Software Requirements Specification

November 1st, 2024

Academy of Legends

Conner Williams

Kevin Huang

Submitted in partial fulfillment

Of the requirements of

ENSE 400

Table of contents

Academy of Legends

1.0 Introduction

1.1 *Product scope*

Our goal is to develop a web application that helps teachers deliver an interactive gamified learning experience in the classroom. Students love video games, and this project will aim to make learning more engaging, allow students to participate in a team environment, and provide feedback on student progress. The team-based approach will encourage students to strive for success while building accountability among peers, as each member plays a role in their team's achievements.

1.2 *Intended use*

Our web application will be specifically designed for classroom use by teachers of students in grade 8 and below, enabling them to create engaging and interactive learning experiences.

1.3 *General description*

Students will log into their account and create a character. Teachers can give gold and experience points (exp) as rewards as well as penalize students by taking away hearts which is their life point system. This will be done in a gamified classroom system. Students earn these rewards by answering questions correctly, performing well on exams, and through other classroom activities. Students will be organized into guilds, and each student's level, exp, hearts, and gold will be tracked. If a student loses all their hearts, they

“fall” in the game and must complete a “pledge” to rejoin. The game also features a leaderboard, called the Legend’s Board, where students can log in to view their ranks. A reward shop allows students to spend earned gold on classroom rewards or activate random classroom events.

2.0 *Functional Requirements*

The admin (teachers) will be able to create guilds as well as add and remove students to each guild. The application will track health points, power points, exp, and allow teachers to sort and search for specific students. They can also randomly select students for certain events. There will be a reward shop where students can purchase classroom perks, and the Legend’s Board will display rankings. Each student will have their own account for login. If time allows, a boss battle feature will be implemented, where a quiz will involve a boss with a set number of hearts that decreases with each correct answer from students.

3.0 External interface Requirements

3.1 *User interface requirements*

The user interface of the web application will be designed with a clean layout that enables easy navigation between each section, such as guild management, reward shop, leaderboard, and character selection. There will be signifiers for gamification elements, which include health points, levels, experience points, and gold.

3.2 *Hardware interface requirements*

The application will primarily be utilized in the classroom, so we will focus on optimizing it for use on Chromebooks. It will support standard input devices, including a mouse and keyboards for navigation within the application.

3.3 *Software interface requirements*

The application will require reliable real-time data synchronization to ensure that updates are reflected immediately for all users. Use of standard web technologies (HTML, CSS, JavaScript) will be used to ensure easy maintenance and updates.

3.4 *Communication interface requirements*

The application will utilize Django and MongoDB for effective communication between users and the server.

4.0 *Non-functional requirements*

4.1 *Performance*

The system will handle up to 75 users at a time without issues. The response time will be quick for simple actions such as giving rewards or penalties. The student's data will also update immediately. The data will be validated to prevent errors such as awarding the wrong number of gold or exp.

4.2 *Usability*

The UI will be accessible to students as young as grade 3 and will display clear feedback for actions as well as use large buttons for an easier to use experience. The system will be modular which will allow for individual updates without affecting other components.