Software Requirements Specification

for

CZ3003 Hydra-Defence

Version 3.1 approved

Prepared by Seenivasan Sashwath Ravilla
Mohamed Shafiq Bin Peer Mohamed
Lim Wi Teow
Heather Chew
Koo Jian Yang
Yeong Wei Xian
Tan Wen Jie
Lim Si Yi

Team Hydra

13/4/2021

Contents

Revision History

Name	**Date**	**Reason for Changes**	**Version**
Lim Wi Teow	14/2/2021	Creation of Document	1.0
Mohamed Shafiq	28/2/2021	Adding Diagrams	1.1
Lim Wi Teow			
Mohamed Shafiq	31/2/2021	Updating document to match selected frameworks	1.2
Mohamed Shafiq	24/3/2021	Updating functional requirements	2.0
Lim Wi Teow	13/4/2021	Finalizing Software Requirements Specification	3.0
Mohamed Shafiq		Insert updated diagrams in Appendix	3.1

1. Introduction

1.1 Purpose

This SRS document serves to provide a comprehensive outline of the requirements for development of a learning-based online game system through use cases and other analysis models. As per guidelines from the Teaching, Learning and Pedagogy Division (TLPD) of NTU, the aim is to gamify and socialize teaching and learning of software engineering courses.

1.2 Document Conventions

The document follows the IEEE template for System Requirement Specifications and all body text uses the Times New Roman font with size 12.

1.3 Intended Audience and Reading Suggestions

This document is intended to be read by developers and technical writers who join the project team. It contains a basic introduction to what the system aims to achieve, and the requirements it must satisfy. The appendix has several diagrams that outline the structure of the system as well.

Technical Writers should familiarize themselves with **System Features** to understand the functional requirements of this product. Developers should familiarize themselves with the systems **External Interface Requirements** and then **System Features**.

After reading through this document, developers can refer to the **Diagrams** folder available at the same directory as this file. This will help with understanding the underlying architecture of the system.

1.4 Product Scope

The software being specified is a game of Tower Defense where players navigate through multiple worlds and levels while learning software engineering. The purpose of the game is to provide an engaging learning experience for students and teach them software engineering practices, while keeping teachers updated on student progress. Benefits of the game includes students having a fun time learning while being exposed to software engineering practices and learning how to plan such projects. Professors of the course are also made aware of problem topics and weak points students face in the course, hence able to concentrate effort in said weak topics.

2. Overall Description

2.1 Product Perspective

HydraDefence serves as a tool for teachers to use to increase engagement during delivery of CZ2006 Software Engineering in NTU. This product may eventually be expanded to be included as a part of other courses, as it can increase student interest in topics via a gamified experience.

2.2 Product Functions

Refer to **Appendix Figure 3** for a top-level data flow diagram.
Refer to **System Features** for a more detailed breakdown of required functionality.

The game must be able to perform the following basic functions:

1. Game will prompt players to answer questions based on CZ2006 - Software Engineering.
2. Players must be able to create a custom game and play with one another.
3. Teachers must be able to generate a report for student performance on each topic.
4. Students must be able to share their points earned.
5. Teachers must be able to send invitation links for custom games via Facebook and Twitter.

2.3 User Classes and Characteristics

The game supports the following users and their characteristics.

User	Characteristics
Student	- Students of NTU who are taking the course CZ2006 Software Engineering. - Forms most of the users of the game as it can number up to 100 of them.
Teachers	- Professors teaching the course CZ2006 Software Engineering - Have access to parts restricted to students like data analysis of student’s progress and adjustment of teaching contents
Database Administrator	- An administrator to regulate and adjust question database and score database. - Have access to all parts of the program. - Handles any technical issues that arise.

2.4 Operating Environment

The game is to be developed using Unity game engine and supports multiple desktop operating systems, most notably Windows OS and macOS which are the most common desktop platforms.

2.5 Design and Implementation Constraints

Hardware Limitations:

1. To enable as many students as possible to play the game, system requirements are to be kept low.
2. The game must be able to run at 60fps on a computer with an Intel i5, integrated graphics and 4Gb of RAM.

Specific Technologies:

1. Developers must use Unity as the game engine.
2. The server which handles all incoming and outgoing requests must be made with the Flask Framework.
3. Developers must use MSSQL for database management.

Language Requirements:

1. Game must be displayed in English.

Programming Standards:

1. Developers must abide by SOLID Principles wherever possible.

HydraDefence was developed by following a client-server architecture, where the server is a remote one. Hence, if the server ever reaches maximum capacity or inexplicably crashes, the game will not be able to function as per normal.

2.6 User Documentation

Flask Documentation: CZ3003-Backend\CZ3003-Backend\html\CZ3003-Backend\index.html
Application screenshots: Interface Report.
Video walk-through: Refer to video included in SVN.

2.7 Assumptions and Dependencies

It is assumed that users have connection to the internet as the game requires access to an online database for the game to start. The current database is also designed under the assumption that it will only have to support students from 1 specific module (CZ2006 Software Engineering). Thus, if the game eventually gets expanded to include other modules, the database will likely have to be restructured to include a way to filter by course code as well.

3. External Interface Requirements

3.1 User Interfaces

All error messages will be displayed on screen, either through popups or warning messages. The layout and contents of these messages will be consistent across all devices since the behavior is defined by Unity instead of the host device.

Refer to the **Interface Report** for more information on different screens and their respective explanation.

3.2 Hardware Interfaces

The user will be mainly interacting with the game using a keyboard, mouse, and monitor. While there are no hard requirements for these peripherals, we recommend at least a 720p monitor with a conventional 16:9 layout for optimal performance.

3.3 Software Interfaces

HydraDefence was developed using various frameworks. We used the Unity Framework for the game layer, Flask as the controller to manage the backend logic, as well as Microsoft SQL Server as a database to store all game details. We chose these three software interfaces as they are versatile and can be easily integrated into with each other.

Unity was selected as the game engine due to its open-source nature, which reduced the long-term operating costs of running this game.
The system uses MSSQL **(version 15.0.2080.9)** for local data storage. Every request made by the front-end layer of our website that consumes or produces data will be handled by a cursor object. This cursor object will be created in Python, in the back-end layer. This cursor is created via the Python pyodbc library, which is a library used to interface with common SQL Databases.

3.4 Communications Interfaces

HTTP protocol standards will be used when communicating to the server via the Internet. Some examples of these communications are when students retrieve quizzes before starting a level, or when students play custom games against one another. To support HTTP protocols well, the Flask Server will be certified, so the HTTPS protocol will be used, thereby improving security.

4. System Features

4.1 Homepage

4.1.1 Description and Priority

Description: The system (game) must have a homepage.
Priority: Medium.

4.1.2 Stimulus/Response Sequences

No stimulus required for the landing page of a game.

4.1.3 Functional Requirements

1. The homepage must have a theme.
2. The homepage must have a title.
3. The homepage must allow users to sign in or create an account to play our game.

4.2 Sign in/Create account

4.2.1 Description and Priority

Description: At the homepage, users must be able to sign in using their Student ID and password and new users must be able to create an account.
Priority: Medium

4.2.2 Stimulus/Response Sequences

Stimulus	**Response**
User clicks on login	System displays the search page if their username and password were correct.
User clicks on Don’t have an Account?	The system shows the fields that must be filled up (username, password, etc).
User clicks on Create an Account	System attempts to create an account with the information in each field. If an error is found, the website creates an alert to tell the user what the error is (eg: password length is insufficient)

4.2.3 Functional Requirements

1. System will prompt all users to select a Username and a Password.

   a. If the user selected the sign-up option for student, it would prompt the student to select from 1 of the 4 available avatars.
   b. These avatars must be visually distinct, with obviously different color schemes.

4.3 World Map and Leaderboard

4.3.1 Description and Priority

Description: The world map allows students to choose the world they want to play and view the leaderboard.
Priority: High.

4.3.2 Stimulus/Response Sequences

Stimulus	Response
User successfully logs in	System will display the user's points and options to go to Adventure Mode, PVP, complete an Assignment, view the Leaderboard or quit.

4.3.3 Functional Requirements

1. Students must be able to choose the world from the world map.
2. Students must be able to click on 'view leaderboard'.
   a. Students will be able to see the top 10 highest scores, along with their own score and position.
   b. If they are within the top 10, then only those 10 players and their points will be shown.

4.4 Worlds, Sections and Levels

4.4.1 Description and Priority

Description: The world map will have multiple worlds for users to progress through. Each world is based on a topic in Software Engineering and has multiple sections and levels in it.
Priority: High

4.4.2 Stimulus/Response Sequences

Stimulus	Response
User clicks on a world	System displays all the sections inside a world, which represent a subtopic for a Phase in the Software Development Lifecycle.
User clicks on a section	System displays all the levels inside a world, which represent a different level of difficulty (both in terms of quiz questions and enemy strength).

4.4.3 Functional Requirements

1. Each world has at least 2 sections in it.
    a. Each section in a world must be based on the same topic.
    b. Each section must focus on different parts of the same topic.
        i. System will assign a proficiency level to each player, ranging from 1 to 3.
        ii. This proficiency level will be tied to the percentage of questions the user has gotten right along the course of the game.
        iii. Proficiency levels will fall within 1 of 3 ranges, “beginner”, “intermediate” and “advanced”.
        iv. All users start with 1 proficiency level.
        v. If a user’s proficiency level = 2, they get moved into intermediate.
        vi. If a user’s proficiency = 3, they are moved into advanced.
    c. System must be able to ask questions at each level depending on the proficiency level of the user.
        i. In quiz (which must have exactly 4 questions), students are asked 2 questions with a difficulty based on the level they are currently playing.
        ii. The remaining 2 questions will be asked based on the proficiency of the user.
    d. System will ask students questions before each level in each world to give them currency they can use to buy weapons in the game.
        i. These questions will be extracted from a database of questions.
        ii. Questions will be segmented by topic, section, then difficulty.
        iii. Questions will either be “easy”, “intermediate” or “advanced”.
        iv. Each question carries 1 point, which translates to $100 in the game.

4.5 Leaderboard

4.5.1 Description and Priority

Description: Students can view details of the leaderboard which displays the top few players with the highest points.
Priority: High

4.5.2 Stimulus/Response Sequences

Stimulus	Response
User clicks on Leaderboard	System displays top 10 players in descending order and the points and position of user currently logged in.

4.5.3 Functional Requirements

1. Leaderboard must display the top 10 players and their corresponding points.
2. Leaderboard must be up to date whenever it viewed. It cannot be updated on a fixed interval and instead is updated each time a user clicks on the View Leaderboard option.

4.6 PVP

4.6.1 Description and Priority

Description: Students can choose to play in the PVP mode to compete with other students.
Priority: High

4.6.2 Stimulus/Response Sequences

Stimulus	Response
User clicks on PVP	System displays options to Create a Room, join a Room, Back
User clicks on Create room	System generates a random access code and displays it. Student then returns to the previous screen
User clicks on join a room	System displays an input box to enter an access code. Student enters it and enters the room if there are no errors.

4.6.3 Functional Requirements

1. Students must be able to compete with other students regardless of their current proficiency level.

2. The points they can use at each level is determined by the points they earn during the quiz completed right before the level starts. If the student wishes to compete with someone outside of their level, they should be free to. These scores do not count towards the leader board.
3. Game has a custom PVP option.
4. Topic for questions asked during the game are determined by the world the custom game is set in.
5. Question difficulty is set by the user when they choose the level they want the PVP round to be based on.
6. User must be able to create a room.
    a. System will generate the map based on the selected world, section and level.
    b. System generates a unique access code.
    c. Other user must enter the access code to join the custom level.

4.7 Performance Page

4.7.1 Description and Priority

Description: The performance page allows teachers to view the performance and progress of the students in the game.
Priority: High

4.7.2 Stimulus/Response Sequences

Stimulus	**Response**
Teacher clicks on Quiz Report	System displays performance metrics of quizzes for all worlds and sections.
Teacher clicks on Assignment Report	System displays performance metrics of an assignment that teacher chooses.

4.7.3 Functional Requirements

1. Teachers must be able to view a report on the performance page of a particular world generated by the system.
2. The report must include the following:
    a. Number of questions answered per level.
    b. Number of questions answered correctly per level.
    c. Top 3 category of questions answered wrongly per level.

      i. Questions from these 3 categories must have been answered by at least 20% of students registered in that academic year.
      ii. These 3 categories are determined by the percentage of participants that got it wrong.

3. Teachers must be able to view reports for quiz performance and assignment performance separately.
4. The quiz report must include the following information:
    a. World number
    b. Section number
    c. Average performance across all 3 levels of difficulty
5. The assignment report must include the following information:
    a. Access Code of the assignment.
    b. Start and End time for the assignment.
    c. Average Score.
    d. Standard deviation.

5. Other Nonfunctional Requirements

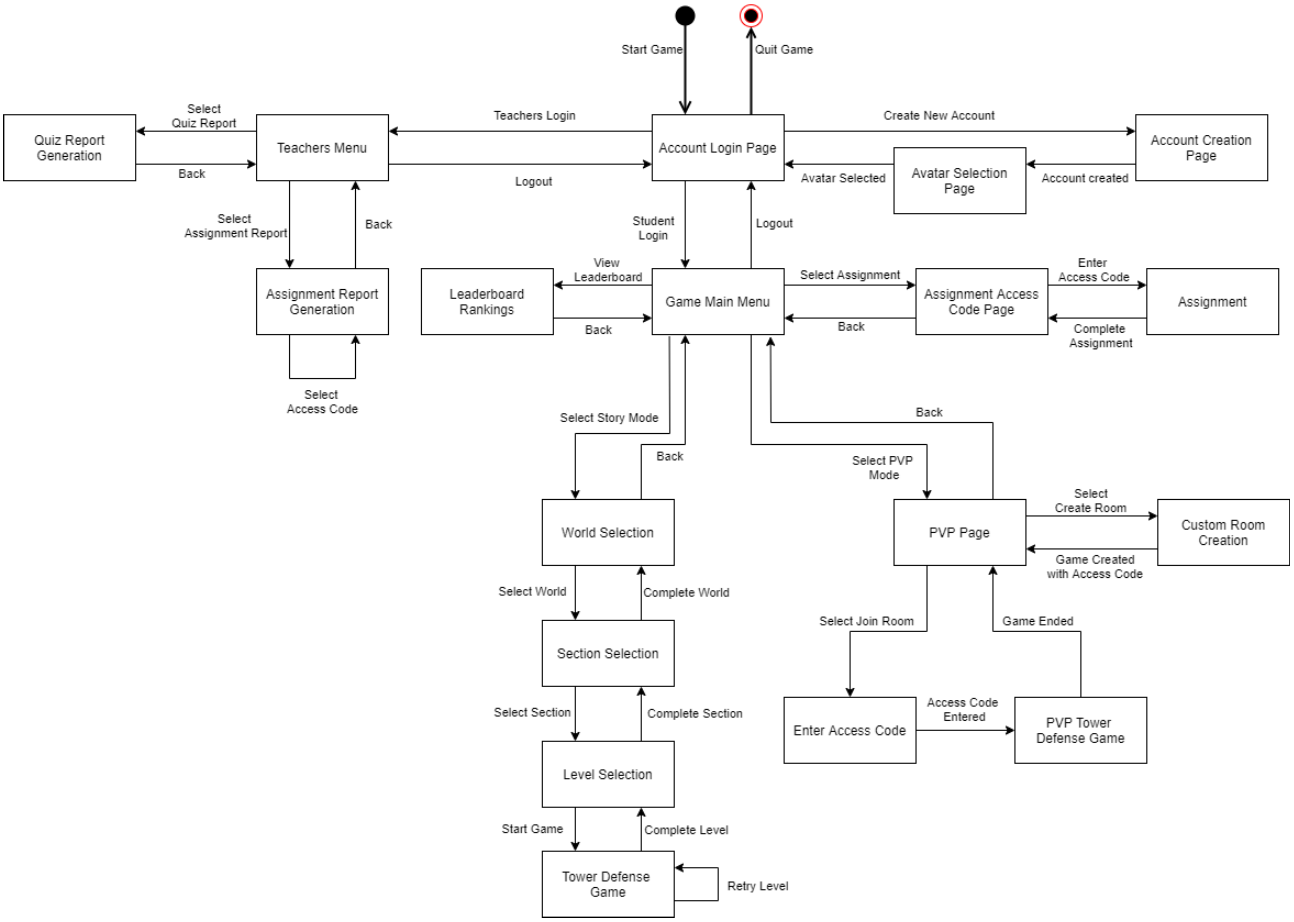
5.1 Performance Requirements

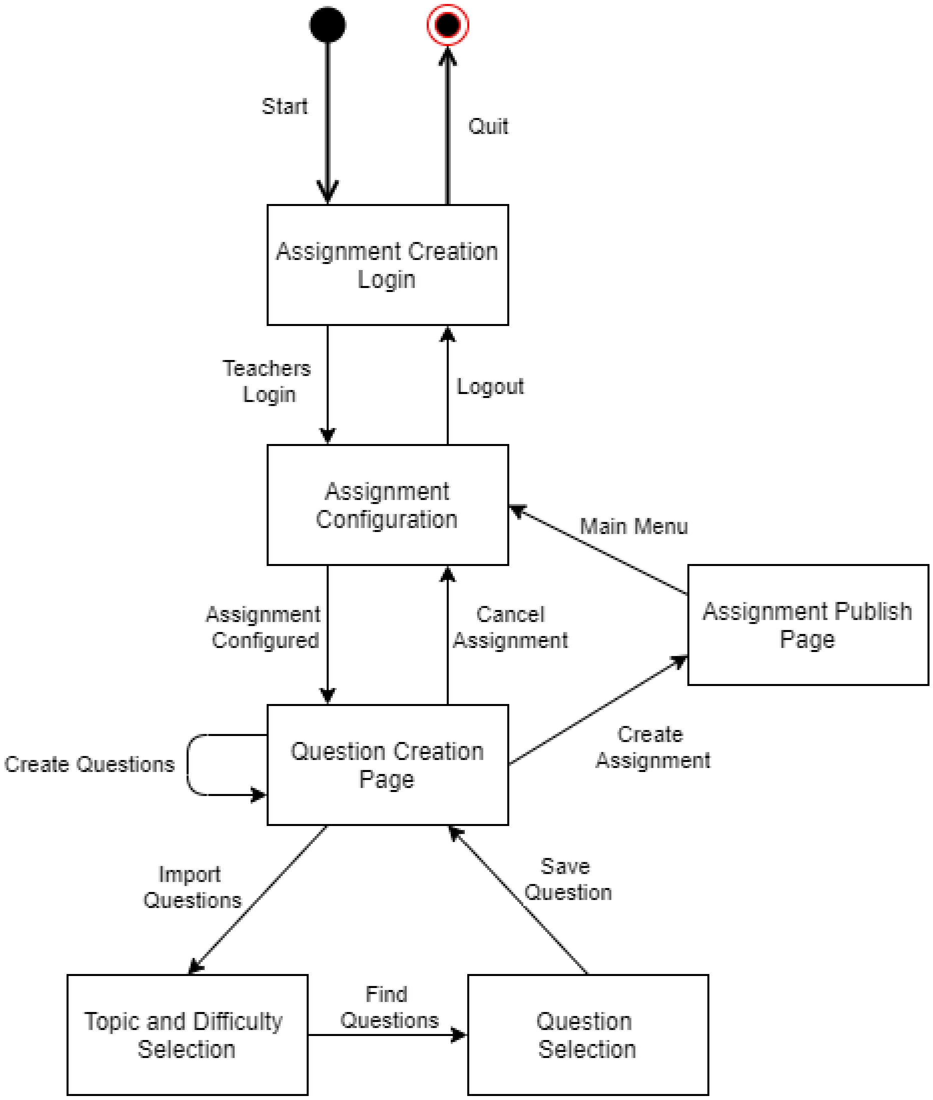
1. The game must be able to handle over 100 numbers of users at any point.
2. When a user signs in, the system must bring them to the homepage within 5 seconds.
3. When a game finishes, the leaderboard will be updated within 10 seconds.

- Appendix A: Glossary

Term	Definition
World	Each world represents a phase of the software development life cycle.
Section	Sub-topic that a student can choose to play.
Level	Difficulty of a game. This affects the gameplay as well as the quiz questions.
Quiz	Set of questions a student has to answer based on their world, section and level selection.
Proficiency level	A measure of a user's competence. Ranges from 1-3.
Story Mode	The main mode that students will play to increase their knowledge on the subject and accumulate points.
PVP	The mode in which students can create their custom game and challenge another friend.
Create Room	Formed when a user creates a custom game. An access code will be generated.
Join Room	The option to enter a custom game using an access code that has been previously generated.
Access Code	A 4-digit number that is generated when a custom pvp game is created. This number is to be entered on both players' devices to start the custom game.
Leaderboard	A table that shows the top 10 students by points including the current student's points and position.
Assignment	A mode in which students can complete assignments created by their teachers.
Quiz report	An overview of the students' performance in the quizzes for the various sections.
Assignment report	An overview of the students' performance in the assignments. Performance can be viewed for each individual assignment.
Avatar	An icon for the student to choose upon account registration.

- Appendix B: Analysis Models




Start
Quit
Assignment Creation Login
Teachers Login
Logout
Assignment Configuration
Main Menu
Assignment Publish Page
Assignment Configured
Cancel Assignment
Create Questions
Question Creation Page
Create Assignment
Import Questions
Save Question
Topic and Difficulty Selection
Find Questions
Question Selection

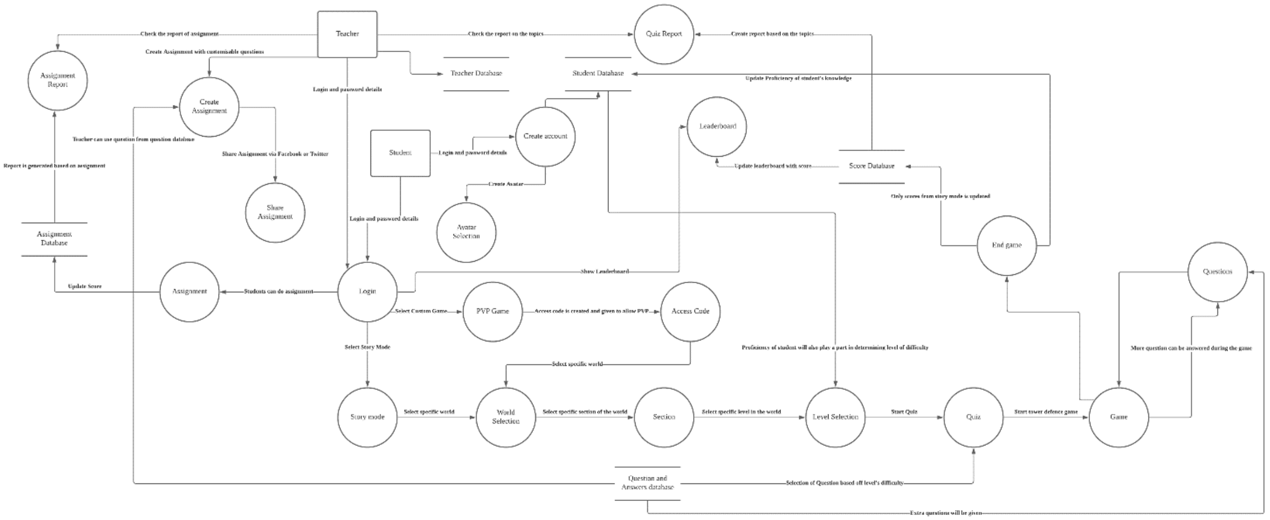

Teacher
Check the report of assignment
Check the report on the topics
Quiz Report
Create report based on the topics
Create Assignment with customisable questions
Assignment Report
Create Assignment
Teacher Database
Student Database
Update Proficiency of student's knowledge
Login and password details
Leaderboard
Teacher can use question from question database
Student
Login and password details
Create account
Update leaderboard with score
Score Database
Report is generated based on assignment
Share Assignment via Facebook or Twitter
Create Avatar
Only scores from story mode is updated
Share Assignment
Avatar Selection
Login and password details
Assignment Database
End game
Show Leaderboard
Questions
Update Score
Assignment
Students can do assignment
Login
Select Custom Game
PVP Game
Access code is created and given to allow PVP
Access Code
Select Story Mode
Proficiency of student will also play a part in determining level of difficulty
More question can be answered during the game
Select specific world
Story mode
Select specific world
World Selection
Select specific section of the world
Section
Select specific level in the world
Level Selection
Start Quiz
Quiz
Start tower defence game
Game
Question and Answers database
Selection of Question based off level's difficulty
Extra questions will be given

Figure 3 - Data Flow Diagram