

---

# **Software Design Specification**

**for**

# **Mobile Security Awareness Training Software**

**Version 1.0**

**Prepared by  
Nova Thomas  
Kelsea Canaday  
Nicholas Ciarlone  
Trent Halama  
Alejandro Ardon**

**Florida Polytechnic University**

**04/24/2024**

# Table of Contents

<b>Table of Contents</b>	<b>ii</b>
<b>1. Introduction</b>	<b>1</b>
1.1 Purpose	1
1.2 System Overview	1
<b>2. Design Considerations</b>	<b>1</b>
2.1 Assumptions	1
2.2 General Constraints	1
<b>3. System Architecture</b>	<b>2</b>
<b>4. Detailed System Design</b>	<b>3</b>
4.1 Class Diagram	3
4.2 Activity Diagram	4
4.3 Sequence Diagram	5
4.4 State Chart Diagram	6
4.5 Algorithms for Components/Methods	8
4.6 Database Design	8
<b>5. User Interface Design</b>	<b>9</b>
<b>Appendix A: Glossary</b>	<b>10</b>

# 1. Introduction

## 1.1 Purpose

*The purpose of this project is to design and develop training software to evaluate a user's awareness of mobile security and cyber attacks on mobile devices in an entertaining game format. The product will be implemented on a simple desktop-based game application that asks users questions about different types of cyber attacks on mobile devices and basic protection techniques, provides feedback on user answers, and displays correct answers. The training game application shall allow users to create their accounts, and it will only accept strong passwords of at least ten or more characters with a unique character, number, and letter mixed within the password. To start the game, a user must provide valid credentials such as username and password. The application shall also store previous game/quiz scores in the user profile.*

## 1.2 System Overview

*The system overview contains desktop-based training software designed to enhance users' awareness of mobile security and cyber attacks in an engaging game format. Operating on Windows, Linux, or MacOS environments, the software prompts users to create accounts with secure passwords. Through gameplay, users- categorized as Students or Educators- interact with text-based scenarios to identify security risks and cyber attacks, receiving immediate feedback after responding. The system records user credentials and game scores in a database, allowing Educators to monitor student progress. Game controls enable pausing, ending, or restarting sessions. With performance requirements demanding a minimum dual-core x86 processor, 2GB RAM, and adequate storage, the software prioritizes usability, reliability, maintainability, and security, reinforcing its educational and protective features.*

# 2. Design Considerations

## 2.1 Assumptions

*The software assumes the availability of compatible hardware, such as desktop computers that use a mouse and keyboard, and internet connectivity for communication purposes. Additionally, it relies on an appropriate operating system environment, including the latest versions of Windows, Linux, or MacOS, to ensure functionality. End-user characteristics are assumed to vary, accommodating experienced and inexperienced individuals in cybersecurity issues. Furthermore, the software anticipates possible changes in functionality over time, necessitating periodic updates to adapt to evolving cyber threats and user requirements. These assumptions and dependencies underline the software's flexibility and adaptability.*

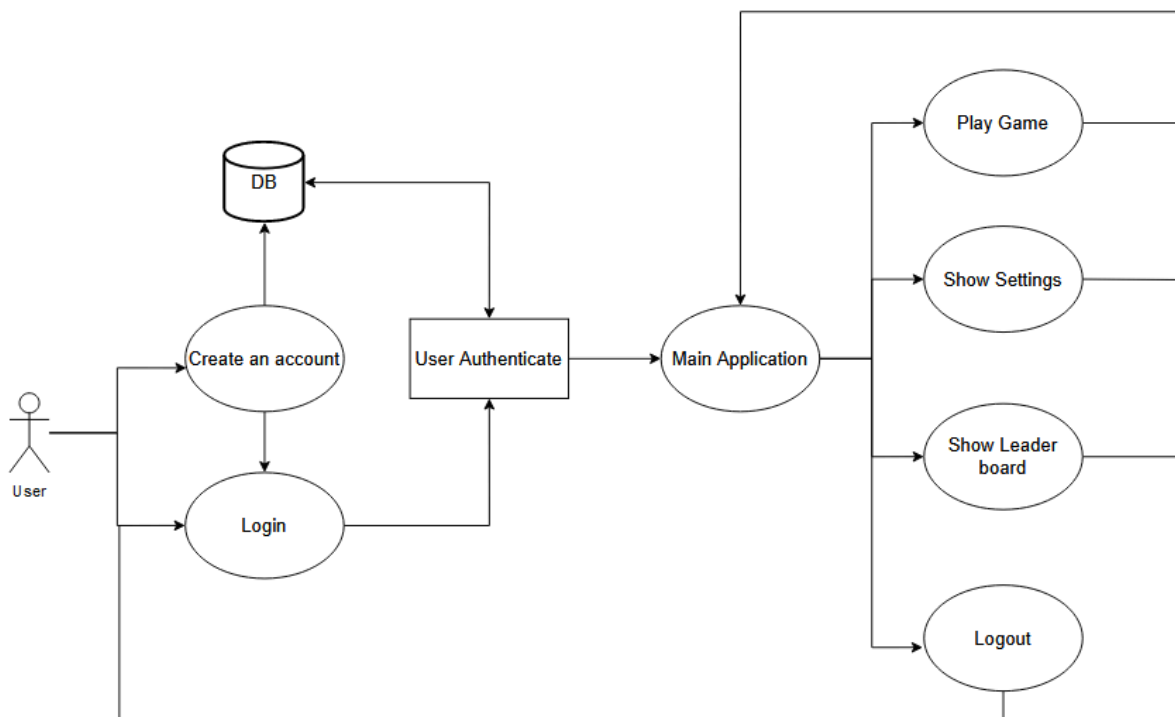
## 2.2 General Constraints

*The hardware and software environment imposes constraints on compatibility and performance, causing a need for specified operating systems and hardware specifications. End-user environments may vary, requiring the software to be user-friendly and adaptable across different*

setups. Availability of resources, including network bandwidth and database access, influences system responsiveness and reliability. Compliance with standards and interoperability requirements ensures seamless integration with existing systems and protocols. Security regulations dictate stringent measures for user authentication and data protection. Memory and capacity limitations affect the software's efficiency. Performance requirements demand optimal utilization of system resources to deliver a smooth user experience. Network communications must be secure to ensure uninterrupted gameplay and data transmission. These constraints collectively shape the design of the system's software, emphasizing compatibility, reliability, and security as key priorities.

### 3. System Architecture

This system architecture diagram is featured below, showcasing the general functionality of the application.

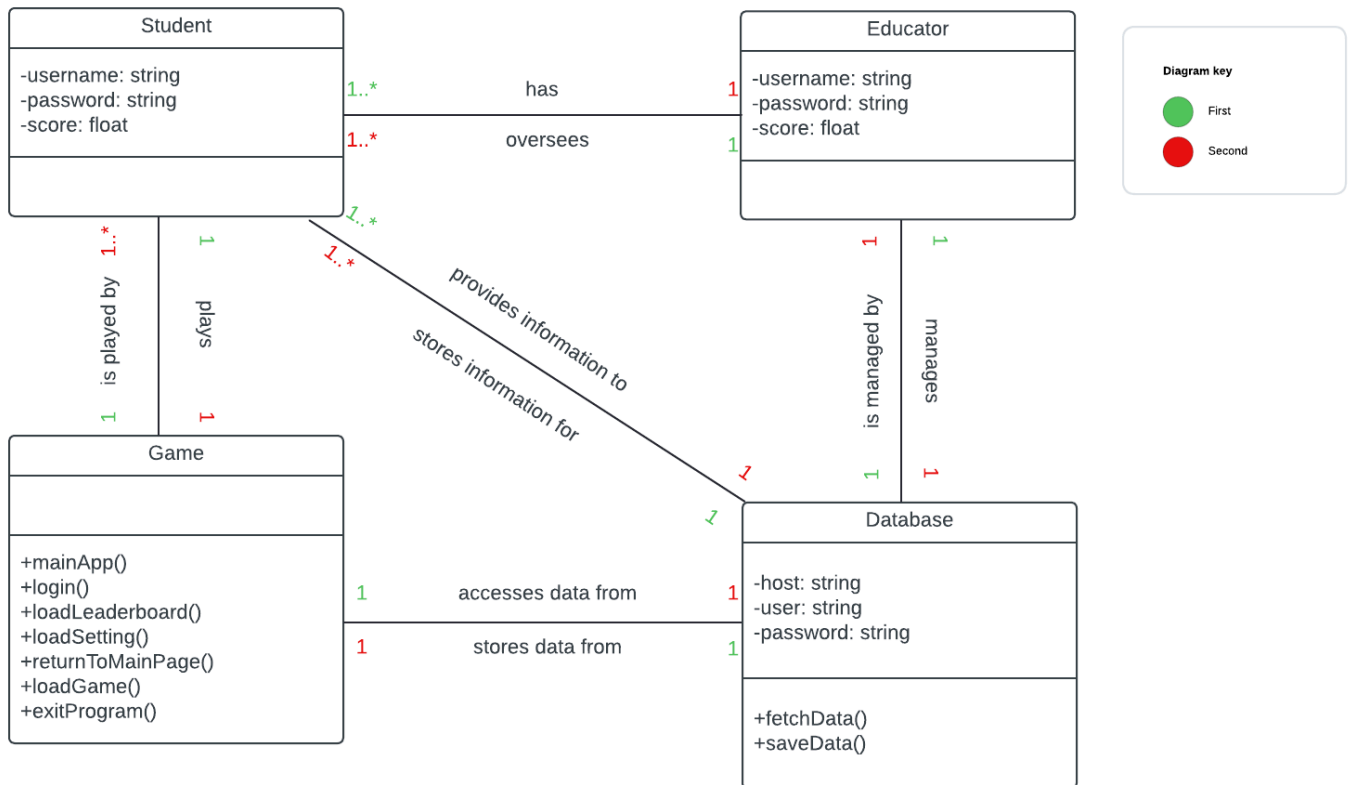


When the user logs in, it will connect to the database to confirm the correct username and password. If it fails, it will prompt the user to try again or create an account. Once the account is verified and connected, it will go to the main application and run the game functionality.

## 4. Detailed System Design

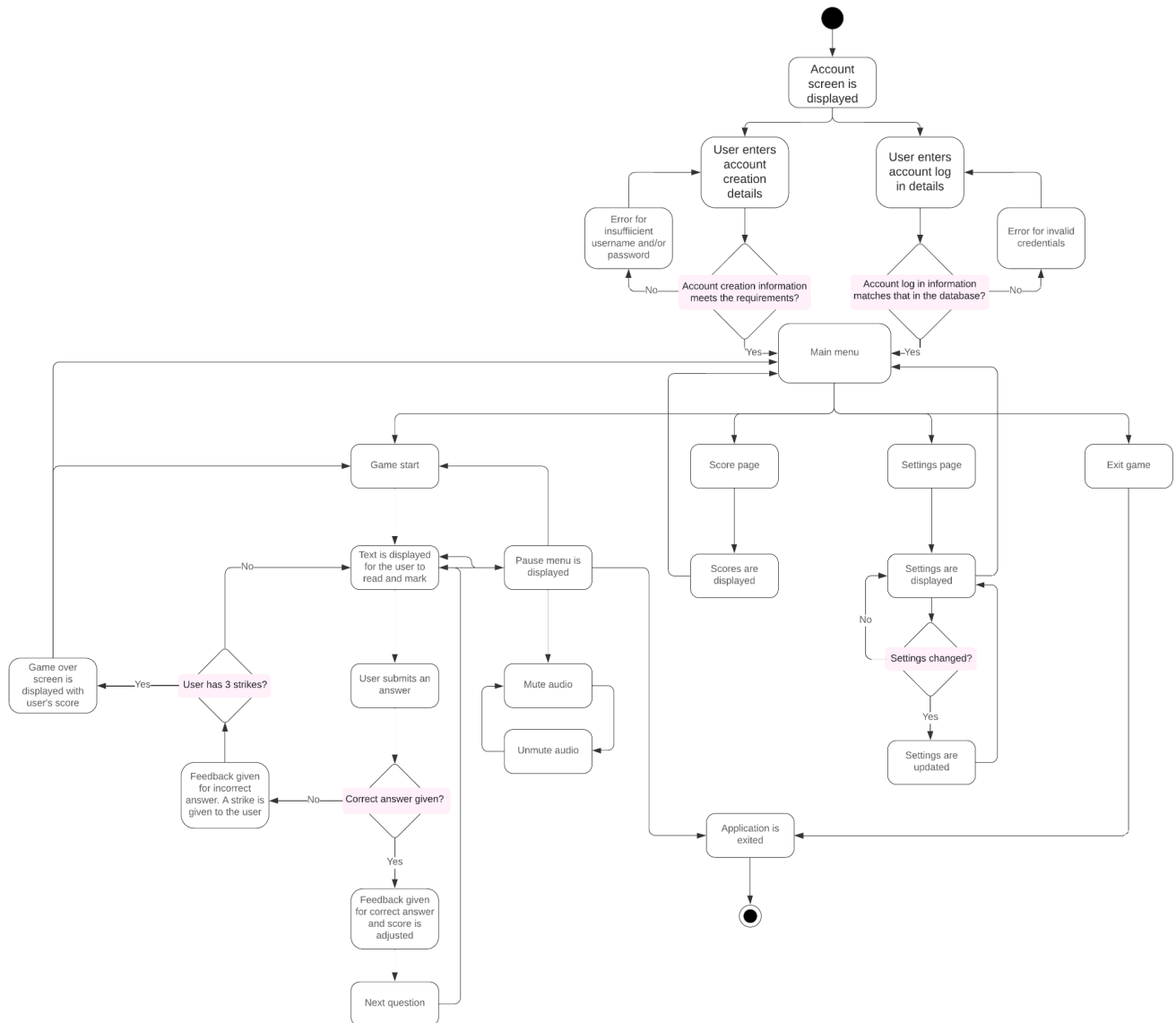
### 4.1 Class Diagram

The class diagram features this primary set of classes: educators, students, games, and the database. The educators and students are similar in that they have a username and password. The database manages the database by hosting the functions to collect and save data. The game class acts as the application manager for the user experience.



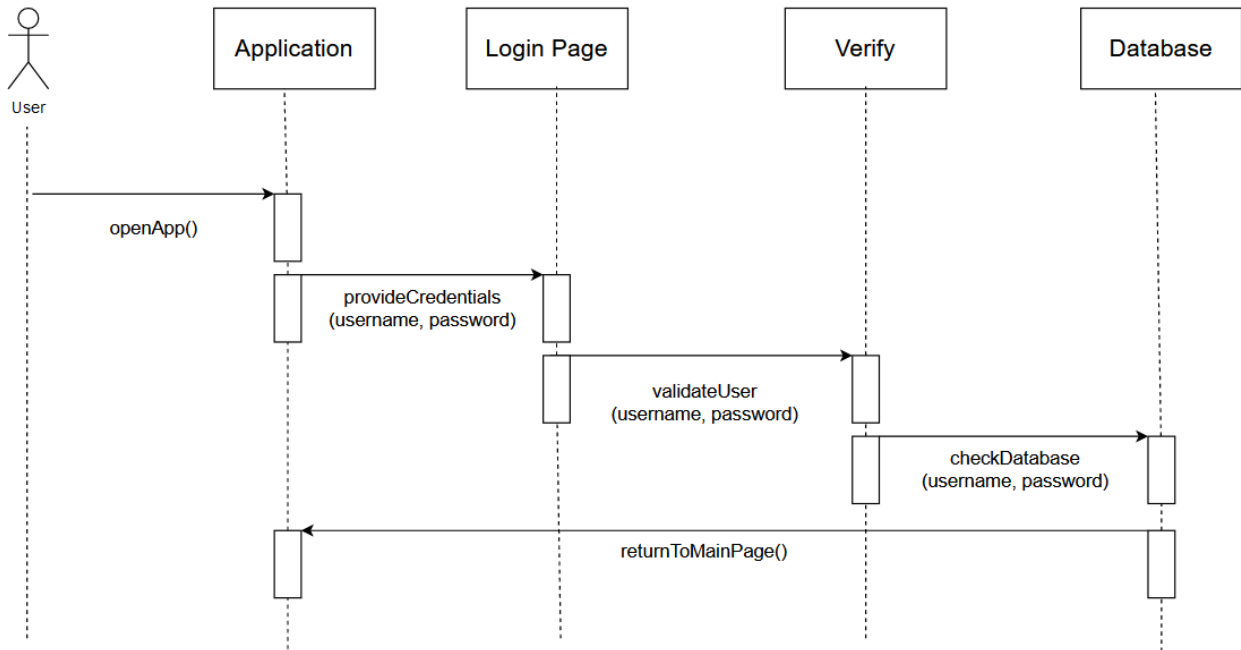
## 4.2 Activity Diagram

The activity diagram details the actions a user may make while using the application and their flow from one to another. The activity diagram depicts the application at an initial state at startup; then it moves to verification of a new or existing account. From that point, it moves to the main game menu and branches into numerous paths traversed based on the user's inputs until the end state is reached (the application closing). These states include the various pages (the settings page, the pause menu, the scores page, the main gameplay loop, and the exit), which additionally have conditions that will affect which path is taken.

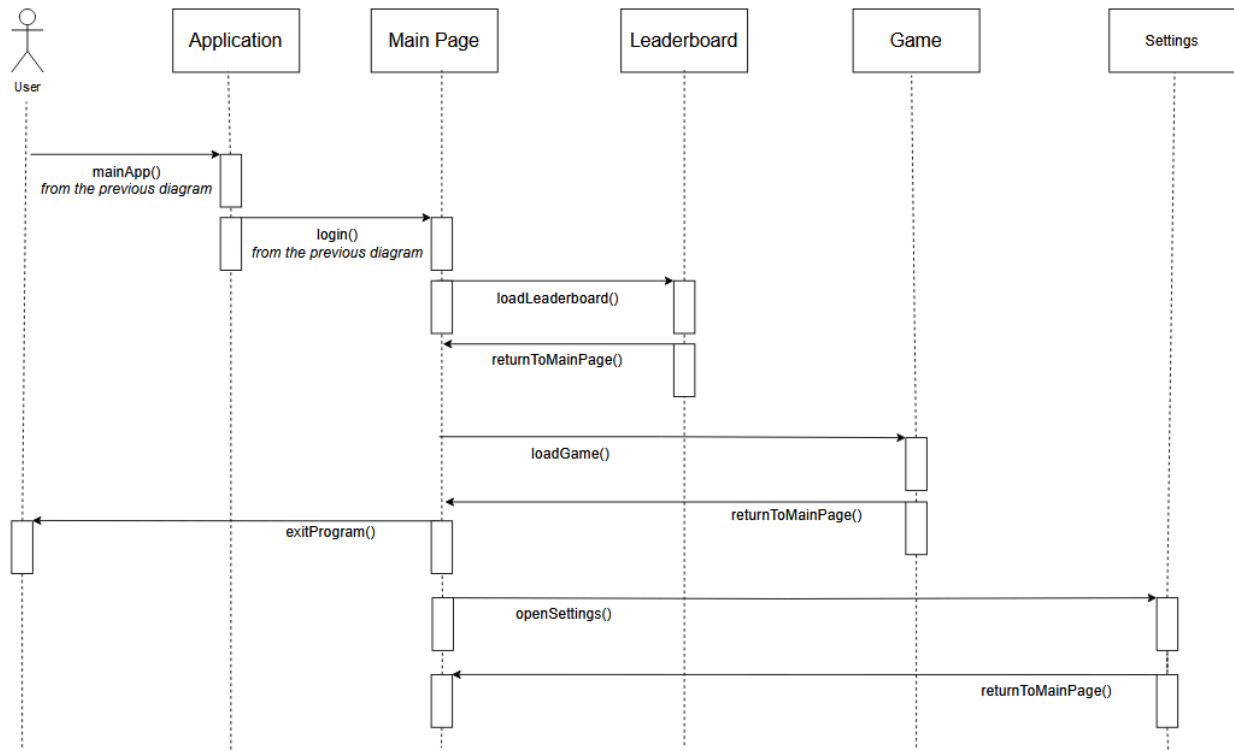


### 4.3 Sequence Diagram

*The sequence diagram simulates that once the user opens the application, it will prompt them to log in with a valid username and password. When the login is verified through the database, it will then open to the main screen where they can play the game. Please see section five for more information about the main screen; both diagrams are below.*



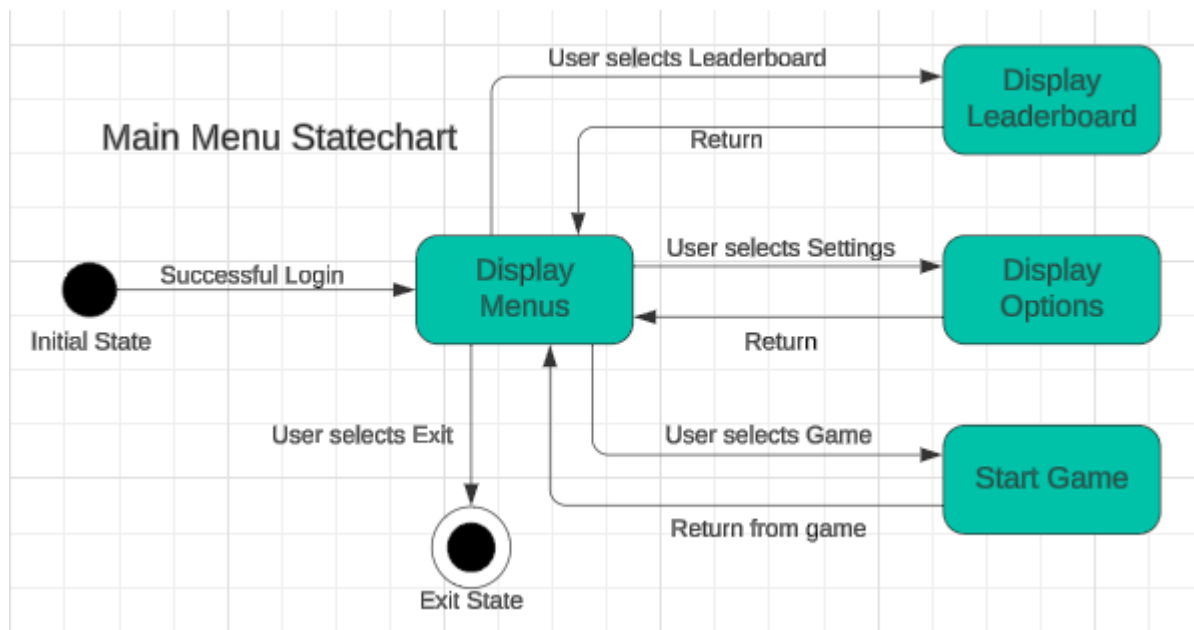
*This diagram represents the login processes. When opening the application, the login process will start. Once the user provides credentials, the database will verify them. When it is verified, it will return to the main application page.*



*This shows more of the application process; the login function refers to the previous diagram. Once login is successful, three options will appear: one showing the leaderboard, one running the game, and the other exiting the program. Another button will feature options/settings.*

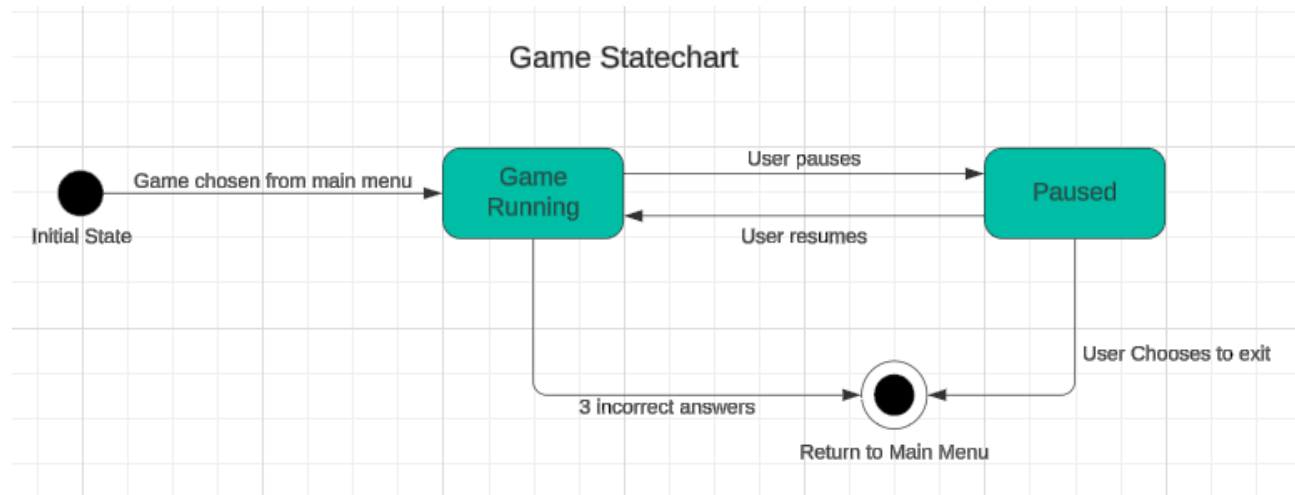
## 4.4 State Chart Diagram

*The state charts below show the different states of the Main Menu, Game, and Audio objects.*

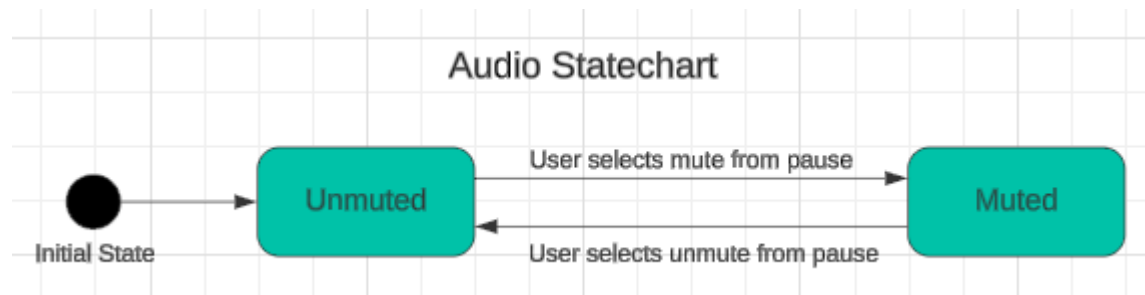




Upon successful login, the main menu will load, allowing the user to pick from four menus: Leaderboard, Options, Start Game, and Exit. When the leaderboard menu is selected, the menu will display the current leaderboard. When the option menu is selected, the menu will display the available options. When the start game option is selected, the game will begin running, as described in the state chart below. When the exit option is selected, the program will exit. The leaderboard, options, and game menus allow the user to return to the main menu.



When the game option is selected from the main menu, the game will begin running. It will continue to run until the game is paused or until the user gets three incorrect answers. If the user gets three incorrect answers, then the game will end, and the user will be returned to the main menu. If the user pauses, they may choose to resume, exit, or change the audio setting. Resuming will return the game to its usual running state. Exiting will return the user to the main menu. Changing the audio is described further in the state chart below.



The audio is unmuted by default. When the user accesses the game's pause menu, they can change the audio to mute. The user can also change the audio back to unmuted if they choose.

## 4.5 Algorithms for Components/Methods

*mainApp(): The main application page.*

*login(): Allow the user to input their credentials. Connect to the database to verify their credentials.*

*loadLeaderboard(): Allow the user to view the current leaderboard. Connect to the database to get the leaderboard information.*

*loadSetting(): Allow the user to view the current settings and make adjustments.*

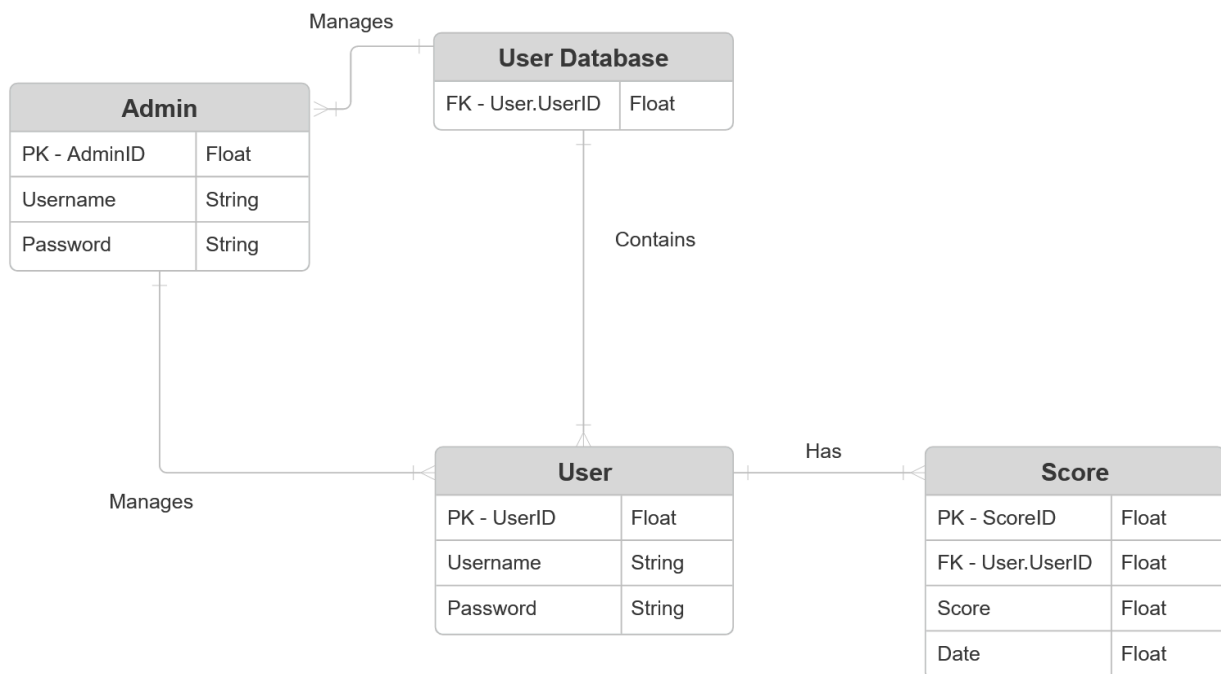
*returnToMainPage(): Allow the user to return to the main page*

*loadGame(): Load the game*

*exitProgram(): Exit the program*

## 4.6 Database Design

*A database will be created to store information about user accounts. An admin account can view all accounts and scores from their previous games. The user accounts can view their previous scores in the game.*



## 5. User Interface Design

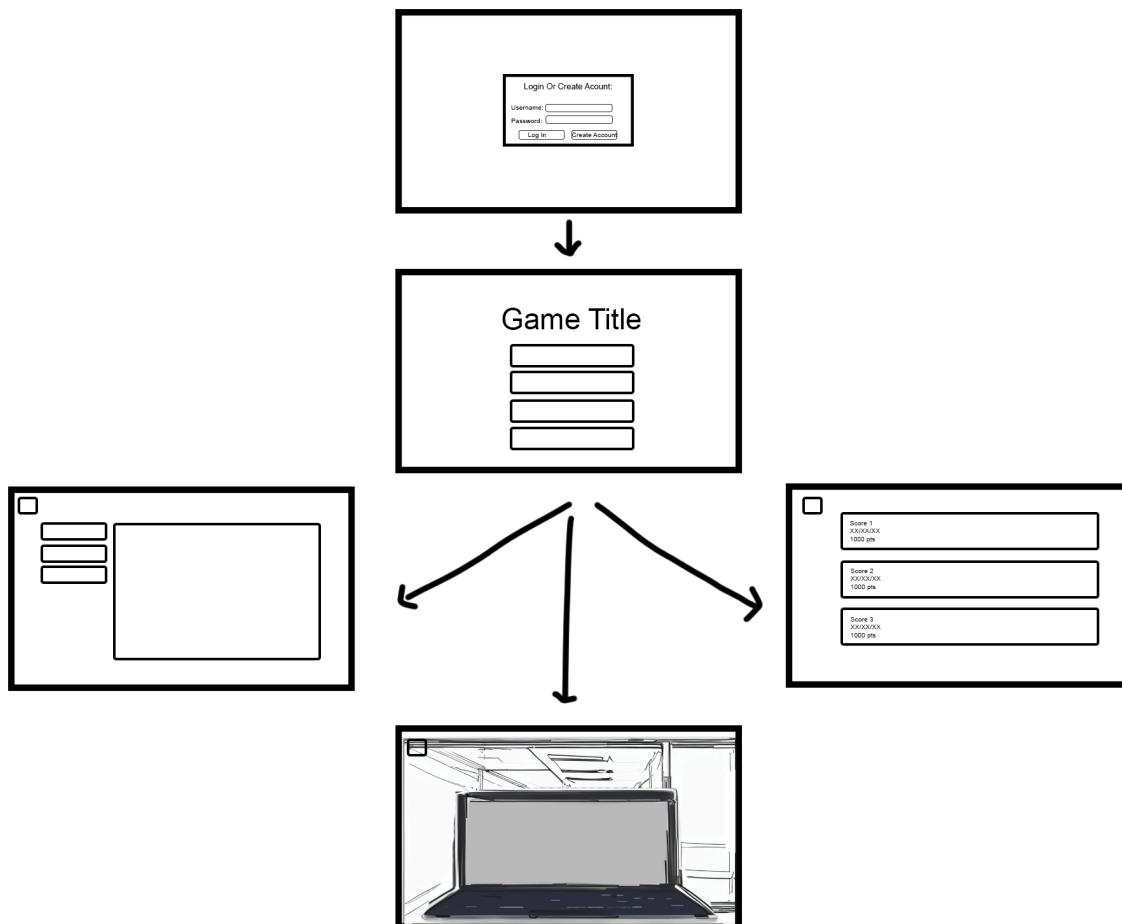
When the user opens the game, they will be prompted with a screen that asks them to enter their username and password or create an account. There will be a menu screen with the title on the top section of the screen and multiple buttons in the center.

The buttons will have a play game button that will take the user to the game screen, where they can start the gameplay loop. On the main screen, another button will take the user to a score page, where they can see their previous game runs. A third button on the main screen will take the user to the settings page, where they can alter the game settings. A fourth button will be an exit game button, where the user can close the application.

In the gameplay screen, there will be a pause menu button that will pause the game, and open up a menu with options to exit the menu, mute or unmute audio, restart the game, and exit the application.

The gameplay screen will also have a status bar to display their score, and how many failed attempts they are at; with three failed attempts, a game over text will show, along with their score, and a button to restart, or take the user back to the title screen.

All pages will also have back buttons that take the user to the previous screen.



## **Appendix A: Glossary**

*PK: Primary Key*

*FK: Foreign Key*