**Software Requirements**

**Specification**

**for**

**Mini Games**

**Version 0**

**Prepared by Andrew Fales, Jacob Lin, Selena Sat, Tyler Anton**

**Righteous Gamers**

**4/8/2024**

**Table of Contents**

1. Introduction.......................................................................................................... 3

1.1 Problem statement......................................................................................... 3

1.2 Summary........................................................................................................ 3

1.3 Product Scope................................................................................................. 3

1.3.1 In scope........................................................................................................... 3

1.3.2 Out of scope.................................................................................................... 3

1.4 References...................................................................................................... 3

2. Overall Description............................................................................................... 3

2.1 Compete analysis............................................................................................ 3

2.2 User Classes and Customer Profile................................................................... 3

2.3 Design and Implementation Constraints........................................................... 4

2.4 Assumptions and Dependencies....................................................................... 4

3. Specific Requirements.......................................................................................... 4

3.1 User Interfaces................................................................................................ 4

3.2 Functional Requirements................................................................................. 4

3.3 Logical Database Requirements (Remove and keep only in Design)................... 4

3.4 Performance and Software Quality Requirements............................................ 5

4. Breakdown of work/ Project timeline plan......................................................... 5

**Revision History**

| **Name** | **Date** | **Reason For Changes** | **Version** |
| --- | --- | --- | --- |
|  | 4/8/2024 | Initial Version | 1.0 |
|  | 5/6/2024 | Mid quarter revision | 2.0 |

# **1.** **Introduction**

## **1.1** **Problem statement**

Users want to play easily accessible online minigames and potentially track their scores.

Users want access to quick accessible recreational activities, to do alone or with their friends.

Users also want an experience involving a persistent account.

## **1.2** **Summary**

We want to develop an accessible and user friendly experience across multiple minigames, where users can track scores for themselves and other users and create unique profiles.

## **1.3** **Product Scope**

### **1.3.1** **In scope**

User A: someone who wants to compare scores with other people (user account)

User B: someone who wants to play very simple minigames (no user account)

Website Features:

* A mini game for users to play. (Tetris, dino run, flappy bird, snake game, minesweeper)
* Main landing page
  + Navbar
  + Login form
  + Game links
* UX/UI, Improve look and feel of website. Lightweight, accessible, and minimalistic. Ex. New York Times game site.
* An account system, where users are able to sign up to our website to keep track of personal scores.
  + Account creation
  + User info/edit
* Individual core record keeping and per game global scoreboard. The backend of the account system.
  + SQLite3 user and high score database
  + PHP Backend to interact with database

### **1.3.2** **Out of scope**

User C: someone who wants to meet other people who play similar minigames online (Friend system)

User D: someone who wants to compete with other people in real time, real time competitive games (tetris with friends for example)

User E: Guest account (score gets tracked but no username)

Website Features:

* User friend system.

## **1.4** **References**

<https://www.nytimes.com/crosswords> - good example of how our home screen should look like

# **2.** **Overall Description**

## **2.1** **Compete analysis**

Coolmathgames - aggregate site for minigames, has a login system

ArmorGames - aggregate site for all types of (originally flash) games, has login and high score system

Kongregate - aggregate site for all types of games, has log in and high score system

NYT Minigames (https://www.nytimes.com/crosswords) - Simple and accessible games that are updated daily.

For most of these competitors, they don’t really have the simplest minigames that are easily accessible. (e.g. dino run game from user chrome, flappy bird, etc). For competitors such as the NYT crosswords games, they are focused entirely on word based games which limits their game audience.

## **2.2** **User Classes and Customer Profile**

User A: User with an account

User B: User without accounts

## **2.3** **Design and Implementation Constraints**

The website will only be supported on desktop and only support modern internet browsers. Users must have an internet connection to play, so we can update global scoreboards. Website will be cloud hosted using Amazon’s AWS EC2 and S3. The website will be limited in user count and accessibility, and will not be designed with scalability in mind.

Godot will be used as a game engine to develop and port minigames to be played in the browser by each user.

No hardware or capital to support high user count.

Users require internet connection.

Cloud hosted on aws EC2 and S3.

Security is important as the website will be public.

HTML/CSS, JavaScript, PHP, SQLite3

Only support desktop.

## **2.4** **Assumptions and Dependencies**

AWS EC2 and S3

Using Godot as a game engine.

SQLite3

# **3.** **Specific Requirements**

Requirement id - 1 (traceability matrix).

## **3.1** **User Interfaces**

Main home screen:

User profile

Assorted mini games

Login

User profile:

Login/out

Scoreboard

Settings

Change username/password

Assorted mini games (tile based navigation):

Home

Scoreboard

Login/out:

User profile

Home

Scoreboard:

User profile

## **3.2** **Functional Requirements**

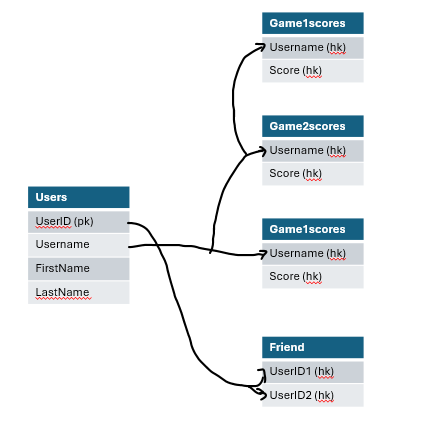
User account system

* Login, logout
* Change password, change username
* Security - password encryption
* SQLite3 Database
* PHP Backend
* Database updates
* Scoreboard system
  + Displays user high scores
  + Database updates

Minigames

* Playable directly from the browser.
* Has access to game data to be able to support individual user data for score keeping.

***3.3*** **Logical Database Requirements**



## **3.4** **Performance and Software Quality Requirements**

Requirement id - 1 (traceability matrix).

We want the website to be lightweight, accessible on most modern browsers and hardware, and reliable.

We want the minigames to be simple (easy to understand), easily accessible (few buttons), and easily quantifiable (scores). Minigames should use minimal resources.

# **4.** **Breakdown of work/ Project timeline plan**

We will work in an agile development cycle.

#### **Collect a numbered list of the project user stories (to be determined) references that remain in the SRS so they can be tracked to closure>**

| **User Story** | **Milestone** | **Assigned to** |
| --- | --- | --- |
| 0 GITHUB/TRELLO SETUP | 0 |  |
| 1 home page (just html/frontend) | 1 |  |
| 2 game page | 1 |  |
| 3 user profile | 1 |  |
| 4 login screen | 1 |  |
| 5 scoreboard screen | 1 |  |
| 6 database set up | 2 |  |
| 7 user login system | 2 |  |
| 8 website security | 2 |  |
| 9 dino run | 4 |  |
| 10 scoreboard system | 4 |  |
| 11… all other minigames | 5 |  |

Revision Blurb:

2.3 - Changed planned hosting to AWS EC2 and S3 instance, added security requirement, added a bit talking about browser dependencies for Godot.

2.4 - Confirmed what dependencies we will be using, no more question marks.

3.1 - Confirmed scope and removed any out of scope features.

3.2 - Added more specificity to functional requirements

Overall most changes were made as a result of further research and finalizing design decisions. Some changes were made to be more specific about certain aspects/functions of the project.