Muscle Mind

Software Design Document

Version <1.1>

10/14/2024

Document Control

Approval

The Guidance Team and the customer shall approve this document.

Document Change Control

|  |  |
| --- | --- |
| Initial Release: | 1.0 |
| Current Release: | 1.1 |
| Indicator of Last Page in Document: | 11 |
| Date of Last Review: | 10/6/2024 |
| Date of Next Review: | 10/14/2024 |
| Target Date for Next Update: | TBD |

Distribution List

This following list of people shall receive a copy of this document every time a new version of this document becomes available:

Guidance Team Members: Veronica Rivas

Customer: CS 4374 Software Construction and Bhanukiran Gurijala

Software Team Members: Ruben Martinez, Alfonso Hernandez, David Duru, Logan Armendariz, Francisco Jimenez, Caleb Lopez, Jose Legarreta

Change Summary

The following table details changes made between versions of this document

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Modifier | Description |
| 1.0 | 10/6/24 | RM and AH | Initial establishment of design |
| 1.1 | 10/14/24 | RM and AH | Aggregate classes and collaboration graphs |
|  |  |  |  |

Table of Contents

[Document Control ii](#_Toc322626963)

[Approval ii](#_Toc322626964)

[Document Change Control ii](#_Toc322626965)

[Distribution List ii](#_Toc322626966)

[Change Summary ii](#_Toc322626967)

[Introduction 1](#_Toc322626968)

[1.1 Purpose and Intended Audience 1](#_Toc322626969)

[1.2 Scope of Product 1](#_Toc322626970)

[1.3 References 1](#_Toc322626971)

[1.4 Definitions, Acronyms, and Abbreviations 1](#_Toc322626972)

[1.1.1 Definitions 1](#_Toc322626973)

[1.4.1 Acronyms and Abbreviations 1](#_Toc322626974)

[1.5 Overview 1](#_Toc322626975)

[2 Decomposition Description 2](#_Toc322626976)

[2.1 System Collaboration Diagram 2](#_Toc322626977)

[2.2 Subsystem Descriptions 2](#_Toc322626978)

[3 Detailed Description of Subsystem <name> 3](#_Toc322626979)

[3.1 Subsystem Description (subsystem cards) 3](#_Toc322626980)

[3.2 Class Description (CRC cards) <class 1 name> 3](#_Toc322626981)

[3.2.1 Contract <contract 1 name> 3](#_Toc322626982)

[3.2.2 Contract <contract 2 name> 3](#_Toc322626983)

[3.3 Class Description <class 2 name> 3](#_Toc322626984)

[3.3.1 Contract <contract 1 name> 3](#_Toc322626985)

[Appendix 4](#_Toc322626986)

# Introduction

## Purpose and Intended Audience

The purpose of the project is to design a game that can help people exercise while having fun. The intention is to have a game that can not only be a fun party trivia or quiz game but also be able to exercise. The primary intended audience would use the SDD to see how the developers will develop a game that can integrate the quiz questions into the game and how incorrect responses would result in an exercise prompt.

## Scope of Product

The Scope of Muscle Minds will include Quiz management, Exercise challenge system, User management and Performance Reporting system. The Quiz management subsystem will mainly cover the creation, storage and presentation of the quiz and its questions while the Exercise Challenge subsystem will cover the integration of the physical exercise prompts as soon as the game detects that the answer is incorrect. The User management subsystem will manage the user accounts with its log in functionality and access to the custom quizzes made by the user while the Performance Reporting subsystem generates the grade reports reporting on feedback with correct to incorrect responses and the completed exercise challenges. What this game will do is motivate users to make their quiz for either fun party questions or a way to study while having physical engagement with its various exercise prompts that it will select at random but what it will not do at this time is provide real time exercise form feedback or completion feedback and will rely on the user to self-report. Our main goal is to create a quiz game that is unique and physically engages with the user to encourage exercise in a fun manner.

## References

[Project Plan Sheet](https://minersutep-my.sharepoint.com/:w:/r/personal/rjmartinez12_miners_utep_edu/_layouts/15/Doc.aspx?sourcedoc=%7BA507532D-06CA-47CE-B503-89A5DB94CCD5%7D&file=scm-plan-template.doc&action=default&mobileredirect=true)

[Project Report](https://minersutep-my.sharepoint.com/:w:/r/personal/rjmartinez12_miners_utep_edu/_layouts/15/Doc.aspx?sourcedoc=%7B189B9891-0520-459D-A31E-4D2C6A17539B%7D&file=Team%203%20Software%20Construction%20Part%201%20Report.docx&action=default&mobileredirect=true)

[GitHub Repository](https://github.com/riddle-me-ruben/muscle-mind)

## Definitions, Acronyms, and Abbreviations

### Definitions

* **Game Session** - An instance of a quiz game that tracks the player's progress, including questions answered, scores, and penalties.
* **Exercise Challenge** - A feature that assigns physical activities (e.g., push-ups, squats) to users as penalties when they answer quiz questions incorrectly, promoting physical engagement.
* **Performance Report** - A summary generated after a quiz game session that shows the user’s scores, completed exercises, and overall performance metrics.

### Acronyms and Abbreviations

* **MM** – Muscle Mind
* **RM** – Ruben Martinez
* **PR** – Pull Request
* **QM** – Quiz Manager

## Overview

Introduction: Purpose and scope of the SDD and software product overview.

Definitions, Acronyms, and Abbreviations: Lists terminology used in the document.

System Overview: High-level description of the system and its goals.

Decomposition Description: Breaks down the system into components and their functions.

System Collaboration Diagram: Diagram showing interactions between subsystems.

Subsystem Descriptions: Details each subsystem, its purpose, and supported functions.

Detailed Description of Subsystems: In-depth design of subsystems and their classes.

Class Description (CRC Cards): Responsibilities and collaborations of each class.

Contract Descriptions: Protocols and methods for each class.

Appendix: Additional diagrams and supporting materials

1. Decomposition Description

In MuscleMind, component descriptions allow designers and maintainers to identify major design entities, such as the quiz system, exercise penalty logic, and user performance reports. These descriptions clarify which entities are responsible for specific functions, like administering quizzes or assigning exercise challenges. This ensures that system requirements are easily traced back to the relevant design entities, aiding in maintenance and updates.

## System Collaboration Diagram

For **MuscleMind**, the system is structured around several key subsystems that collaborate to create an interactive and educational experience. Each subsystem focuses on a specific area of functionality, making the system modular and easier to manage. Below is an explanation of how the subsystems work together, including their main roles and collaborations:

### Major Subsystems and Their Collaborations

1. **User Management Subsystem**
   1. **Classes**: UserAuthentication, UserProfile
   2. **Description**: Manages user-related features such as registration, login, and profile management. It handles user authentication, ensures secure access, and stores user information like scores and quiz history.
   3. **Collaborations**:
      1. Works with the Database Management Subsystem to store and retrieve user data.
      2. Interacts with the Quiz Management Subsystem to ensure authenticated users can access their quizzes.
2. **Quiz Management Subsystem**
   1. **Classes**: QuizManager, QuestionManager
   2. **Description**: Responsible for handling all quiz-related activities, including creating quizzes, storing questions, and managing the flow of quizzes during gameplay.
   3. **Collaborations**:
      1. Collaborates with the User Management Subsystem to link quizzes to specific users and retrieve their custom quizzes.
      2. Interacts with the Exercise Challenge Subsystem when a player answers incorrectly to trigger exercise penalties.
      3. Uses the Database Management Subsystem to save and retrieve quiz information.
3. **Exercise Challenge Subsystem**
   1. **Class**: ExercisePenaltyManager
   2. **Description**: Manages the exercise penalties that are triggered when players answer quiz questions incorrectly. This subsystem ensures that physical challenges are randomly assigned and tracked for each user.
   3. **Collaborations**:
      1. Works with the Quiz Management Subsystem to know when to assign an exercise challenge based on incorrect answers.
      2. Collaborates with the Performance Reporting Subsystem to track and log exercises performed by the user during the session.
4. **Performance Reporting Subsystem**
   1. **Class**: PerformanceReport
   2. **Description**: Generates reports based on player performance during quizzes, including scores and completed exercise penalties. This subsystem tracks user progress and provides feedback.
   3. **Collaborations**:
      1. Interacts with the Game Session Subsystem to gather data on player performance during a game session.
      2. Collaborates with the Database Management Subsystem to store and retrieve past performance reports and user history.
5. **Game Session Subsystem**
   1. **Class**: GameSession
   2. **Description**: Manages the game flow, tracking the sequence of questions, timing, scoring, and exercise challenges during a session. It ensures the gameplay is dynamic and engaging.
   3. **Collaborations**:
      1. Works closely with the Quiz Management Subsystem to handle quiz questions and timing.
      2. Interacts with the Exercise Challenge Subsystem to trigger physical challenges based on player responses.
      3. Collaborates with the Performance Reporting Subsystem to log performance data for reports.
6. **Database Management Subsystem**
   1. **Class**: DatabaseManager
   2. **Description**: The central hub for all data operations, including user information, quizzes, questions, performance reports, and exercise logs. This subsystem ensures that all data is securely stored and easily retrievable.
   3. **Collaborations**:
      1. Supports all other subsystems (User Management, Quiz Management, Exercise Challenge, and Performance Reporting) by providing access to stored data.
      2. Ensures that the data remains consistent and is accessible by managing CRUD (Create, Read, Update, Delete) operations.

## Subsystem Descriptions

The Quiz Management Subsystem handles the creation, storage, and presentation of quizzes and questions, supporting contracts for quiz creation, storage, and presentation to users. The Exercise Subsystem integrates physical exercise prompts when incorrect answers are detected, with contracts for incorrect answer detection, exercise challenge assignment, and prompt presentation. The User Management Subsystem manages user accounts, including login functionality and access to custom quizzes, supporting contracts for user authentication, profile management, and custom quiz access. Finally, the Performance Reporting Subsystem generates grade reports based on quiz performance, with contracts for performance tracking, report generation, and data storage and retrieval.

1. **User Management Subsystem**
   1. **Purpose**: Manages all user-related features such as registration, login, and profile management. It ensures secure access and stores user-related information like scores and quiz history.
   2. **Contracts**:
      1. UserAuthentication: Manages login, registration, and logout processes.
      2. UserProfile: Allows updating and retrieval of user profile information.
   3. **Location in Document**: This subsystem and its components are discussed in detail in section 3.1 (User Management Details).
2. **Quiz Management Subsystem**
   1. **Purpose**: Responsible for handling all quiz-related activities, including creating, storing, and retrieving quizzes and questions during gameplay.
   2. **Contracts**:
      1. QuizManager: Manages the creation and retrieval of quizzes.
      2. QuestionManager: Handles question addition, storage, and retrieval.
   3. **Location in Document**: This subsystem and its components are further detailed in section 3.2 (Quiz Management Details).
3. **Exercise Challenge Subsystem**
   1. **Purpose**: Manages exercise penalties triggered when players answer quiz questions incorrectly, ensuring the physical challenges are assigned and tracked.
   2. **Contracts**:
      1. ExercisePenalty: Randomly assigns and tracks physical challenges.
   3. **Location in Document**: Details of this subsystem can be found in section 3.3 (Exercise Challenge Subsystem Details).
4. **Performance Reporting Subsystem**
   1. **Purpose**: Generates reports based on player performance during quizzes, including scores and completed exercises, to track progress and provide feedback.
   2. **Contracts**:
      1. PerformanceReport: Generates and retrieves performance reports for users.
   3. **Location in Document**: The subsystem and its details are elaborated in section 3.4 (Performance Reporting Details).
5. **Game Session Subsystem**
   1. **Purpose**: Manages the flow of the game by tracking the sequence of questions, timing, scoring, and exercise penalties during each session.
   2. **Contracts**:
      1. GameSession: Manages the initiation, progress, and completion of game sessions.
   3. **Location in Document**: This subsystem is discussed in section 3.5 (Game Session Subsystem Details).
6. **Database Management Subsystem**
   1. **Purpose**: Acts as the central hub for data operations, including managing user information, quizzes, questions, performance reports, and exercise logs. It ensures secure storage and retrieval of all data entities.
   2. **Contracts**:
      1. DatabaseManager: Provides access to CRUD (Create, Read, Update, Delete) operations for various data entities.
   3. **Location in Document**: The details of this subsystem are included in section 3.6 (Database Management Subsystem Details).

### Summary

Each subsystem's purpose is to support a specific aspect of the Muscle-Mind game, ensuring a modular and maintainable structure. The contracts listed indicate the key functionalities each subsystem supports, and the detailed discussions for these components are provided in subsequent sections of the document, allowing for an organized and consistent reference throughout the project documentation.

1. Detailed Description of Subsystems

## Subsystem Descriptions (subsystem cards)

|  |
| --- |
| **Subsystem Name:** User Management |
| **Classes:** UserAuthentication, UserProfile |
| **Collaboration Graph:** |
| **Description:** Manages all user-related features such as registration, login, and profile management. It ensures secure access and stores user-related information like scores and quiz history. |
| **Contracts:**   * UserAuthentication: Manages login, registration, and logout processes for users. * UserProfile: Manages updating and retrieval of user profile details like quiz history and performance. |

|  |  |
| --- | --- |
| **Class Name:** UserAuthentication | |
| **Description:** Handles the processes involved in authenticating users, including logging in, registering new accounts, and logging out. | |
| **Super classes:** UserAccountManager | |
| **Contracts:**  **Contract 1: User Authentication**  **Description: Handles user login and logout, validating credentials, and maintaining active sessions.**  - Allow users to change their passwords and update profile settings.  - Securely validate and save updated account information.  **Private Responsibilities:**  - Encrypt user passwords before storage.  - Handle session token generation and expiration.  … | **Collaborations:**  Collaborating Class: UserProfile | Contract #:2  Collaborating Class: UserAuthentication | Contract #:1 |
| **Comments:**   * This class requires secure encryption standards for password management. * Known attributes include userToken and passwordHash. | |

### Contract - User Authentication

This contract ensures secure user authentication via email and password. It also manages session tokens to keep track of logged-in users while having met the pre-condition of the user providing their current password to authorize sensitive updates while the post-condition would be that the new updated information would be stored in the database. Its responsibility includes facilitating the secure updates of user credentials and profile information.

|  |
| --- |
| **Subsystem Name:** Quiz Management |
| **Classes:** QuizManager, QuestionManager |
| **Collaboration Graph:** Inserting image... |
| **Description:** Responsible for handling all quiz-related activities, including creating, storing, and retrieving quizzes and questions during gameplay. |
| **Contracts:**   * **Quiz Management:** Provides functionalities for managing quizzes. |

|  |  |
| --- | --- |
| **+Class Name:** QuizManagement | |
| **Description:** Manages the creation, updating, and retrieval of quizzes. | |
| **Super classes: None** | |
| **Contracts:**  **Contract 2: QuizManager**  **Description:** Provides functionalities for managing quizzes such as grading and report summaries.  **Private Responsibilities:**  -Validate quiz details before storage.  -Link quizzes to users. | **Collaborations:**  Collaborating Class: QuizManager | Contract #:2  Collaborating Class: QuestionManager | Contract #:4 |
| **Comments:**   * Shows how the user can create quizzes or show premade quizzes. * Manages grades for users and performs computations. | |

### Contract – Quiz Management

The Quiz Management contract is responsible for managing the creation, storage, and retrieval of quizzes within the system. It defines the methods and protocols needed to allow users to create new quizzes, update existing ones, and retrieve them for gameplay. The contract ensures that each quiz is associated with an authenticated user, maintaining security and access control. It also validates quiz data for consistency, ensuring that all questions and options are correctly formatted and stored. By providing these functionalities, the contract maintains the integrity and availability of quizzes, allowing the system to offer a dynamic and engaging user experience. Additionally, the contract supports interactions with the DatabaseManager to handle data operations and with UserAuthentication to verify user permissions, ensuring that only authorized users can create, modify, or access their quizzes.

|  |
| --- |
| **Subsystem Name:** Exercise Challenge Subsystem |
| **Classes: ExcercisePenaltyManager** |
| **Collaboration Graph:** |
| **Description: Assigns and tracks exercise penalties during quiz gameplay.** |
| **Contracts:**   * **Penalty Assignment- Manages the assignment of exercise challenges** |

|  |  |
| --- | --- |
| **+Class Name:** **ExcercisePenaltyManager** | |
| **Description:** Assigns and tracks exercise penalties during quiz gameplay. | |
| **Super classes: None** | |
| **Contracts:**  **Contract 5:** Penalty Assignment  **Description:** Manages the assignment of exercise challenges.   * **Pre-conditions**: Quiz question must be answered incorrectly. * **Post-conditions**: Exercise penalty is logged.   **Private Responsibilities:**  - Encrypt user passwords before storage.  - Handle session token generation and expiration.  … | **Collaborations:**  Collaborating Class: ExcercisePenaltyManager | Contract #:5 |
| **Comments:**   * Distributes penalties to users in the form of exercises. * Exercises can be random (pushups, situps). | |

### Contract – Exercise Challenge

The Exercise Challenge contract manages the assignment and tracking of physical exercise penalties when players answer quiz questions incorrectly. This contract outlines the methods and protocols required to randomly generate and assign appropriate exercise challenges, such as push-ups or squats, based on quiz results. It ensures that each challenge is logged and associated with the relevant user and game session, maintaining accurate tracking of physical activities performed during gameplay. The contract interacts with the QuizManager to determine when a penalty should be assigned and collaborates with the PerformanceReport system to log the penalties as part of the user’s performance data. By doing so, it promotes an engaging and interactive experience while integrating physical activities into the learning environment.

|  |
| --- |
| **Subsystem Name:** Performance Reporting Subsystem |
| **Classes: PerformanceReport** |
| **Collaboration Graph:** |
| **Description: Manages the generation and retrieval of performance reports.** |
| **Contracts:**   * Reporting - Compiles and retrieves user performance data |

|  |  |
| --- | --- |
| **+Class Name:** PerformanceReport | |
| **Description:** Manages the generation and retrieval of performance reports. | |
| **Super classes:** None | |
| **Contracts:**  **Contract 7: Performance**  **Description: Compiles and retrieves user performance data**  - A way to generate performance and stats  - A chance to recieve feedback  **Private Responsibilities:**   * Show scores to UI. | **Collaborations:**  Collaborating Class: ExcercisePenaltyManager | Contract #:7 |
| **Comments:**   * Gives performance report to user in friendly format. * Shows scores as well as progress | |

### Contract – Performance Report

The Performance Report contract is responsible for generating, storing, and retrieving user performance reports based on quiz results and exercise challenges completed during gameplay. This contract defines the methods and protocols needed to compile quiz scores, track completed exercise penalties, and produce a comprehensive report that summarizes the user’s performance. It ensures that reports are accurately generated by gathering data from the GameSession and Exercise Challenge subsystems. The contract also facilitates retrieving historical reports for users, allowing them to review and track their progress over time. To maintain data integrity, it interacts with the DatabaseManager for storing and accessing report data securely. By implementing these functionalities, the contract supports continuous feedback for users, helping them assess their performance and motivating them to improve in future sessions.

|  |
| --- |
| **Subsystem Name:** Game Session Subsystem |
| **Classes: GameSession** |
| **Collaboration Graph:** |
| **Description: Controls gameplay, managing quiz sequences and scoring.** |
| **Contracts:**   * **UserAuthentication**: Manages login, registration, and logout processes for users. * **UserProfile**: Manages updating and retrieval of user profile details like quiz history and performance. |

|  |  |
| --- | --- |
| **+Class Name:** GameSession | |
| **Description:** Handles the processes involved in authenticating users, including logging in, registering new accounts, and logging out. | |
| **Super classes: UserAccountManager** | |
| **Contracts:**  **Contract 1: Game Management**  **Description: Handles user login and logout, validating credentials, and maintaining active sessions.**  - Allow users to change their passwords and update profile settings.  - Securely validate and save updated account information.  **Private Responsibilities:**  - Encrypt user passwords before storage.  - Handle session token generation and expiration.  … | **Collaborations:**  < Collaborating class with contract number >> |
| **Comments:**   * This class requires secure encryption standards for password management. | |

|  |
| --- |
| **Subsystem Name:** Database Management Subsystem |
| **Classes: QuizManager, QuestionManager** |
| **Collaboration Graph:** Inserting image... |
| **``** |
| **Contracts:**   * **UserAuthentication**: Manages login, registration, and logout processes for users. * **UserProfile**: Manages updating and retrieval of user profile details like quiz history and performance. |

|  |  |
| --- | --- |
| **+Class Name:** UserAuthentication | |
| **Description:** Handles the processes involved in authenticating users, including logging in, registering new accounts, and logging out. | |
| **Super classes: UserAccountManager** | |
| **Contracts:**  **Contract 1: User Authentication**  **Description: Handles user login and logout, validating credentials, and maintaining active sessions.**  - Allow users to change their passwords and update profile settings.  - Securely validate and save updated account information.  **Private Responsibilities:**  - Encrypt user passwords before storage.  - Handle session token generation and expiration.  … | **Collaborations:**  < Collaborating class with contract number >> |
| **Comments:**   * This class requires secure encryption standards for password management. * Known attributes include userToken and passwordHash. | |

### Contract - User Authentication

This contract ensures secure user authentication via email and password. It also manages session tokens to keep track of logged-in users while having met the pre-condition of the user providing their current password to authorize sensitive updates while the post-condition would be that the new updated information would be stored in the database. Its responsibility includes facilitating the secure updates of user credentials and profile information.

Appendix

<< collaboration diagrams (high-level and subsystem collaboration graphs) >>

