Software Requirements Specification

for

AdventureLearn

Prepared by Team NanyangGuys

CZ3003 Software System Analysis and Design

Table of Contents

1.Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of AdventureLearn, a mobile phone game application which aims to enrich the learning experience of software engineering courses by gamifying and socializing the teaching and learning of the teaching content.

1.2 Document Conventions

The following conventions are used in this document.

	Font type	**Font Size**
Heading	Arial	23
Subheading	Arial	14
Main	Arial	11

1.3 Intended Audience and Reading Suggestions

This document is intended to be read and used by the developers, project managers, users and testers of the application. The reading sequence of this document should begin from the Overall Description section for first time users but developers, project managers and testers may choose to proceed to the External Interface Requirements sections that provide more pertinent usage.

Software Developer	Software developers will need to take reference from and update this software requirement specification when making improvements or modifying the application to better comply with the stated functional requirements.
Project Manager	Project manager will need to take reference from this software requirement specification to understand the progress of the project and ensure the project is in line with the time requirements.

Users (students and teachers)	Users will need to take reference from this software requirement specification to understand the application's functions and limitations.
Testers	Testers will need to take reference from this software requirements specification in order to prepare optimal test cases to verify the functionality of the application based on the requirements specified.

1.4 Product Scope

AdventureLearn is a social game on mobile devices that aims to gamify and socialize teaching and learning of software engineering courses. The goal is to increase students' learning interests and interactions, thus enhancing their effectiveness and efficiency of learning.

1.5 References

IEEE Template for System Requirement Specification Documents:
https://goo.gl/nsUFwy

2. Overall Description

2.1 Product Perspective

AdventureLearn was developed as an attempt to introduce a social game on mobile devices to gamify and socialize teaching and learning of software engineering courses. Students can learn and compete with each other via playing the game, and teachers can assess the mastery of course via data analysis.

2.2 Product Functions

The main functions of the product are:

- Login to the account through google play service
- Select character
- View created level
- Create level
- View challenge inbox
- Send challenge
- Play challenge
- View assignments
- Complete assignment
- Select world
- Select section
- Select level
- Play level
- View leaderboard
- Publish assignments
- Generate summary report
- View class
- Create class

2.3 User Classes and Characteristics

- Students should be able to do the following functions:
    - Select character
    - Select world, section, and level
    - Play level
    - Create custom levels
    - Send and complete challenges from other students
    - View and complete assignments given by teachers
    - View the leaderboards

- Teachers should be able to do the following functions:
    - Create assignments for students to do
    - Generate a summary report on students' playing history
    - Create classes

2.4 Operating Environment

- Android operating system Version 8 and above
- Dependencies
    - facebook-android-SDK: 5.4.0.
- Database
    - Google Cloud SQL

2.5 Design and Implementation Constraints

AdventureLearn makes use of the following APIs to operate:

- Facebook API is used for user login authentication.
- RESTful API is used for manipulating and retrieving relevant data from the database

AdventureLearn makes use of external library

- MySQL library is used for connection and queries to the Google Cloud SQL database.

2.6 User Documentation

As per the scope of the project, this document will serve as documentation for the users to understand the functionality of the application.

2.7 Assumptions and Dependencies

The following table describes the assumptions made during the operation of the application.

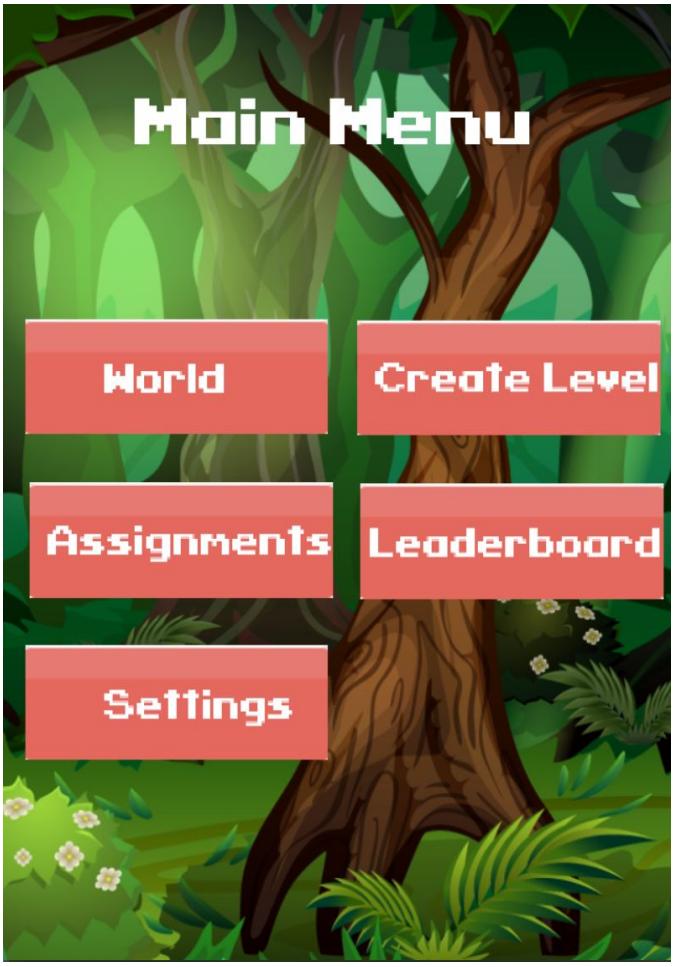
- There will be no service interruptions in online database
- Users have Internet connection.
- Users are sufficiently proficient in English.

3. External Interface Requirements

3.1 User Interfaces

The User Interface shall follow Android Material Design.

Launch page / Login Page
Student can login to AdventureLearn through facebook login authentication.



Menu Page

Main menu landing page for Student who successfully logged into AdventureLearn. Students will be able to access different functionalities in the game through Main Menu.

Select World

- Worlds will be locked according to the user's playing progression.
- Student will be able to select from a series of different world to play.

Select Section / Select level

- In every world, Student will be able to select from a series of section and levels with each section and level getting progressively more difficult.
- Level will be locked according to the user's playing progression.
- Completed levels will show the Student's stars earned for that level

Play Level

Every question relating to the section will be in MCQ format with 4 options.
Student must choose an answer for the question.

Level Completed
If user successfully cleared the level, this page will be displayed.

Create Challenge

Students will be able to design their own level with custom questions and answers to challenge each other.

Leaderboard

Student will be able to access the leaderboard to view their own personal rankings and the top 10 players for each world.

Main Menu (Teacher)
Main menu landing page for Teacher who successfully logged into AdventureLearn.
Teacher will be able to access different functionalities in the game through Main Menu.

Create Class
Teacher will be able to create class and add multiple Students to a class.

Create Assignment
Teacher will be able to create assignment and publish the assignment to selected class.

View Class Performance

Teacher will be able to view the class performance through data visualisation charts for each world or section to understand the class's students overall mastery of the course.

View Student Performance

Teacher will be able to view the student performance through data visualisation charts for each world or section to understand the student overall mastery of the course

3.2 Hardware Interfaces

- Mobile Phones
    - Any mobile phone device capable of running Android operating system.
- Tablets
    - Any tablet device capable of running Android operating system.
- Devices using phone emulator
    - Any device operating application through a phone emulator requires an Android operating system.

3.3 Software Interfaces

- Operating System
    - Android operating system is selected for its compatibility with devices and ease of entry to market.
- Social media
    - Facebook is selected as the platform for user authentication and share achievements.
- Godot
    - Open source game engine used for software development.

3.4 Communications Interfaces

- HTTPS Communication between Application and Database (Port: 443)
- Google Cloud SQL(Port: 3306)

4. System Features

4.1 User Login

4.1.1 Description and Priority

Users can login with their Facebook/Twitter/Gmail account.
Priority: High

4.1.2 Stimulus/Response Sequences

Stimulus: The user enters a valid username and password to login.
Response: The user is authenticated and the application's Homepage is displayed.

Stimulus: The user enters either an invalid username or password.
Response: The system will display a message, "Wrong username or password, please try again".

4.1.3 Functional Requirements

1. Students and teachers must login to the system before the game can be accessed.
    1.1. The student and teachers must be able to login through several platforms.
        1.1.1. Students and teachers must be able to login with his/her Facebook account.
        1.1.2. Students and teachers must be able to login with his/her Twitter account.
        1.1.3. Students and teachers must be able to login with his/her Gmail account.
    1.2. The system must validate that the required fields are filled up.
    1.3. The system must validate that the login credentials are valid.
    1.4. The system must display an error message if the login credentials are invalid.

4.2 Select Game Character

4.2.1 Description and Priority

Different characters can be chosen, each having a unique perk.
Priority: Medium

4.2.2 Stimulus/Response Sequences

Stimulus: The user clicks on “Select Character” on the Homepage.
Response: The system will displays the available characters for selection.

4.2.3 Functional Requirements

1. The system must provide the student with 4 different characters to choose from.
    1.1. Each character has a unique perk.
        1.1.1. The first character will have a 25% chance to get 1 reduced option.
        1.1.2. The second character will have a 25% chance to deal double damage.
        1.1.3. The third character will have a 25% chance to take no damage from answering incorrectly.
        1.1.4. The fourth character will have a 25% to heal 25 health points.

4.3 Game Mechanic

4.3.1 Description and Priority

The game consists of a series of worlds to be explored, each representing different phases of the life cycle of software engineering, ranging from requirements engineering and architectural design to implementation and software testing. Each world consists of several sections, each representing specific topics of each phase from basic ones to advanced ones. Each section consists of several levels, each representing questions relevant to specific topics with increasing difficulties.
Priority: High

4.3.2 Stimulus/Response Sequences

Stimulus: The user clicks on “Select World” on the Homepage.
Response: The system displays the available worlds.

Stimulus: The user selects a world.
Response: The system displays the available sections of the selected world.

Stimulus: The user selects a level.
Response: The system starts displaying the MCQ questions of the selected level.

Stimulus: The user completes a game level.
Response: The system rewards the user with points according to the difficulty of the level and records the level as “Completed”.

Stimulus: The user fails the game level.
Response: The system gives the message “Level failed. Please try again!” and records the level as “In Progress”.

4.3.3 Functional Requirements

1. The system must provide a series of worlds for the student to explore.
    - 1.1. Each world must represent the different phases in the life cycle of Software Engineering: Requirements Engineering, Architectural Design, Implementation, and Software Testing.
    - 1.2. Each world in the game must have several sections, each representing the specific topics of each phase from basic ones to advanced ones.
        - 1.2.1. Each section must have several levels, each representing questions relevant to specific topics with increasing difficulties.
        - 1.2.2. For each level, the student must correctly answer 5 questions correctly.
            - 1.2.2.1. Every question will be in MCQ format with 4 options.
            - 1.2.2.2. In each level, the student’s character will start with 100 health points.
            - 1.2.2.3. In each level, there will be a monster with 100 health points.

1.2.2.4. Each time the student answers the question correctly, the student's character must launch an attack at the monster which takes away 20 health points from the monster.

1.2.2.5. Each time the student answers the question wrongly, the monster must launch an attack at the student's character which takes away 10 health points from the student's character.

2. The system must use a 3-star ranking system for each level to indicate mastery level.
   2.1. The stars must be awarded based on the remaining health points after a level has been cleared.
      2.1.1. For 0% health remaining health points, no stars shall be awarded.
      2.1.2. For 1% to 50% remaining health points, 1 star shall be awarded.
      2.1.3. For 51% to 79% remaining health points , 2 stars shall be awarded.
      2.1.4. For 80% and above remaining health points, 3 stars shall be awarded.

4.4 Game Leaderboard

4.4.1 Description and Priority

To keep students engaged, a leaderboard is used to display the total points attained by a player and ranking those players by points obtained.

Priority: Medium

4.4.2 Stimulus/Response Sequences

Stimulus: The user clicks on “Leaderboard” on the Homepage.
Response: The system displays the leaderboard.

4.4.3 Functional Requirements

1. The system must have a leaderboard.
    1.1. The students must be able to view every student's score.
    1.2. The leaderboard must be updated every 5 seconds.
    1.3. Name must be displayed in the following format: <Last Name> <First Name>.
    1.4. Score must be displayed in numerical characters.
    1.5. Rank must be displayed in numerical characters.

4.5 Game Level Design

4.5.1 Description and Priority

Students can create their own levels and send them as a challenge to other students. Teachers can create their own levels and send them as an assignment to different class groups.

Priority: High

4.5.2 Stimulus/Response Sequences

Stimulus: The user clicks on “Create Level” on the Homepage.
Response: The system displays the Level Creation page.

Stimulus: The student selects a created level and clicks on “Send Challenge”.
Response: The system displays a list of the students to select.

Stimulus: The teacher selects a created level and clicks on “Send Assignment”.
Response: The system displays a list of the class groups to select.

4.5.3 Functional Requirements

1. Users must be able to design their own levels.
    1.1. Students must be able to send their custom made levels to other students to challenge them.
    1.2. Teachers must be able to send their custom made levels to class groups as assignments.
        1.2.1. Teachers can notify the students through Facebook/Twitter.

4.6 Data Analysis

4.6.1 Description and Priority

The system analyzes each student's playing history. The results are used to adjust the difficulty of the following levels, and also serve as an indicator of the overall mastery of the course.

4.6.2 Stimulus/Response Sequences

Stimulus: The teacher clicks on "Summary Report" on Homepage.
Response: The system displays the data analytics of students' playing history.

4.6.3 Functional Requirements

1. The system must be able to analyze each student's playing history to continuously obtain individual and overall mastery of the course in real-time.
    1.1. Based on the analysis results, the questions of the following levels can be customized.
        1.1.1. Each level has 3 difficulties: Easy, Normal, Hard.
            1.1.1.1. For students who have achieved an average of 2.5 to 3 stars for the previous 5 levels, the next level will be at Hard difficulty.
            1.1.1.2. For students who have achieved an average of 2 to 2.4 stars for the previous 5 levels, the next level will be at Normal difficulty.
            1.1.1.3. For students who have achieved an average of below 2 stars for the previous 5 levels, the next level will be at Easy difficulty.
    1.2. The system must be able to generate a summary report for the teachers.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The system must support at least 100 unique student accounts
- The response time for menu navigation must be below 1 second
- The response time for processing student's answer selection must be below 0.1 second
- The time taken for the game to update the overall leaderboard must be below 10 seconds

5.2 Safety Requirements

No safety requirements have been identified.

5.3 Security Requirements

- Student-sensitive data should be stored using and encrypted with an algorithm of equal strength to AES
- All students must have an account in order to access the game.

5.4 Software Quality Attributes

- Flexibility
    - Provisions shall be made for the future usage of multiple languages.
    - No piece of text that might be displayed to a student shall reside in program source code. Every piece of text that a student might see must be modifiable without changing source code. That is, no student-visible text will be "hard-coded."
- Maintainability
    - The system shall not be shut down for maintenance more than once in a 24-hour period.
    - A development programmer who has at least one year of experience supporting this software application shall be able to add a new product feature, including source code modifications and testing, with no more than one week of labor.
    - A new consumer type code must be able to be added to the game within 12 business hours.
    - Function calls shall not be nested more than two levels deep.
- Reusability
    - The game must be able to save student's credentials.
    - The game must be able to accommodate the teaching of another subject/course.

- Reliability
    - The game must be able to be fully displayed within 3 seconds upon refresh.
    - The game must be able to login successfully upon submission of student name and password wit
    - The game will be regularly updated and serviced to ensure its smooth running.

5.5 Business Rules

- Students must be a member of Nanyang Technological University and taking a course related to Software Engineering to access the functionalities of LearnEZ.
- Teacher must be a member of Nanyang Technological University and teaching or facilitating a course related to Software Engineering to access the functionalities of LearnEZ.may imply certain functional requirements to enforce the rules.

6. Other Requirements

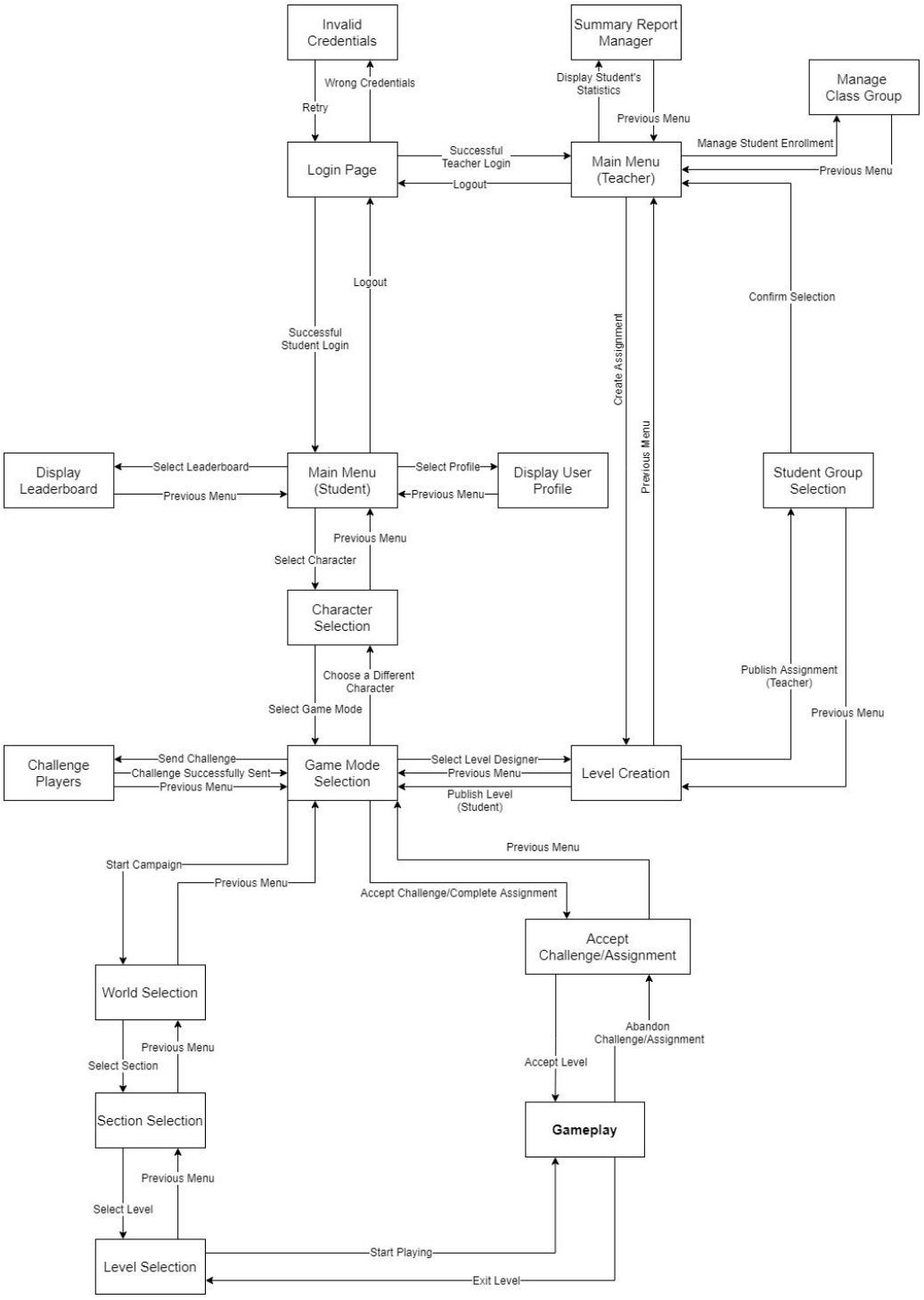
Appendix A: Glossary

Data Dictionary

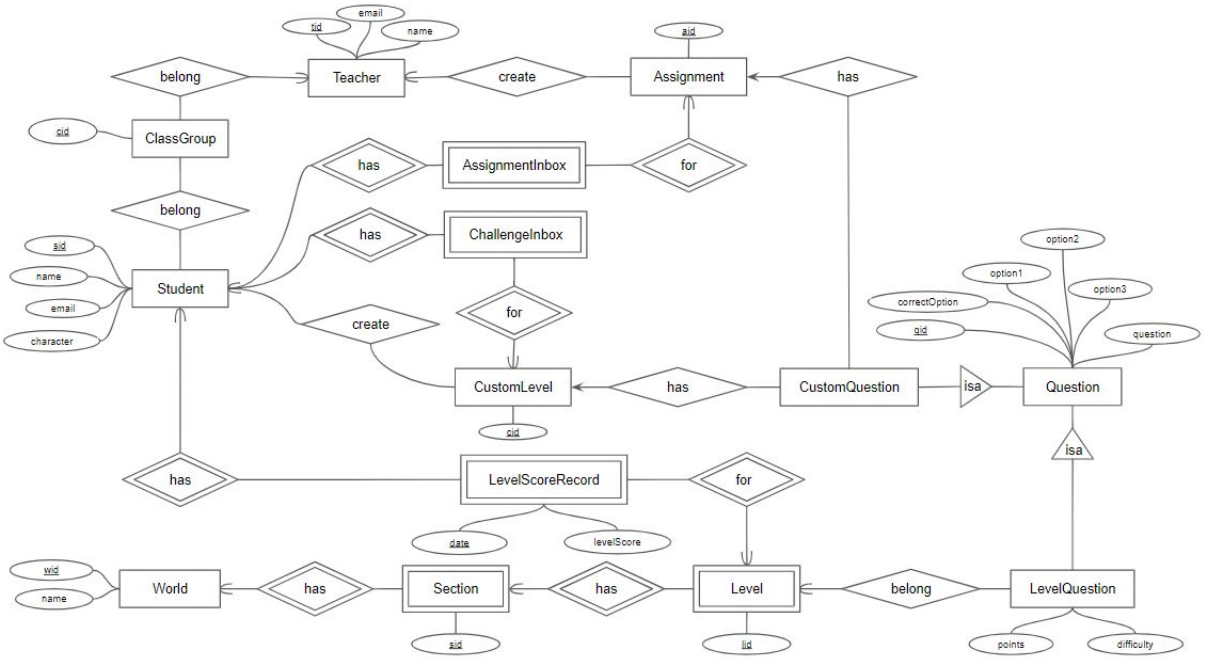
Students	Students refer to users who are playing the game for educational purposes.
Teachers	Teachers refer to users who are able to create classes, create assignments, and access the analytics of students' playing history.
Users	Users refer to both students and teachers.
Classes	Class refers to a particular group of students.
Score	Metric displayed to account for the Student's proficiency.
World	A world is a collection of 5 sections. Each world represents a phase in the life cycle of Software Engineering.
Section	A section is a collection of 5 levels. Each section represents a specific topic of a corresponding phase.
Level	A level is a collection of 5 questions related to the topic of the section.
Question	A MCQ question.
Challenge	A custom made level made by students.
Assignment	A custom made level made by teachers.
Leaderboard	A scoreboard where the top 10 scoring students are listed in descending order.
Character	In game sprite with specific skills
Summary report	Only accessible to the Teacher. Summary report must include Student's name, School and Score.

Appendix B: Analysis Models

Dialog Map



ER Diagram


email
tid
name
Teacher
belong
create
aid
Assignment
has
cid
ClassGroup
has
AssignmentInbox
for
belong
has
ChallengeInbox
sid
name
email
character
Student
create
for
option2
option1
option3
correctOption
qid
question
CustomLevel
has
CustomQuestion
isa
Question
qid
isa
has
LevelScoreRecord
for
date
levelScore
wid
name
World
has
Section
has
Level
belong
LevelQuestion
sid
lid
points
difficulty

Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>