

SRS Documentation— MathGames CS-362

Jared Dyreson, Peyman Dinani, Kevin Durate
California State University, Fullerton

Contents

1	Introduction	3
1.1	Purpose	3
1.2	Scope of the problem	3
1.3	Intended Audience	3
2	Overall Description	4
2.1	User Objectives	4
2.2	Product Functions	4
2.3	Operating Environment	4
2.4	Similar System Information	4
2.5	User Characteristics	5
2.6	Design and Implementation Constraints	5
2.7	Assumptions and Dependencies	5
3	Functional Requirements	6
3.1	The app shall display a login page	6
3.2	The application shall communicate with the web server	6
3.3	The application shall display the game screen	6
3.4	The application shall play a happy sound and an overlay	6
3.5	The application shall play a sad sound and an overlay	7
4	Quality Attributes for MathGames	8
5	Non-Functional Requirements	9
5.1	Security	9
5.2	Usability	9
5.3	Backup	9
6	Interface Requirement	10
6.1	GUI	10
6.2	Hardware Interface	10
6.3	Software Interface	10
7	SWOT Analysis for MathGames	11
7.1	Strengths	11
7.2	Weaknesses	11
7.3	Opportunities	11
7.4	Threats	12

8	UML Diagrams	13
8.1	Use Case Diagram	13
8.2	Sequence Diagram	13
8.3	Activity Diagram	14
9	References	15

1 Introduction

1.1 Purpose

The following document contain the software requirements specified for our CPSC362 MathGames game. The document will go through the detail, purpose, implementation, and software requirements of MathGames.

1.2 Scope of the problem

Learning math can be challenging at times. MathGames is designed to take advantage of the internet of things to make practicing math simple, enjoyable and even competitive. By utilizing easy to use and seamless networking with other MathGames users, it provides an environment to better engage the math learners. MathGames distracts the players of the tedious nature of practicing math by letting players compete against each other in an enjoyable way.

1.3 Intended Audience

MathGames is designed for first graders with simplicity in mind. The main focus of MathGames is ease of use and connecting with other MathGames players.

2 Overall Description

2.1 User Objectives

The objective for the end user is to get a better understanding of basic mathematics, mostly intended for children of the ages five to seven.

2.2 Product Functions

This product acts as a simple application to answer questions presented by the program to test the child's knowledge.

2.3 Operating Environment

This product has been tested on both Windows and Linux based operating systems. Intended target for server deployment is Ubuntu server, however other distributions should work out of the box with limited configuration. For local testing, system administrators can run the project locally and are able to connect to it that way.

The steps for this are as follows:

- Navigate to the root folder of the MathGames project
- Open another console window in that current folder
- Run the following command to instantiate the web server “python Web-server/Server.py”
- After that, you can run “python main.py” in the other console window, and there should appear an interface asking for login credentials

2.4 Similar System Information

There are hundreds of other applications that serve the same functionality as MathGames. However, most of them are single player and do not involve other peers.

2.5 User Characteristics

The intended audience does not need any prior technical knowledge and can use this software independently.

2.6 Design and Implementation Constraints

The design of this software has led to a simple timing issue between the client machine and the deployed web server. Since the Python web server sends information every second, the graphical interface needs to keep up with it. That being said, whatever is added in further iterations must keep this dynamic in mind.

2.7 Assumptions and Dependencies

This SRS assumes that:

- User understands simple English
- User understands how to do basic arithmetic, such as addition and subtraction
- An estimate of less than 1 megabyte of disk space is required to install this application

List of dependencies include:

- [websocket-client](#) — an API that interfaces with WebSocket
- [pyglet](#) — an API that allows for the playing of audio cross-platform

These dependencies can be installed by running “pip install websocket-client pyglet” and should be handled by the system administrator. It is not expected for the end user to handle installation of dependencies.

3 Functional Requirements

3.1 The app shall display a login page

Description: upon clicking the “main.py” file in the root of the project directory, two different windows can appear. If there are no errors with establishing a connection to the server, it will prompt them for their name. Else, it will cease execution and inform the user to enable the server.

Pre-condition: User has downloaded MathGames.

Post-condition: The login screen is displayed.

3.2 The application shall communicate with the web server

Description: A connection is established between the client and the server to begin the game

Pre-condition: User has entered the correct information in the login screen

Post-condition: The user will be greeted with the game window

3.3 The application shall display the game screen

Description: The game will begin and the user can select input to send to the server

Pre-condition: The game window has appeared to the user

Post-condition: The user can thus begin playing.

3.4 The application shall play a happy sound and an overlay

Description: The game will utter a chime that indicates that the user has given the correct input.

Pre-condition: Correct response was given

Post-condition: Chime is played to the user and a small window is shown, displaying a happy face

3.5 The application shall play a sad sound and an overlay

Description: The game will utter a sad sound that indicates that the user has given the incorrect input.

Pre-condition: Incorrect response was given.

Post-condition: Sound is played and a small window is shown, displaying a frowning face.

4 Quality Attributes for MathGames

Quality Attribute	Brief Description
RT-1	The game will respond to server availability based on the quality of internet connection.
RT-1	The game will show an error/ notify user if the server is not available within seconds of launching.
SS-1	The game will ask the player name right after launching and checking for server availability.
SA-1	The game will be available as long as the server is up and running.
AC-1	The game will be accessible by windows or mac computers with python installed and internet connectivity

- **RE: Response Time**
- **SS: System Security**
- **SA: System Availability**
- **AC: System Accessibly**

5 Non-Functional Requirements

5.1 Security

There are no current security threats in the latest build of MathGames.

5.2 Usability

- The user will require a working internet connection
- After starting the application, the user will be required to enter their name in a small text field

5.3 Backup

- The latest score of the user will be saved on the server as long as the server has not been reset

6 Interface Requirement

6.1 GUI

The application is run typing in “./main.py” in the root of the project directory. A small window should appear prompting you to type in a username. After that, a connection is then established between the client and the server, allowing access to the game. The game will then begin, wherein the user can punch in their attempt at getting the right answer. If they are successful, there will be a splash screen telling you “good job” and a happy sound effect will play. Otherwise, another splash screen will appear and a sad sound effect will appear.

6.2 Hardware Interface

This application is expected to be used on a desktop computer that has an Ethernet connection. Optionally, you can run this on a laptop with a wireless connection, neither option will degrade the experience of MathGames.

6.3 Software Interface

Please refer to the Operating Environment section here [Operating Environment](#)

7 SWOT Analysis for MathGames

7.1 Strengths

- S1: The application has very little load on the CPU, allowing many lower-powered devices to run this application with minimal resources.
- S2: The application features eye catching graphics that are intended for younger generations.
- S3: The application has a minimalistic interface, which is simple for the user to memorize and use.

7.2 Weaknesses

- W1: The application asks questions that result in answers from 0 to 10 which may become an issue when the user becomes proficient enough for more advanced mathematics.
- W2: The application requires the server to be running as it is intended to be a multiplayer game.
- W3: The application does not pair users based on their skill set. Thus, a new player can possibly be playing against a very experienced player - causing an unfair game.

7.3 Opportunities

- O1: This application serves the purpose of providing a quick and intuitive way for younger audiences to practice simple arithmetic. Many other mathematics applications have complex settings, goals, and gameplay mechanics. This program does what it is intended to do and nothing more. This will attract many users as they know exactly what the application is for.
- O2: This application offers a quick way to play math games in seconds after installing. No need to configure settings or fiddle around with creating an account, just launch the application and type in your name. Many other

mathematical applications require account sign ups, long setup process, and more. Parents and guardians can get their child playing almost immediately after installing.

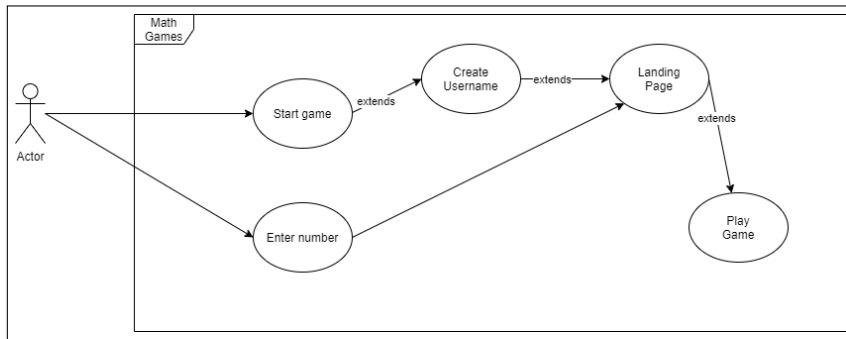
- O3: This application does not store any sensitive data nor allow for communication with other players. With the growing fear of software collecting data, parents and guardians often worry what data is being collected on their child. This application keeps parents and guardians at ease since there is no sensitive data collected.

7.4 Threats

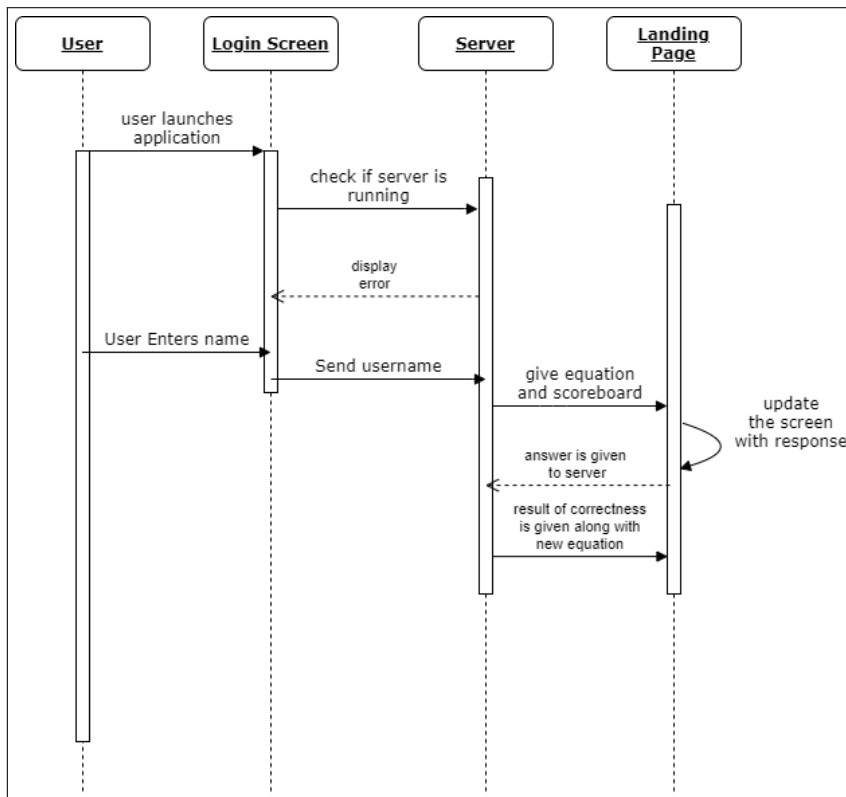
- T1: There is no profit to be made from this application as it is free to play without in-app purchases. Thus, sustaining this application will require our company to gather volunteers or pay developers (at a loss) to continually maintain and update it. This is a threat as our company will continually lose money on the application.
- T2: The audience for this application is very limited as it is intended for very young students. Once a student advances in school, they will most likely not use the application again. This will make it very difficult to achieve long-term users. This is a threat as long-term users help our company grow. If we cannot sustain a strong fan-base, it will be very difficult to expand our company.
- T3: The application is not available for mobile devices, which is a problem since younger audiences use mobile devices more than personal computers. As phones become more ubiquitous than personal computers, the less market share we are able to obtain.

8 UML Diagrams

8.1 Use Case Diagram



8.2 Sequence Diagram



8.3 Activity Diagram

9 References

- [Draw.io](#) — feature rich site to create the UML diagrams in this document
- [L^AT_EX](#) — powerful typesetting language to create this document
- [StackOverflow](#) — for general questions regarding specific areas of interest and program behavior

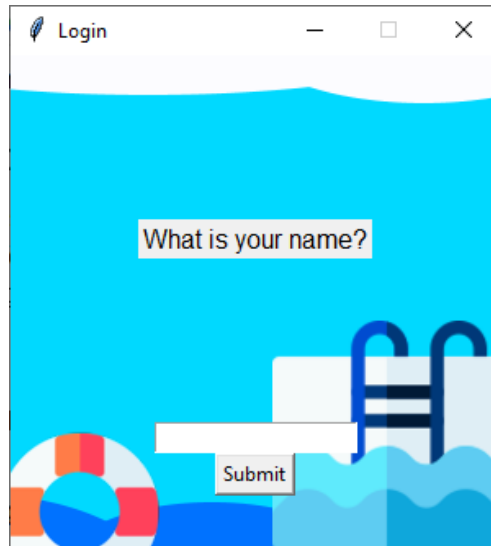


Figure 1: Login Screen

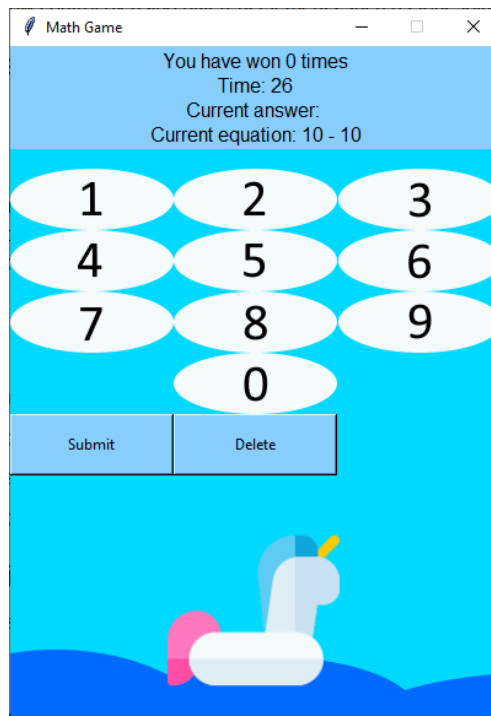


Figure 2: Game Screen

