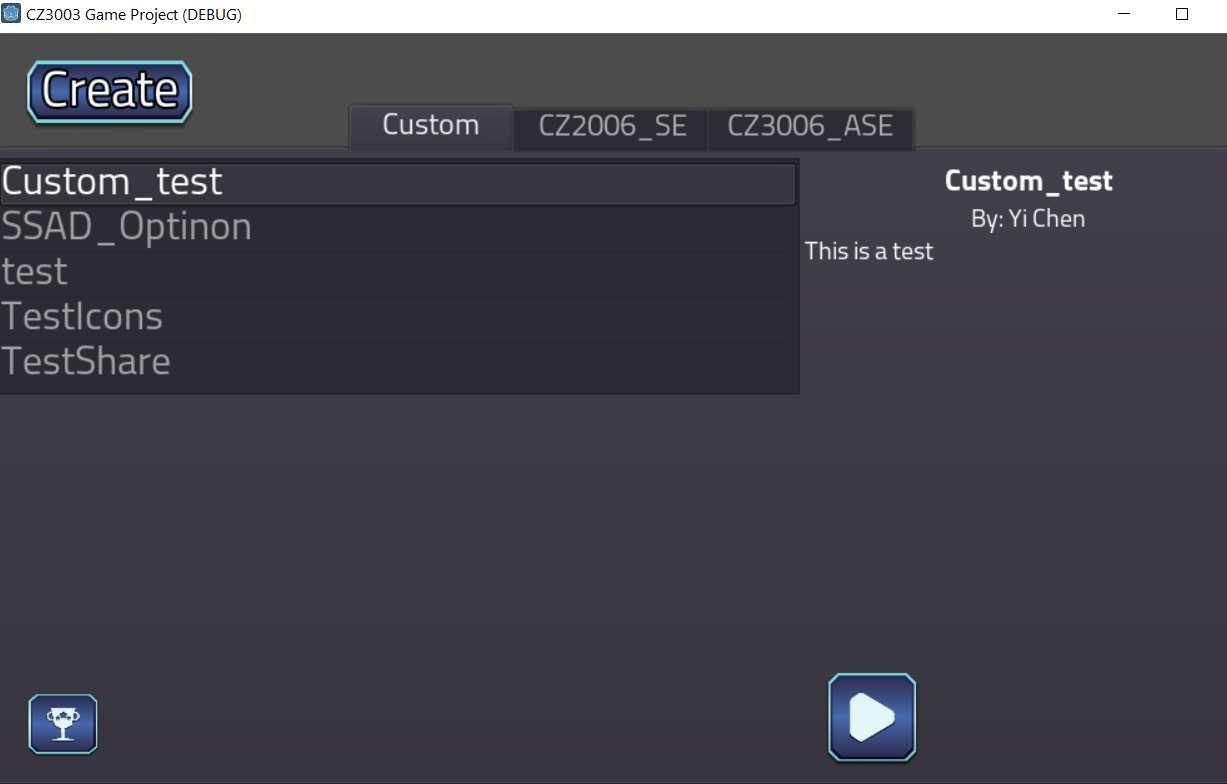
## User Interfaces

This section serves to give our reader a snippet of our game and the main logic behind each scene

This will be the first window that player will see when run the game.

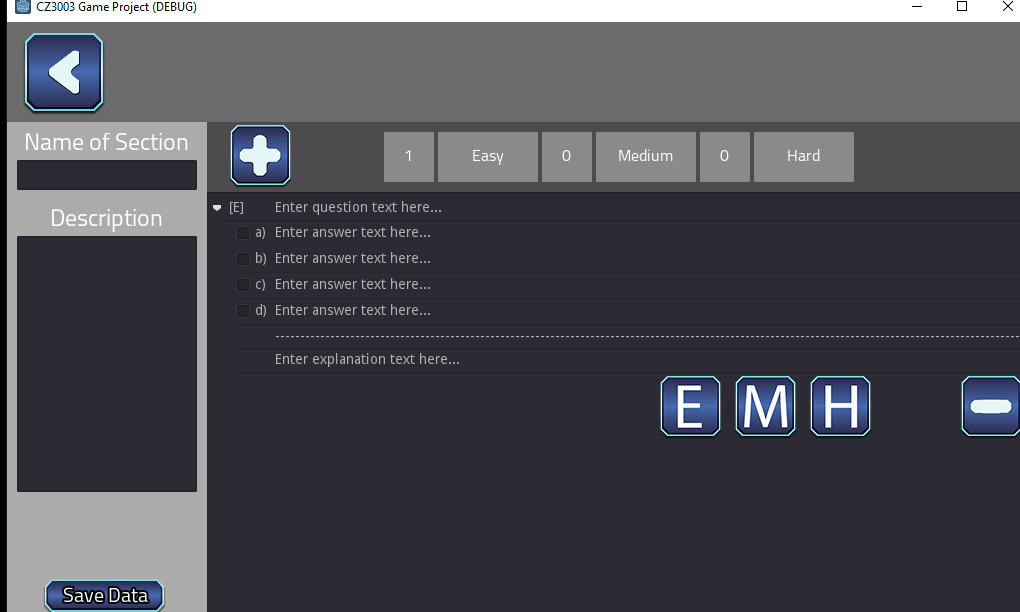


Its main logic is to authenticate the user.

Upon successful authentication this world selection page will appear .

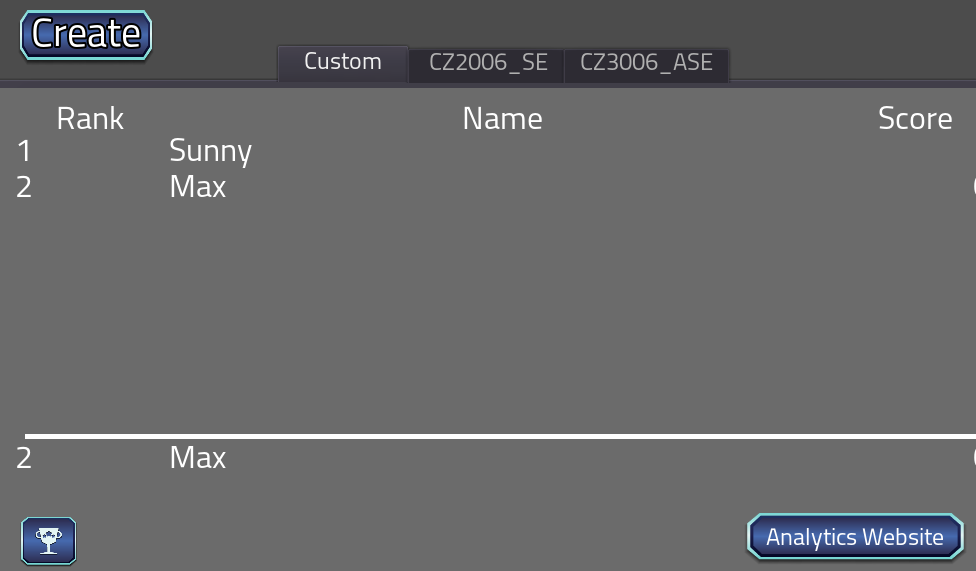
Its main logic to serve as a starting point for our game.

This will be the window if the Player click on create on the world selection page.

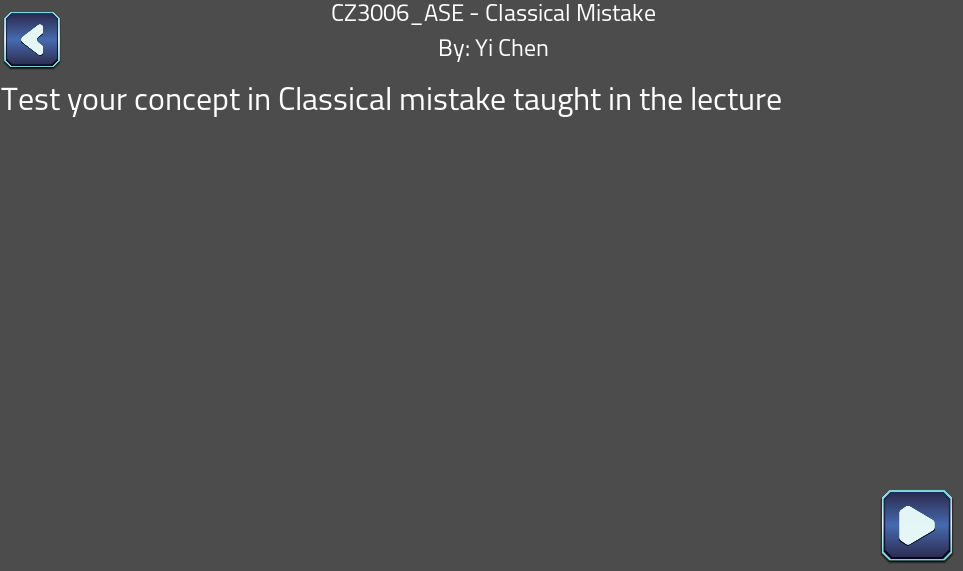


Its main logic is to create the Section and it corresponded question. Upon creation it an option to share to FB will appear

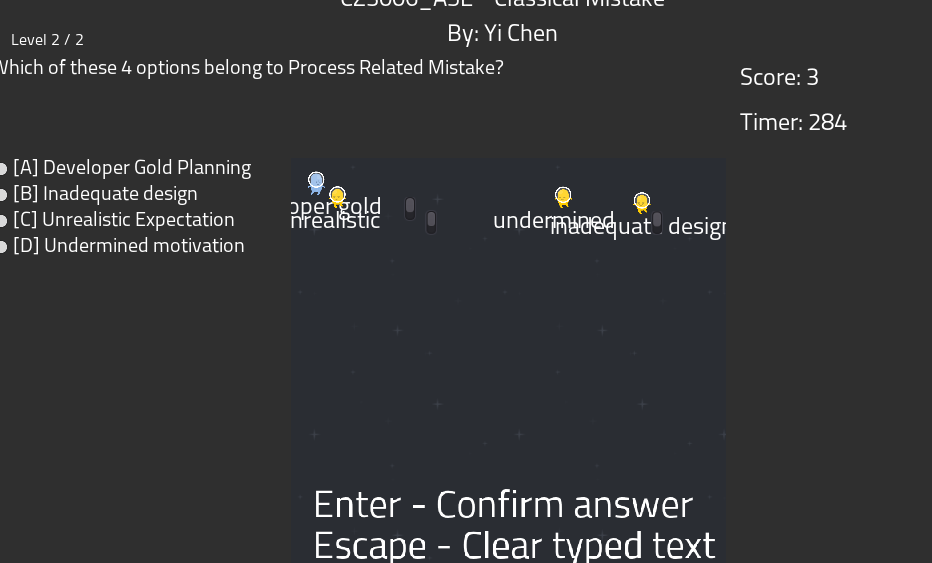
This will be the Leader board snippet.

**

Its main logic is to show who the top players are and to have a link for data Analytics.

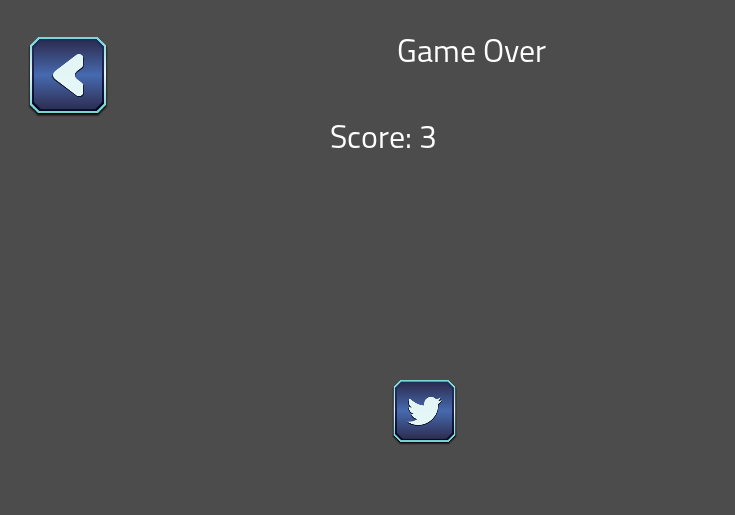
**This is the Game ready window upon press the button on the left the game will commence.

The snippet of one of the mini games

**

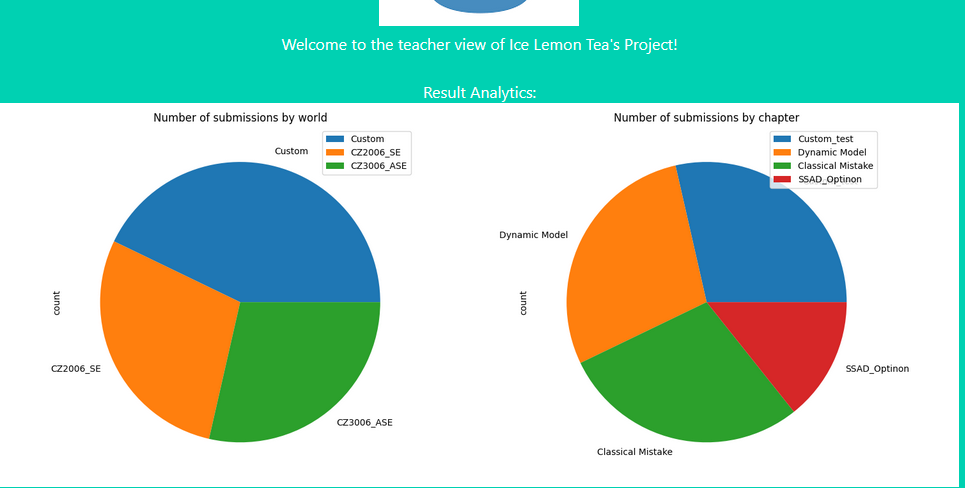
Main Logic is to get the user input base on playing the game.

Upon completion of the game this is what the window will be like



An option to share to twitter will appear.

Below here is a snippet of the data analytics web application.



Main logic is to provide the Admin of the progress of the game.

## Hardware Interfaces

Users will interact with the software through a keyboard and mouse as follows:

1. The mouse will be used for navigation.
2. The keyboard will be used for logging in and gameplay depending on the scenario.

## Software Interfaces

The Godot application will serve as a fat client which will communicate with the other part of our system to meet our functional requirement.

1. The client will use a React-web application for the social media sharing.
   1. Library used will be “react-share”.
2. The client will communicate with React API end point for Database CRUD option
   1. Library used will be “mysql2” on the React API side
3. The client will have a link to our data analytics web application.
   1. No library will be used.

## Communications Interfaces

Our System will have mainly three communications.

1. Godot application will communicate to our centralized database through a HTTP (hypertext transfer protocol) call to the REST-API endpoint   
   1. Data pass to API endpoint is either through a HTTP query String or in a JSON format
   2. Data return to the Godot application from our API endpoint will always be return in JSON.
2. Godot application will also make a standard HTTP webpage call for.
   1. Social media sharing .
   2. Data analytics
3. Connection within our REST API endpoint to the MYSQL local instance.

# System Features

Our system features are as follow(s)

1. Player can learn about their subject topic using game.
2. Users can share their newly created section through social media.
3. Users can create new chapter into custom world
4. Admin can perform data analytics.

## Functional Requirement

### Godot authentication

1. User must be able to authenticate with their username and password.  
   1. If the username or password is incorrect prompt “Please enter valid username and password”
   2. If the authentication is successful, World selection scene will appear.
   3. If external device is not connected to NTUVPN prompt “Please ensure NTU VPN is connected”

### Creation of the Section and Question

1. User must be able to create the Section and the question with various difficulties.  
   1. The field to be entered are “Section name, “Description”, “Question Text”, “four options” and explanation/
   2. Upon successfully creation, an option to share to social media will appear.
      1. Social media to be shared to “Facebook”

### Selection of World, Section

1. Players are able Select the world and it correspond section with the use of the mouse.  
   1. Upon selection. Player would then be able to press on the “Play button: to commence the game.

### Playing the Game

1. Players must be able to answer the corresponding Section Quiz via the use of mini games

.

* 1. The Game will have an overall timer for all the mini games. If the timer reach “0” the game will finish
  2. Score will be updated accordingly to the player respond in the mini game and the question difficulties.
  3. Mini game will be selected randomly.
     1. Mini Games to be done are “Pac-man and Typing game”.

1. Once all of the correspond section question had been answered the game will finish.

### Saving Player Score

1. Player score for that section in that correspond world must be store in a centralized database once the game is over.
2. Player will then be giving an option to share their game score onto the social media.
   1. Social media to be shared to “Twitter”.

### View Leader board

1. Player will be able to view the leader board for that section.   
   1. The leader board must have a link to the Data Analytics Homepage

### Data Analytics

1. Player will need to be authenticated first to perform data analysis on the web application.  
   1. If the credential supplied is Incorrect.   
       Prompt “Please check your login details and try again”.
   2. If the credential is correct it will lead to their profile page
2. Only Admin will be allowed to view stat for the entire overall.  
   1. If a student’s access the view stat page it will lead them to its own analytics page which will be empty.
   2. If Admin access it. They will be display with various graph for analysis.   
      Type of graph to be shown.
      1. Number of submissions by world
      2. Number of submissions by chapter
      3. Boxplot for Score vs Chapter
      4. Average Score vs Chapter for each Player
      5. Boxplot for Score vs Player
      6. Avg Score vs Player for each Chapter
      7. Box plot for Score vs Player
      8. Score vs Player for each world
      9. Boxplot for Score vs World
      10. Score vs World foe each player

# Other Non-functional Requirements

## Flexibility

For our project, the most critical non function requirement is Flexibility. Because we are an inexperienced team. During development it is predict that we might be making a lot more program structure change. Hence, we should minimize the ripple effect and the domino effect as much as possible.

## Performance Requirements

Our game would need to meet the following performance requirement.

1. The Server must support at least 2 players online simultaneously.
2. The response time for displaying the scenario must be less than 1 min.
3. Players’ answers must be submitted and received a response within 2min.
4. Communication within various subsystem should not exceed 3 min.

# Future Work

Due to some constraint. Certain non-critical function was left out intentionally. They are.

1. Picking of Character
2. Creating more mini game- So far, we only supply 2.
3. World Creation
4. Student Analytics such as their own History
5. Creation of account.