

# SRS Progress Sheet Updates

**Jaswanthi Jagarlamudi**

05-15-23	1.Revised grammar and sentence structure
05-08-23	1.Updated all the sections of SRS
05-01-23	1.Updated System Features 6 <ul style="list-style-type: none"><li>Adjust questions based on difficulty level<ol style="list-style-type: none"><li>Description and Priorities</li><li>Stimulus/Response</li><li>Functional Requirements</li></ol></li></ul> 2.Updated UML, Sequence Diagram 3.Updated Addition Practice Screen
04-24-23	1.Updated System Features 5 <ul style="list-style-type: none"><li>Addition Practice screen<ol style="list-style-type: none"><li>Description and Priorities</li><li>Stimulus/Response</li><li>Functional Requirements</li></ol></li></ul> 2.Updated Software Quality Attributes 2.Updated UML Use Case, Class, Sequence Diagrams
04-17-23	1.Updated other requirements <ul style="list-style-type: none"><li>Storage solution</li><li>Design</li><li>Task</li><li>Cohesiveness</li></ul>
04-10-23	1.Updated Congratulations Screen 2.Updated Other Non-Functional Requirements <ul style="list-style-type: none"><li>Performance Requirements</li><li>Safety Requirements</li></ul>

04-03-23	<p>1.Updated System Features 4</p> <ul style="list-style-type: none"> <li>● Adding Puzzle screen             <ol style="list-style-type: none"> <li>1. Description and Priorities</li> <li>2. Stimulus/Response</li> <li>3. Functional Requirements</li> </ol> </li> </ul> <p>2.Updated UML Class, Sequence diagrams</p>
03-20-23	<p>1.Added Adding Quiz screen</p> <p>2.Updated System Features 3</p> <ul style="list-style-type: none"> <li>● Adding Quiz screen             <ol style="list-style-type: none"> <li>1. Description and Priorities</li> <li>2. Stimulus/Response</li> <li>3. Functional Requirements</li> </ol> </li> </ul> <p>2.Updated UML Class, Sequence diagrams</p>
03-13-23	<p>1.Updated System Features 2</p> <ul style="list-style-type: none"> <li>● Generate random numbers and validate the answers             <ol style="list-style-type: none"> <li>1. Description and Priorities</li> <li>2. Stimulus/Response</li> <li>3. Functional Requirements</li> </ol> </li> </ul> <p>2.Updated UML Class, Sequence diagrams</p>
03-06-23	<p>1.Updated System Features 1</p> <ul style="list-style-type: none"> <li>● Title / Menu Screen             <ol style="list-style-type: none"> <li>1. Description and Priorities</li> <li>2. Stimulus/Response</li> <li>3. Functional Requirements</li> </ol> </li> </ul> <p>2.Updated UML Use Case, Class, Sequence diagrams</p>
02-27-23	<p>1.Updated hardware interfaces required</p> <p>2.Updated performance requirements</p> <p>3.Updated Use Case, Class, Sequence diagrams</p>
02-20-23	<p>1.Updated Overall Description</p> <p>2.Added Appendix B: Analysis models</p> <p>A. UML Use Case diagram</p>

	<p>B. UML Class diagram</p> <p>C. UML Sequence diagram</p> <p>3.Added Initial Screen, Home Screen, Puzzle Screen, Store Screen to the user interface</p>
02-13-23	<p>1.Updated Introduction</p> <ul style="list-style-type: none"> <li>● Purpose</li> <li>● Intended audience and reading suggestions</li> <li>● Product scope</li> </ul> <p>2.Updated external interface requirements</p> <ul style="list-style-type: none"> <li>● User Interfaces</li> <li>● Software Interfaces</li> </ul>
02-06-23	<p>1.Worked on UML Diagram, Splash screen</p>
01-30-23	<p>1.Created initial version of document</p> <p>2.Updated SRS with Table of Contents</p>

---

# **Software Requirements Specification**

**for**

## **Kid Math Game with Cat**

**Version 2.3**

**Prepared by Team 1 of Software Engineering Class**

**California State University, San Bernardino**

**05-15-23**

*Copyright © 1999 by Karl E. Wieggers. Permission is granted to use, modify, and distribute this document*

# Table of Contents

<b>Table of Contents</b>	<b>1</b>
<b>Revision History</b>	<b>3</b>
<b>List of Figures</b>	<b>5</b>
<b>1. Introduction</b>	<b>6</b>
1.1 Purpose	6
1.2 Document Conventions	6
1.3 Intended Audience and Reading Suggestions	6
1.4 Product Scope	6
1.5 References	6
<b>2. Overall Description</b>	<b>7</b>
2.1 Product Perspective	7
2.2 Product Functions	7
2.3 User Classes and Characteristics	7
2.4 Operating Environment	8
2.5 Design and Implementation Constraints	8
2.6 User Documentation	8
2.7 Assumptions and Dependencies	8
<b>3. External Interface Requirements</b>	<b>8</b>
3.1 User Interfaces	8
3.1.1 Initial Screen/Loading Screen	8
3.1.2 Main Screen/Home Screen	9
3.1.3 Adding Puzzle Screen	9
3.1.4 Adding Fun Screen	10
3.1.5 Adding Quiz Screen	10
3.1.6 Addition Practice Screen	11
3.1.7 Addition Practice Beginner Level Screen	11
3.1.8 Addition Practice Intermediate Level Screen	12
3.1.9 Addition Practice Advanced Level Screen	12
3.1.10 Congratulations Screen	13
3.2 Hardware Interfaces	13
3.3 Software Interfaces	14
3.4 Communications Interfaces	14

<b>4. System Features</b>	<b>14</b>
4.1 Title/Menu Screen	14
4.1.1 Description and Priorities	14
4.1.2 Stimulus/Response	15
4.1.3 Functional Requirements	15
4.2 Generate random numbers and validate the answers	15
4.2.1 Description and Priority	15
4.2.2 Stimulus/Response	15
4.2.3 Functional Requirements	16
4.3 Adding Quiz Screen	16
4.3.1 Description and Priority	16
4.3.2 Stimulus/Response	16
4.3.3 Functional Requirements	16
4.4 Adding Puzzle Screen	17
4.4.1 Description and Priority	17
4.4.2 Stimulus/Response	17
4.4.3 Functional Requirements	17
4.5 Addition Practice Screen	18
4.5.1 Description and Priority	18
4.5.2 Stimulus/Response	18
4.5.3 Functional Requirements	18
4.6 Adjust Questions based on difficulty level	19
4.6.1 Description and Priority	19
4.6.2 Stimulus/Response	19
4.6.3 Functional Requirements	19
<b>5. Other Nonfunctional Requirements</b>	<b>19</b>
5.1 Performance Requirements	19
5.2 Safety Requirements	19
5.3 Security Requirements	20
5.4 Software Quality Attributes	20
5.5 Business Rules	20
<b>6. Other Requirements</b>	<b>21</b>
6.1 Storage Solution	21
6.2 Design	21

6.3 Task	21
6.4 Cohesiveness	21
<b>Appendix A: Glossary</b>	<b>21</b>
<b>Appendix B: Analysis Models</b>	<b>22</b>
A. UML Use Case Diagram	22
B. UML Class Diagram	25
C. UML Sequence Diagram	29
<b>Appendix C: To Be Determined List</b>	<b>31</b>

## Revision History

Name	Date	Reason for Changes	Version
Jaswanthi Jagarlamudi	02-06-23	1.Created initial version of document 2.Updated SRS with Table of contents	1.0
		1.Updated Introduction	



Jaswanthi Jagarlamudi	02-13-23	<ul style="list-style-type: none"> <li>• Purpose</li> <li>• Intended Audience and Reading Suggestions</li> <li>• Product Scope</li> </ul> 2.Updated External Interface Requirements <ul style="list-style-type: none"> <li>• User Interfaces</li> <li>• Software Interfaces</li> </ul> 1) Unity 1) Visual Studio 2) C# and .NET	1.1
Jaswanthi Jagarlamudi	02-20-23	1.Updated Overall Description <ul style="list-style-type: none"> <li>• Product Perspective</li> <li>• Product Functions</li> <li>• User classes and characteristics</li> <li>• Operating Environment</li> </ul> 2.Added Appendix B: Analysis Models A. UML Use Case Diagram B. UML Class Diagram C. UML Sequence Diagram 3.Added Initial Screen, Home Screen, Puzzle Screen, Store Screen to the user interface	1.2
Jaswanthi Jagarlamudi	02-27-23	1.Updated hardware interfaces required 2.Updated performance requirements 3.Updated use case, class, sequence diagrams	1.3
Jaswanthi Jagarlamudi	03-06-23	1.Updated System Features 1 <ul style="list-style-type: none"> <li>• Title / Menu Screen <ol style="list-style-type: none"> <li>1. Description and Priorities</li> <li>2. Stimulus/Response</li> <li>3. Functional Requirements</li> </ol> </li> </ul> 2.Updated UML class, sequence diagrams	1.4
	03-13-23	1.Updated System Features 2 <ul style="list-style-type: none"> <li>• Generate random numbers and validate the answers</li> </ul>	

Jaswanthi Jagarlamudi		1. Description and Priorities 2. Stimulus/Response 3. Functional Requirements 2.Updated UML use case, class, sequence diagrams	1.5
Jaswanthi Jagarlamudi	03-20-23	1.Added Adding Quiz screen 2.Updated System Features 3 <ul style="list-style-type: none"> <li>Adding Quiz screen               <ol style="list-style-type: none"> <li>1. Description and Priorities</li> <li>2. Stimulus/Response</li> <li>3. Functional Requirements</li> </ol> </li> </ul> 3.Updated UML use case, class, sequence diagrams	1.6
Jaswanthi Jagarlamudi	04-03-23	1.Updated System Features 4 <ul style="list-style-type: none"> <li>Adding Puzzle screen               <ol style="list-style-type: none"> <li>1. Description and Priorities</li> <li>2. Stimulus/Response</li> <li>3. Functional Requirements</li> </ol> </li> </ul> 2.Updated UML use case, class, sequence diagrams	1.7
Jaswanthi Jagarlamudi	04-10-23	1.Updated Congratulations Screen. 2.Updated Other Non-Functional Requirements <ul style="list-style-type: none"> <li>Performance Requirements</li> <li>Safety Requirements</li> </ul>	1.8
Jaswanthi Jagarlamudi	04-17-23	1.Updated other requirements. <ul style="list-style-type: none"> <li>Storage solution</li> <li>Design</li> <li>Task</li> <li>Cohesiveness</li> </ul>	1.9
Jaswanthi Jagarlamudi	04-24-23	1.Updated System Features 5 <ul style="list-style-type: none"> <li>Addition Practice screen               <ol style="list-style-type: none"> <li>1. Description and Priorities</li> <li>2. Stimulus/Response</li> <li>3. Functional Requirements</li> </ol> </li> </ul> 2.Updated Software Quality Attributes	2.0

		3.Updated UML use case, class, sequence diagrams	
Jaswanthi Jagarlamudi	05-01-23	1.Updated System Features 6 <ul style="list-style-type: none"> <li>Adjust questions based on difficulty level</li> <li>1. Description and Priorities</li> <li>2. Stimulus/Response</li> <li>3. Functional Requirements</li> </ul> 2.Updated UML use case, class, sequence diagrams 3.Updated Addition Practice Screen	2.1
Jaswanthi Jagarlamudi	05-08-23	1.Updated all the sections of the documentation	2.2
Calvin Glisson	05-15-23	1.Revised grammar and sentence structure	2.3

## List of Figures

<b><u>Figure 01-</u></b> Loading Screen/Initial Screen for Kid Math Game with Cat	09
<b><u>Figure 02-</u></b> Main Screen/Home Screen for Kid Math Game with Cat	09
<b><u>Figure 03-</u></b> Adding Puzzle Screen for Kid Math Game with Cat	10
<b><u>Figure 04-</u></b> Adding Fun Screen for Kid Math Game with Cat	10
<b><u>Figure 05-</u></b> Adding Quiz Screen for Kid Math Game with Cat	11
<b><u>Figure 06-</u></b> Addition Practice Screen for Kid Math Game with Cat	11
<b><u>Figure 07-</u></b> Addition Practice Beginner Level Screen for Kid Math Game with Cat	12
<b><u>Figure 08-</u></b> Addition Practice Intermediate Level Screen for Kid Math Game with Cat	12
<b><u>Figure 09-</u></b> Addition Practice Advance Level Screen for Kid Math Game with Cat	13
<b><u>Figure 10-</u></b> Congratulations Screen for Kid Math Game with Cat	13
<b><u>Figure 11-</u></b> UML Use Case Diagram 1	22
<b><u>Figure 12-</u></b> UML Use Case Diagram 2	23
<b><u>Figure 13-</u></b> UML Use Case Diagram 3	23

<a href="#"><u>Figure 14-</u></a> UML Use Case Diagram 4	24
<a href="#"><u>Figure 15-</u></a> UML Use Case Diagram 5	24
<a href="#"><u>Figure 16-</u></a> UML Use Case Diagram 6	25
<a href="#"><u>Figure 17-</u></a> UML Class Diagram 1	26
<a href="#"><u>Figure 18-</u></a> UML Class Diagram 2	26
<a href="#"><u>Figure 19-</u></a> UML Class Diagram 3	27
<a href="#"><u>Figure 20-</u></a> UML Class Diagram 4	28
<a href="#"><u>Figure 21-</u></a> UML Sequence Diagram 1	29
<a href="#"><u>Figure 22-</u></a> UML Sequence Diagram 2	30
<a href="#"><u>Figure 23-</u></a> UML Sequence Diagram 3	30
<a href="#"><u>Figure 24-</u></a> UML Sequence Diagram 4	31

# 1. Introduction

## 1.1 Purpose

The game application “Kid Math Game with Cat” is an attractive, interactive, and entertaining game. This game is created using the Unity engine. Kids can enhance their mathematical skills by playing this game for fun. The game teaches counting and addition and provides a challenging and engaging educational experience for kids.

## 1.2 Document Conventions

The document was developed using the IEEE’s Software Requirement Specification.

## 1.3 Intended Audience and Reading Suggestions

The SRS provides a way for the user to verify that the game developed is in line with the original idea. To have a general overview of the project, view the description. Part 2: For a detailed explanation of the game play elements and how they connect to the character, see System Features. Part 3: If you are excited about the game’s interface and how to use the front-end menus, view External Interface Requirements. Part 4: The technical requirements that the project will hold are listed in Nonfunctional Requirements.

## 1.4 Product Scope

The game “Kid Math Game with Cat” aids in teaching kids counting and addition concepts. The various game styles make the concept understandable to players of any skill level. Additionally, this game offers a fun exercise to kids where kids can learn the concept of addition using different items. Hints are also introduced to the users to help them understand the concepts and solve the levels.

## 1.5 References

- **Software Engineering: A Practitioner's Approach**  
<https://ebookcentral.proquest.com/lib/csusb/detail.action?docID=6328275&pq-origsite=primo>
- **GitHub page: Kids Math Game with Cat**  
[Team-1-Spring23/Math-App \(github.com\)](https://github.com/Team-1-Spring23/Math-App)
- **IEEE Template for System Requirement Specification Documents:**  
<https://goo.gl/nsUFwy>
- **Math Kids – Add, Subtract, Count app.**  
[Math Kids - Add, Subtract, Count on the App Store \(apple.com\)](https://apps.apple.com/us/app/math-kids-add-subtract-count/id1444444444)
- **NuGet** is a package manager designed to enable developers to share reusable code.  
<https://www.nuget.org/>

## 2. Overall Description

### 2.1 Product Perspective

The game is intended to closely imitate the original kid math game app that is available in the play store. The primary goal of this game is to make the mathematical concept of addition simpler and more enjoyable for kids to learn.

## 2.2 Product Functions

Helping kids learn basic addition by implementing addition games of various difficulty levels. Implementing high quality cartoon animations wherever necessary for the effectiveness of the child's learning and enjoyment. The following is a summary of the major features implemented in the game. They are separated into categories depending on their function:

- **Title / Menu Screen:** This is the application's initial viewable screen, which includes buttons for each game, consisting of Adding Quiz, Adding Puzzle, Adding Fun, and Addition Practice.
- **Cat Character:** A cat character is created to help engage children.
- **Generating random numbers and validating them:** Different numbers are generated along with the correct response for the addition games.
- **Adjust questions based on difficulty level:** There will be an intermediary screen where the game redirects us to the next levels based on the kid's performance from Beginner to Intermediate to Advanced.
- **Generate questions and validate the answers with reactions:** Adding puzzles are generated and the cat character reacts depending on whether the user gives a correct or incorrect answer.
- **Kid learning progression track:** For every successive puzzle the kid plays, he/she will move to the next level of the game which tracks the development of his/her performance.

## 2.3 User Classes and Characteristics

The control scheme is designed to be intuitive, and the game play is fair enough to be understood by everyone. Therefore, the experience with the game will not be a major factor in dividing the end users. However, any game has some basic division among its end users as hardcore players and casual players. The hardcore players for this game are obviously kids who are attending their primary school.

## 2.4 Operating Environment

This application will be launched on both the play store (Android) and App Store (iOS). A web-based platform is not currently intended.

## **2.5 Design and Implementation Constraints**

The kid math game is a very minimalistic app both in its functionality and user experience. As of now there are no constraints that are noticed for the developers.

## **2.6 User Documentation**

While this app is a simple and minimalistic one, end users will not face much difficulty.

We have not implemented any complex operations that pose compatibility issues. Kids who are about to start their learning journey can use the app without facing any difficulties.

## **2.7 Assumptions and Dependencies**

We will use Unity 2D for graphics. After the testing phase, we will decide the minimum requirements and oldest android version to be supported, then release the game on the market. As of now we have not noticed any dependencies for this application as the application does not have any complex operations and it is a standalone application.

# **3. External Interface Requirements**

The interface specifications for the system are described in this section of the SRS. User, hardware, software, and communication interface requirements are defined.

## **3.1 User Interfaces**

### **3.1.1 Initial Screen/Loading Screen:**

This screen is the initial loading screen for Kid Math Game with Cat.



**Figure 01:** Loading Screen/Initial Screen for Kid Math Game with Cat

### 3.1.2 Main Screen/Home Screen:

This screen consists of buttons for the Adding Puzzle, Adding Fun, Adding Quiz, and Addition Practice games.

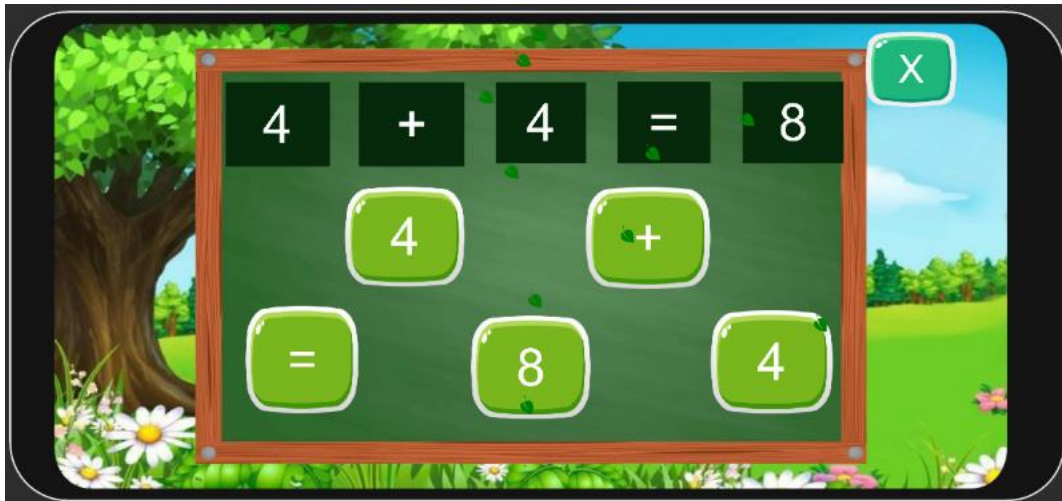


**Figure 02:** Main Screen/Home Screen for Kid Math Game with Cat



### 3.1.3 Adding Puzzle Screen:

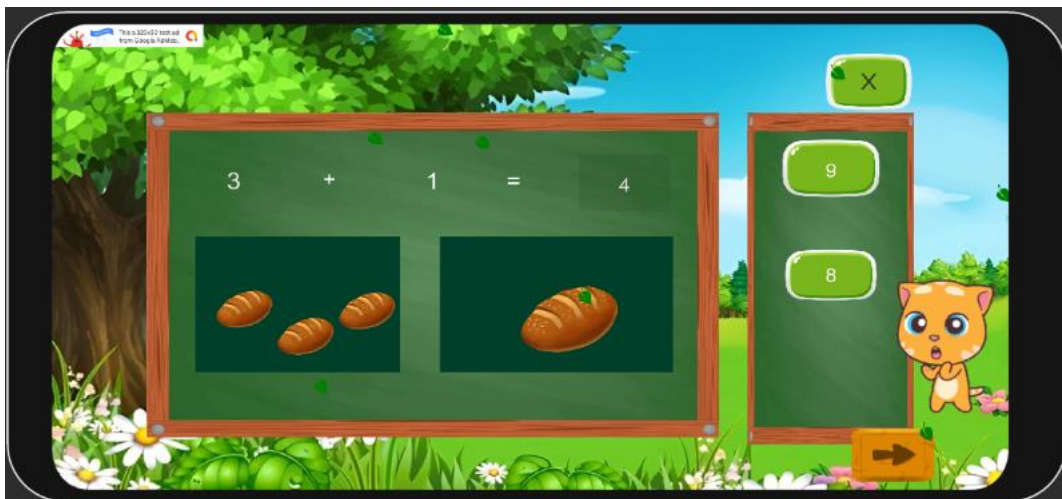
This screen consists of solving addition puzzles with different numbers and addition operators which help the kid learn addition by solving the puzzle displayed.



**Figure 03:** Adding Puzzle Screen for Kid Math Game with Cat

### 3.1.4 Adding Fun Screen:

This screen is a game for adding numbers using different objects to represent the number of items being added, so that the kids can choose the correct answer by counting the number of items displayed on the screen.



**Figure 04:** Adding Fun Screen for Kid Math Game with Cat

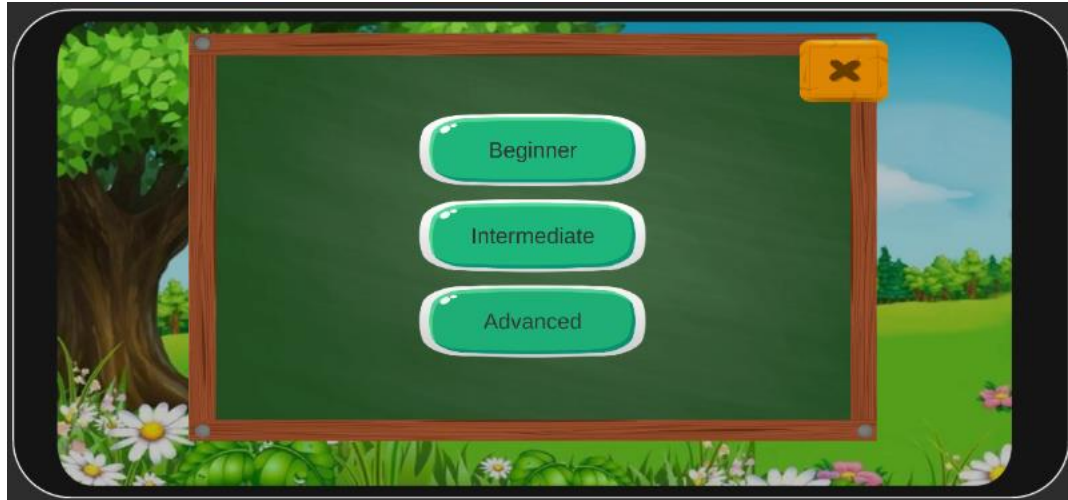
### 3.1.5 Adding Quiz Screen:

This screen consists of adding quizzes with different numbers. Kids can perform addition operations and choose appropriate answers.

**Figure 05:** Adding Quiz Screen for Kid Math Game with Cat

### 3.1.6 Addition Practice Screen:

This screen consists of buttons for the difficulty levels beginner, intermediate and advanced, where the kid can choose to solve addition practice problems according to his level of progress.



**Figure 06:** Addition Practice Screen for Kid Math Game with Cat

### 3.1.7 Addition Practice Beginner Level Screen:

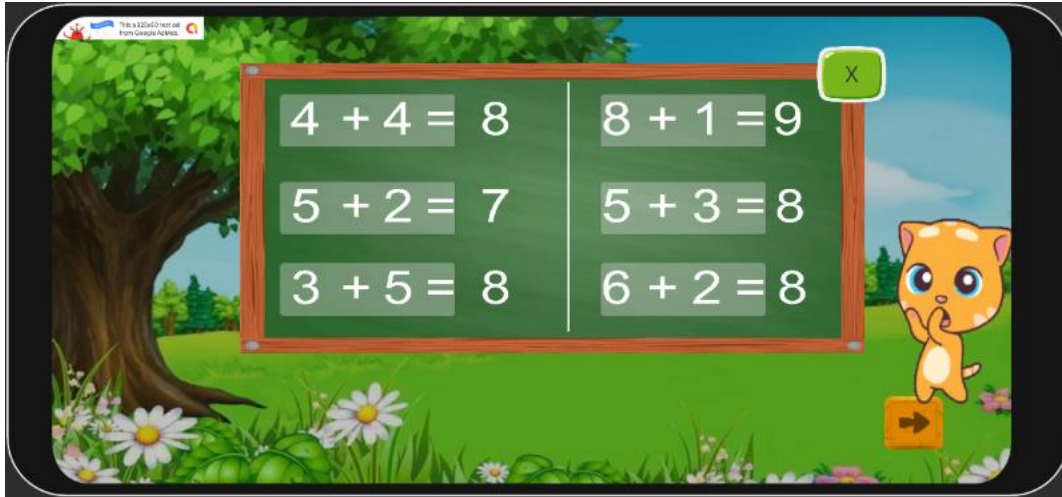
This screen consists of beginner level addition practice puzzles where the kid who has beginner or basic knowledge on addition operations can solve the puzzles to enhance their math skills.



**Figure 07:** Addition Practice Beginner Level Screen for Kid Math Game with Cat

### 3.1.8 Addition Practice Intermediate Level Screen:

This screen consists of intermediate level addition practice puzzles where a kid who has intermediate knowledge of addition operations can solve addition puzzles to enhance their math skills.

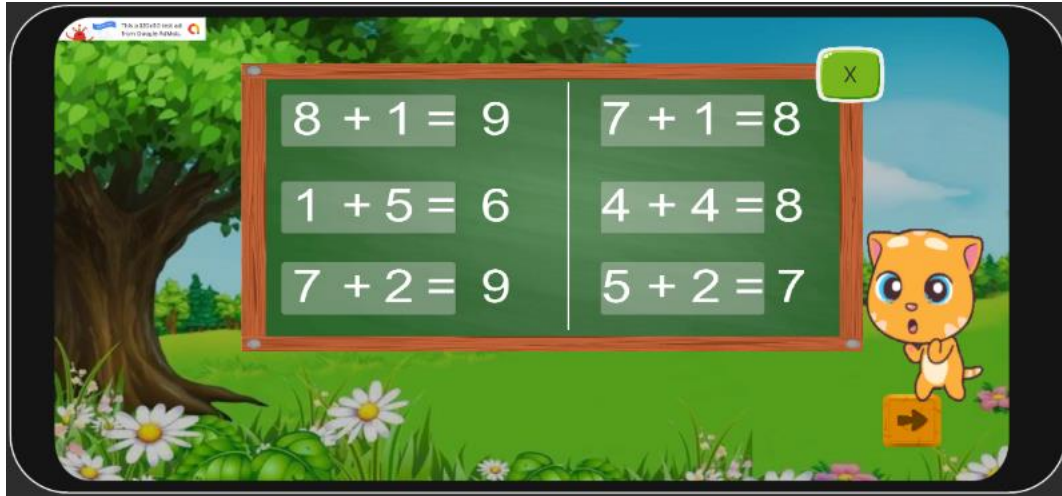


**Figure 08:** Addition Practice Intermediate Level Screen for Kid Math Game with Cat

### 3.1.9 Addition Practice Advanced Level Screen:

This Screen consists of advanced level addition practice puzzles where a kid who has advanced knowledge of addition operations can solve addition puzzles to enhance their math skills.

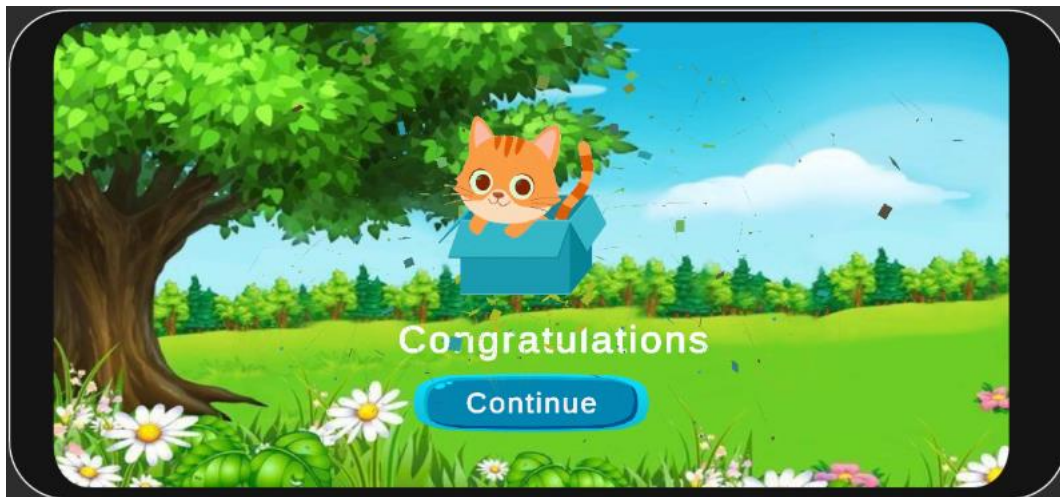




**Figure 09:** Addition Practice Advance Level Screen for Kid Math Game with Cat

### 3.1.10 Congratulations Screen:

Whenever an appropriate task is completed, the congratulations screen will appear and prompt the user to continue the game.



**Figure 10:** Congratulations Screen for Kid Math Game with Cat

## **3.2 Hardware Interfaces**

The minimum hardware requirements of Math Kid game are a 500-Megahertz CPU and 1024 megabytes of RAM. Also, a compatible graphics card is required as Math Kid game uses an OpenGL 2D engine to speed up graph visualization. A system with these specifications can handle a network of approximately 1000 edges and nodes.

## **3.3 Software Interfaces**

### **3.1.1 Unity**

The application will use the Unity game engine for the user interface of the Connectome application. While the application will not contain any game components, the framework makes the interface with the Math kid Game easy. A 32-bit personal version of the unity game engine is used for the project. None of the components of the professional version should be needed for the application. Unity version 5.5 is used for the development of the project.

### **3.1.2 Visual Studio**

A streamlined code editor, Visual Studio Code supports development activities like task execution, debugging, version management. It seeks to offer only the tools a developer needs for a brief code-build-debug cycle and leaves more complicated processes to IDEs with more features, like Visual Studio IDE.

### **3.1.3 C# and ASP.NET**

The application will run on version 6.0 of the C# language. This is the most recent version of the C# programming language. The C# programming language is used for all the backend programming for the application. Visual Studio is used to compile all the code that is developed in C#. Version 3.5 of the .NET framework is used for development in C# because of limitations imposed by the Unity game engine.

## **3.4 Communications Interfaces**

Since there is no network activity needed, there are no requirements of any interface.

## **4 System Features**

### **4.1 Title/Menu Screen**

#### **4.1.1 Description and Priority**

The title screen is the screen the player will see first upon entering the game. Through this interface, the player can choose to start the game, resume the game, or adjust the options. Since the home / menu screen is the “hub” for all activities in the project, it must be included.

#### **4.1.2 Stimulus/Response Sequences**

**Step 1:** The player will launch the game from their portable device.

**Step 2:** The start screen loads and appears prompting the player to start with the game.

**Step 3:** The player presses the button, triggering different functionalities of the game, such as the Adding Puzzle, Adding Quiz, Adding Fun and Adding Practice games.

#### **4.1.3 Functional Requirements:**

**RFQ-1:** The home / menu screen must load and appear every time the game is launched.

**RFQ-2:** If the player quits the game during any stage of a level, they must be returned to the main screen.

**RFQ-3:** If the player presses the exit button, the game will end and return the player to the mobile device’s regular interface.

**RFQ-4:** If the player completes the game, it will take them to the congratulations screen and then return the player to the menu screen.

### **4.2 Generate random numbers and validate the answers**

#### **4.2.1 Description and Priority:**

The player, after going to the menu screen, navigates to the Adding Puzzle screen, where addition questions are displayed, and the player must choose the correct

answer and drag it to the correct location. If the response is correct, an animation of a cat will appear to show that it is correct; otherwise, it prompts the player to let them know that it is an incorrect response. If the answer is correct, it moves on to the next level.

#### **4.2.2 Stimulus/Response Sequences:**

**Step 1:** The player navigates to the Adding Puzzle screen from the main screen.

**Step 2:** In the Adding Puzzle screen we can see a question board and some random numbers along with accurate number for response.

**Step 3:** An addition question will be generated with random numbers and displayed on the question board.

**Step 4:** In the answer board random numbers are generated, including the correct answer.

**Step 5:** Once the answer is selected from the board, a cat animation will pop up reacting to whether the answer was correct or incorrect.

#### **4.2.3 Functional Requirements:**

**RFQ-1:** The cat should appear on the screen and indicate the answer is correct if the player selects the correct answer from the random numbers.

**RFQ-2:** When the player chooses the correct answer and drops it in the right place, a cat animation needs to play.

**RFQ-3:** When the player chooses the wrong response, it should indicate the answer is incorrect.

**RFQ-4:** By clicking on the home button displayed at the top right corner, the player can go back to the menu screen and start the game again.

### **4.3 Adding Quiz Screen**

#### **4.3.1 Description and Priority:**

The player, after going to the menu screen, navigates to the adding quiz screen,



where addition questions are displayed in the question board. Here, the player must choose the correct answer and drag it to the correct location. If the response is correct, an animation of a cat will appear to indicate that it is correct; otherwise, it prompts the player to let them know that it is an incorrect response. If the answer is correct, it moves to the next level.

#### **4.3.2 Stimulus/Response Sequences:**

**Step 1:** The player navigates to the Adding Quiz screen from the main screen.

**Step 2:** In the Adding Quiz screen we can see a question board and some random numbers along with the correct answer.

**Step 3:** An addition question will be generated with random numbers and displayed on the question board.

**Step 4:** In the answer board random numbers are generated along with the correct answer.

**Step 5:** Once the answer is selected from the board, it will either popup a cat animation if it is correct or display as incorrect if it is wrong.

#### **4.3.3 Functional Requirements:**

**RFQ-1:** The cat should appear and indicate the answer is correct on the screen if the player selects the correct answer from the random numbers.

**RFQ-2:** When the player chooses the correct answer and drops it in the right place, a cat animation needs to play.

**RFQ-3:** When the player chooses the wrong response, it should indicate that the answer is incorrect.

**RFQ-4:** By clicking on the home button displayed at the top right corner, the player can go back to the menu screen and start the game again.

## **4.4 Adding Puzzle Screen**

### **4.4.1 Description and Priority:**

The player, after going to the menu screen, navigates to the Adding Puzzle screen by clicking on the Adding Puzzle button where a puzzle with numbers, addition operators and equal to operators is displayed in the question board. Here, the player must choose an answer and drag the appropriate number and operator to the correct place. If the response is correct, an animation of a cat will appear to show that it is correct; otherwise, it indicates to the user that it is an incorrect answer. If the answer is correct, it moves to the next level.

### **4.4.2 Stimulus/Response Sequences:**

**Step 1:** The player navigates to the Adding Puzzle screen from the main screen.

**Step 2:** In the Adding Puzzle screen we can see a question board and some random numbers along with addition and equal to operators to complete the puzzle.

**Step 3:** An addition puzzle will be generated with numbers and addition/equal to operators and displayed on the question board.

**Step 4:** In the answer board random numbers are generated along with + and = operators to drag them to the correct place to form a perfect addition operation.

**Step 5:** Once the answer is selected and dropped in an appropriate place from the board, it will indicate with a cat animation whether the answer is correct.

### **4.4.3 Functional Requirements:**

**RFQ-1:** The cat should appear on the screen and indicate the answer is correct if the player selects and drops the correct number or operator from the generated puzzle.

**RFQ-2:** When the player chooses the correct answer and drops it in the right place, a cat animation needs to play.

**RFQ-3:** When the player chooses the wrong response, it should indicate that the answer is incorrect.

**RFQ-4:** By clicking on the home button displayed at the top right corner, the player can go back to the menu screen and start the game again.

## **4.5 Addition Practice Screen**

### **4.5.1 Description and Priority:**

The player, after going to the menu screen, navigates to the Addition Practice screen by clicking on the Addition Practice button. Various puzzles with numbers, addition operators and equal to operators are displayed in the question board and here the player must choose an answer and drag the appropriate number and operator to the correct place. If the response is correct, an animation of a cat will appear to show that it is accurate; otherwise, it indicates to the user that it is an incorrect response. If the answer is accurate, it moves on to the next level.

### **4.5.2 Stimulus/Response Sequences:**

**Step 1:** The player navigates to the Addition Practice screen from the main screen.

**Step 2:** In the Addition Practice screen we can see a question board and some random numbers along with addition and equal to operators for completing the puzzle.

**Step 3:** An addition puzzle will be generated with numbers and addition/equal to operators and displayed on the question board.

**Step 4:** In the answer board random numbers are generated along with + and = operators that need to be dragged to the correct location to form a perfect addition operation.

**Step 5:** Once the answer is selected and dropped in an appropriate place from the

board, a cat animation is displayed to indicate whether the answer was correct or incorrect.

#### **4.5.3 Functional Requirements:**

**RFQ-1:** The cat should appear on the screen and indicate that the answer is correct if the player selects and drops the correct answer or operator from the generated puzzle.

**RFQ-2:** When the player chooses the correct answer and drops it in the right place, a cat animation needs to be played.

**RFQ-3:** When the player chooses the wrong response, it should indicate that the answer was incorrect.

**RFQ-4:** By clicking on the home button displayed at the top right corner, the player can go back to the menu screen and start the game again.

## **4.6 Adjust Questions based on Difficulty level.**

### **4.6.1 Description and Priority:**

Once the player enters the Addition Practice Screen, he/she will be navigated to a difficulty level screen with a popup screen with three buttons "Beginner," "Intermediate," and "Advanced". The player can use these buttons to select which level of the game he/she wants to enter. Every user can play any level based on their choice.

### **4.6.2 Stimulus/Response Sequences:**

**Step 1:** The player navigates to the next screen from the Addition Practice Screen.

**Step 2:** In this next screen the player will move to the intermediate screen.

**Step 3:** Following this a popup with three buttons will appear labelled "Beginner", "Intermediate", and "Advanced", and the player can select his/her desired

screen.

**Step 4:** The player presses any of the three buttons, triggering its respective

function to move to a new game.

#### **4.6.3 Functional Requirements:**

**RFQ-1:** The questions will range according to the levels chosen by the player.

## **5 Other Nonfunctional Requirements**

### **5.1 Performance Requirements**

Considering the capability of modern smartphones and android operating systems, performance should not be a problem. Phones with lesser hardware, however, can experience certain issues and may operate slowly. No matter the hardware, the game is designed to provide a fun experience on all android phones. The game's functioning will be simple enough and easy to understand. The graphics will not be very complex, in order to avoid slowing down the system.

### **5.2 Safety Requirement**

The Kid Math Game with Cat will not interfere with or harm any of the other applications loaded on the player's phone. Additionally, the game will not make the player's phone overheat, protecting the phone's internal parts from harm. To avoid potential harm to the player, the Kid Math Game with Cat should not be played when the player's attention is split between several things.

### **5.3 Security Requirements**

Since the Kid Math Game with Cat concept won't request any personal information from the player, it cannot compromise such information. This math game can be played without the need for player authentication. Simply downloading the application will allow the gamer to begin playing the addition games. Therefore, this game can be played by anybody who has access to a player's phone. Any unauthorized player will be able to play math game if they get their hands on the original player's phone. It is the player's obligation to make sure that no other player or individual can access their phone.

## **5.4 Software Quality Attributes**

Kid Math Game concept with Cat responds to the player's directions promptly to guarantee dependability and accuracy. The results of a player's action in the game should be visible to other players in milliseconds, not ten seconds afterwards. The Kid Math Game concept with cat is flexible and adaptable, and it will automatically save the player's progress once each level is finished. In this way, if the player's phone runs out of power while they are playing, they can resume the game at a logical beginning point.

The player will find the graphical user interface to be quite easy to use. The Kid Math Game concept with early levels also gradually introduces all the different commands that the player has access to, as well as any other game features that the player might need to be familiar with to advance. These introductions and pointers won't be provided in more challenging and raised levels because a player will already be familiar with all the resources required to finish the level. This strategy will guarantee that the user experiences a hard game while also gradually learning all the game concepts. Our game does not favor either ease of use or ease of learning more than the other. Any player should be able to pick up and start playing the Kid Math Game idea with Cat right away without spending too much time getting used to the controls. The player won't be overwhelmed by needing to utilize every control at once in the early levels because of the controls' simplicity and intuitiveness. As a player advances through the game and completes each level, the gameplay will gradually introduce each command, making it easier for them to learn and use. The player should be fully capable of using the command introduced in every given level, as well as other commands introduced in earlier levels, by the time that level is through.

## **5.5 Business Rules**

N/A

# **6 Other Requirements**

## **6.1 Storage solution**

We looked at two potential game data storage systems. Using a file or an integrated database were the options. We chose to utilize Azure App Service as our back-end technology since it offers excellent support for databases and has user-friendly interfaces for getting and storing data. With the class Azure App Service, it provides a simple method to create, edit, and manage Azure App Service. Even if the application were to end abruptly, a database and periodic game data saving will guarantee that no data will be lost.

## **6.2 Design**

The database is created, and the game content is added when the game is installed. The game retrieves the current game state at the beginning, which includes the current task and the path. If there is no current game state, the current task is set to the first task in the newly retrieved route from the database.

## **6.3 Task**

The user's first user interface screen is a welcome screen with a text greeting and a button for moving on to the next activity. The next job in the route is fetched from the database when the user presses the next task button. When the task is finished, the application automatically retrieves the following one. The welcome page will appear once more after the user departs the game, and the route and task are then pulled from the database. The task that the user was working on when the application was terminated will be loaded if the user clicks the next task button.

## **6.4 Cohesiveness**

High cohesion promotes reusability, reliability, and robustness. Each module, class, and file in the entire project is somehow dependent upon one another because it is a gaming project, so it cannot be finished alone. We would need access to both the real character's location as well as the locations of any colliding objects to identify the primary character's collision with any limits, walls, or other obstacles. Because each item in a game must access the information or fields of the other objects for the game's physics to function, practically all game code is cohesive, making this project's cohesiveness higher than that of typical software projects.

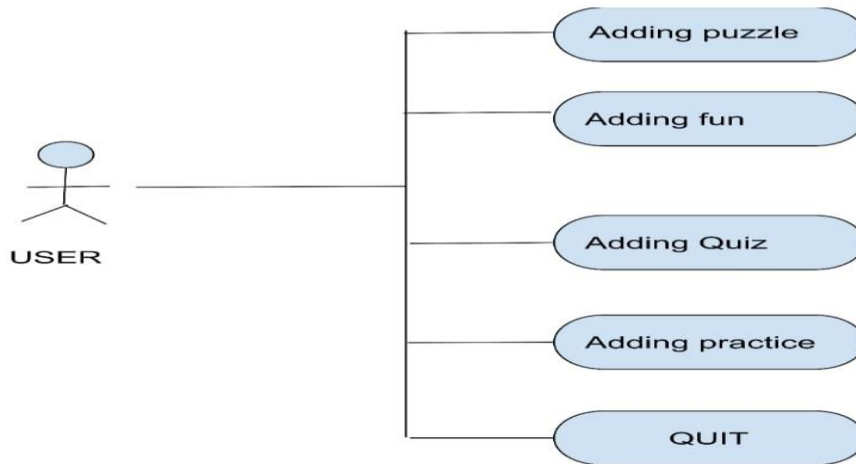
## **Appendix A: Glossary**

N/A

## Appendix B: Analysis Models

### A. UML Use Case Diagram

Use case diagrams are the diagrams which are used to show the relation between actors and their interactions. A use case diagram shows various use cases and different types of users the system has.



**Figure 11:** UML Use Case Diagram 1



Implementation of Addition Puzzle Screen

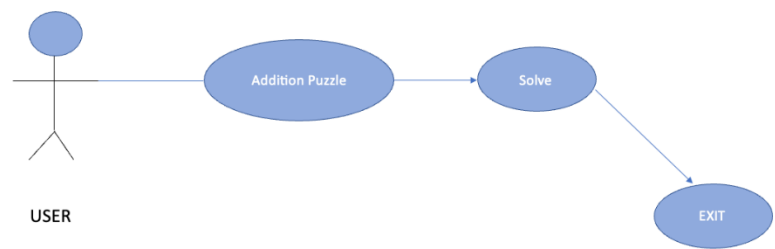


Figure 12: UML Use Case Diagram 2

Implementation of Addition Fun screen

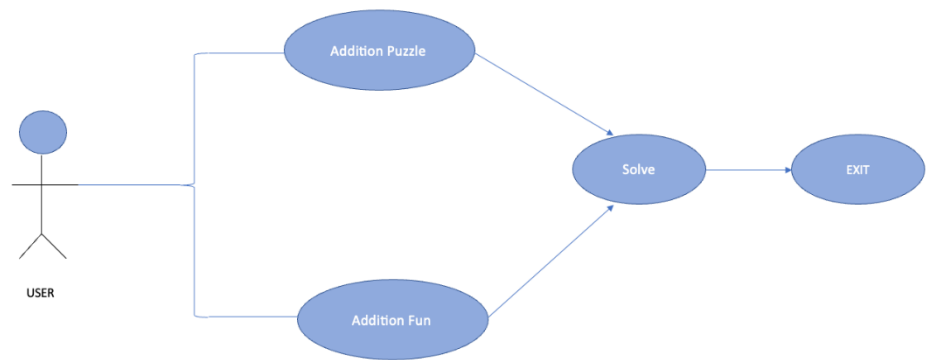
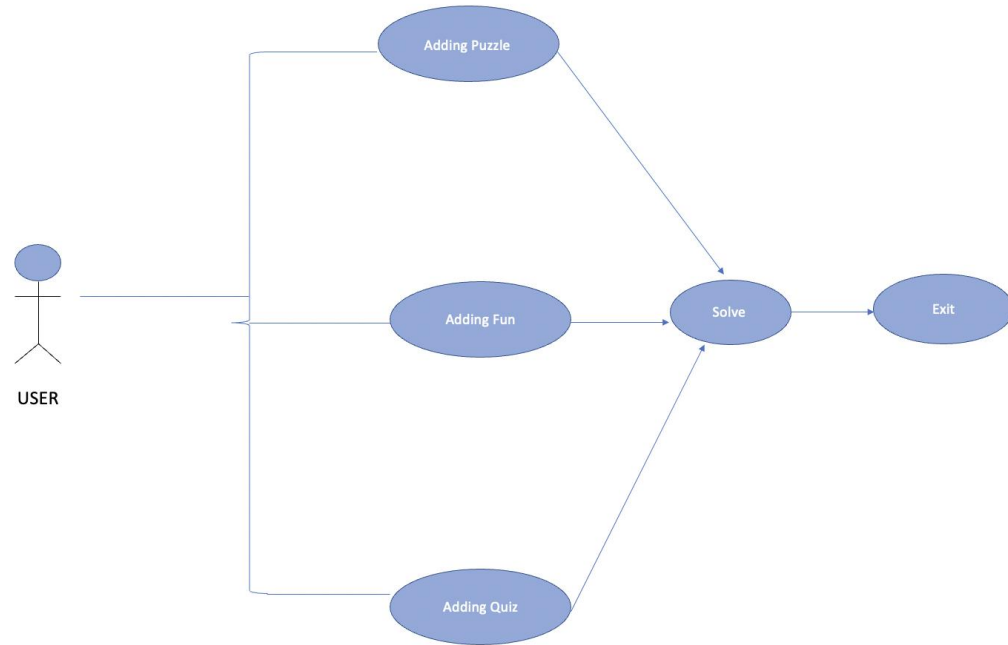


Figure 13: UML Use Case Diagram 3

Implementation of Addition Quiz Screen



**Figure 14:** UML Use Case Diagram 4

## Implementation of Addition practice screen

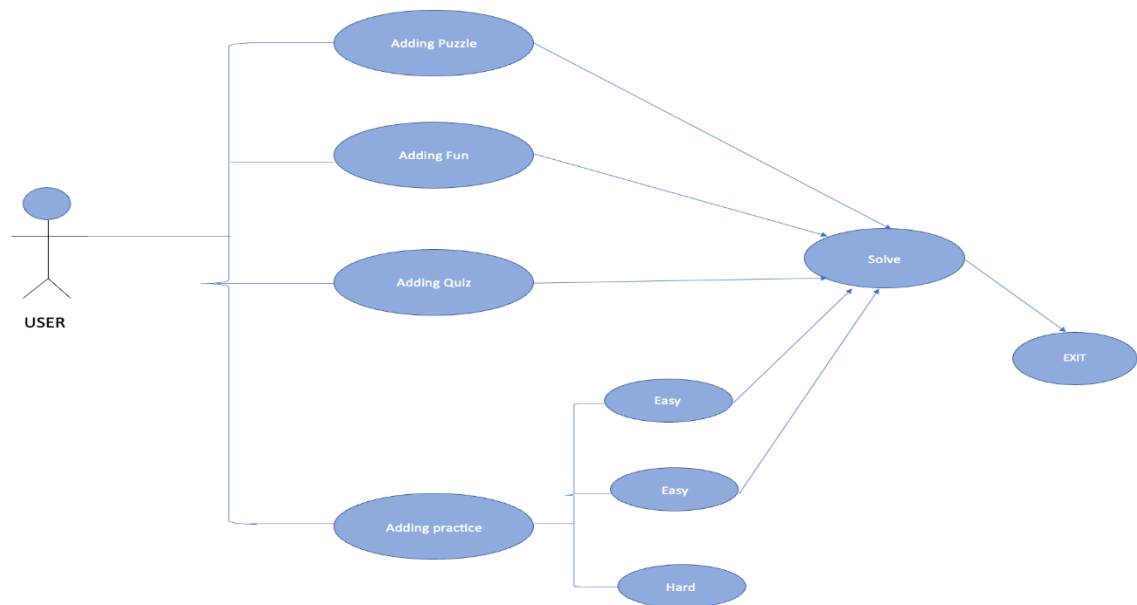
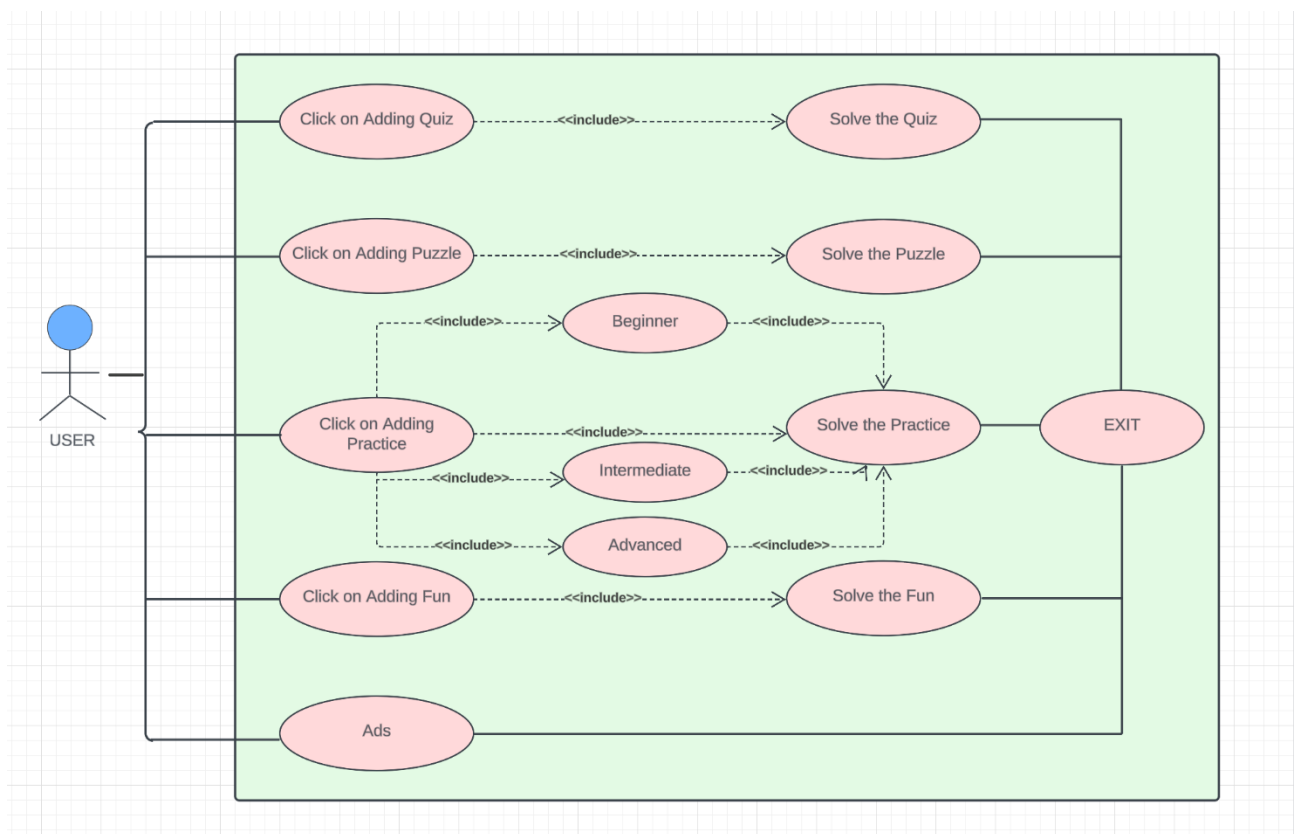
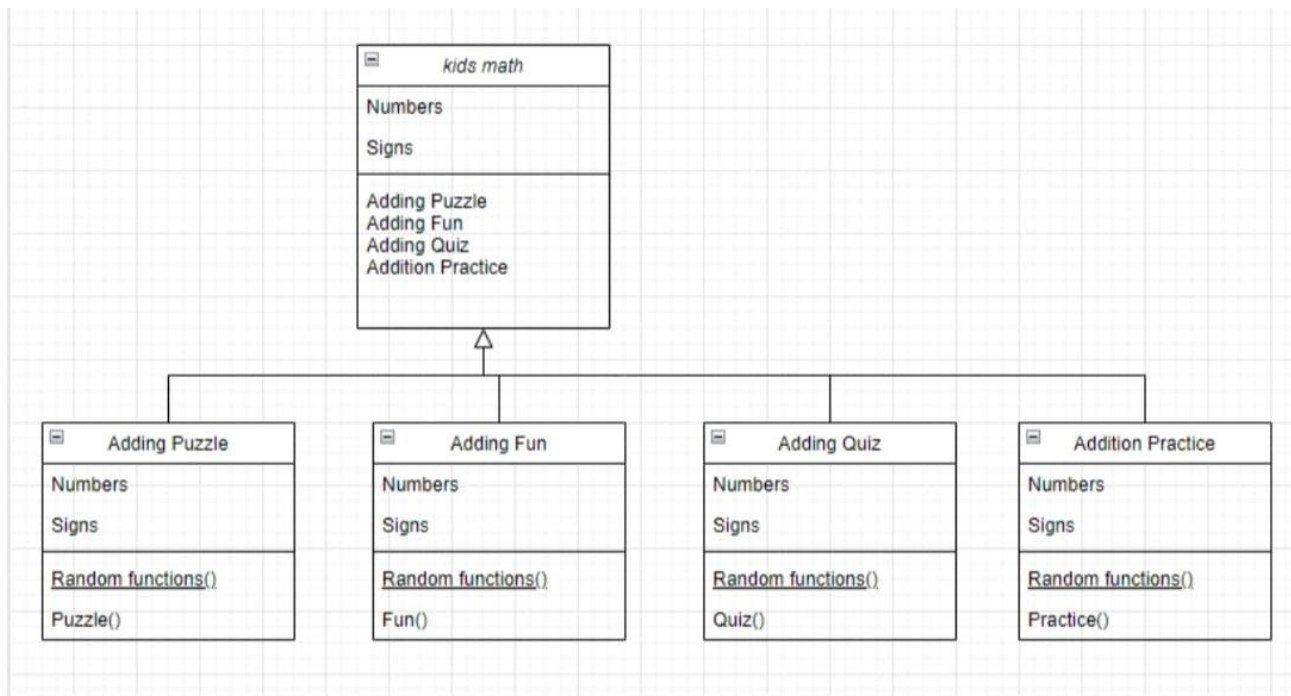


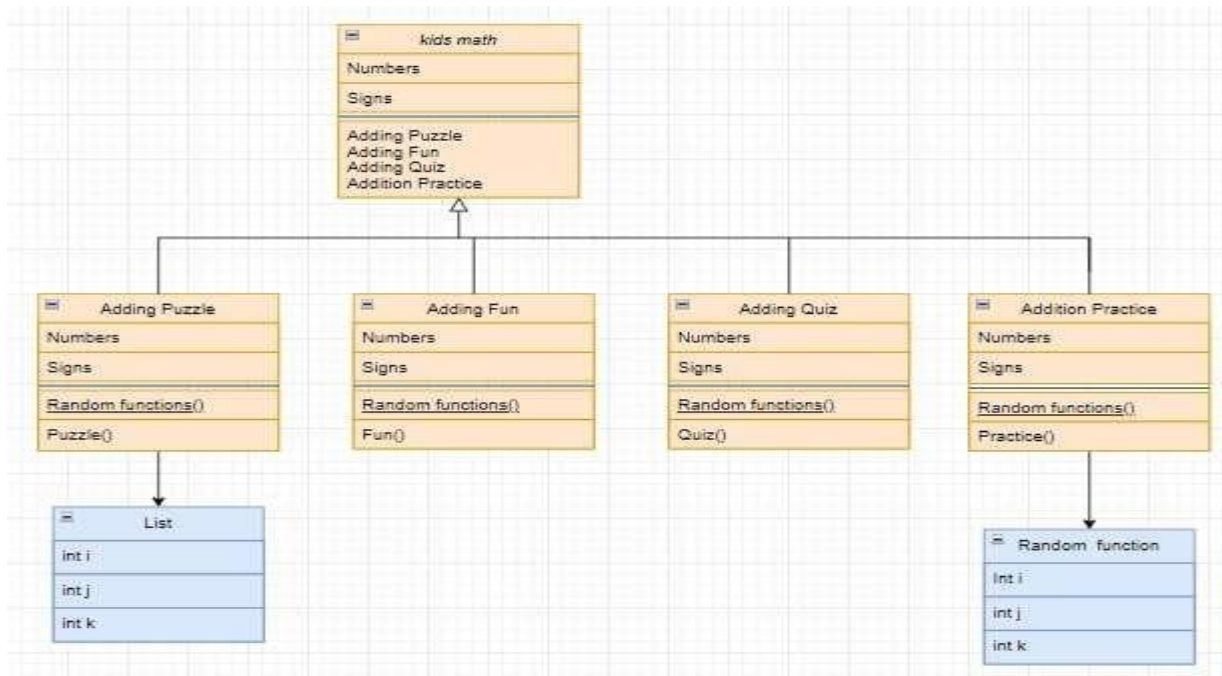
Figure 15: UML Use Case Diagram 5



**Figure 16:** UML Use Case Diagram 6**B. UML Class Diagram:**

UML diagram describes a system by visualizing the different types of objects within a system and the kinds of static relationships that exist among them. It also illustrates the method operations and attributes of the classes.

**Figure 17:** UML Class Diagram 1

**Figure 18:** UML Class Diagram 2

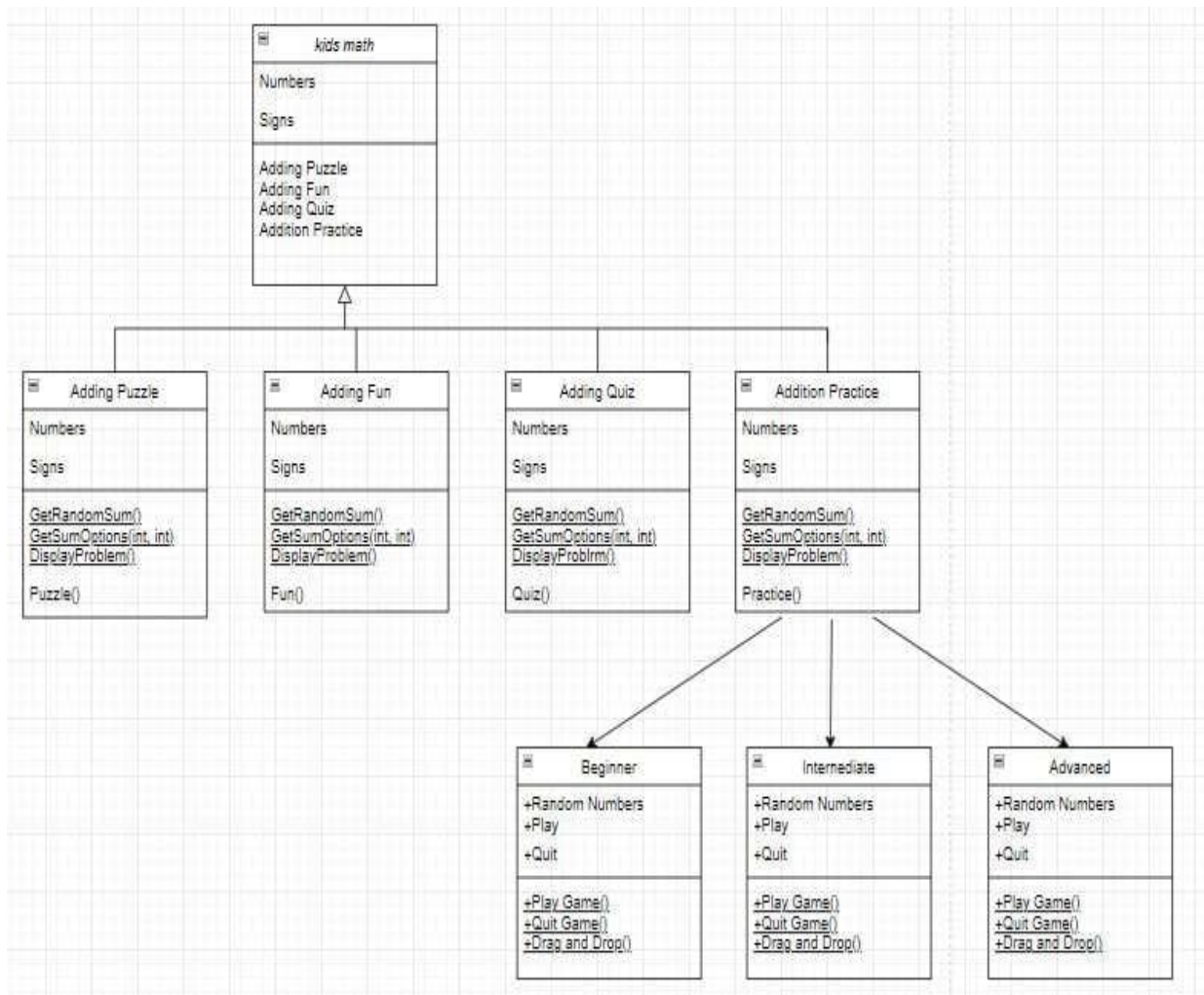


Figure 19: UML Class Diagram 3

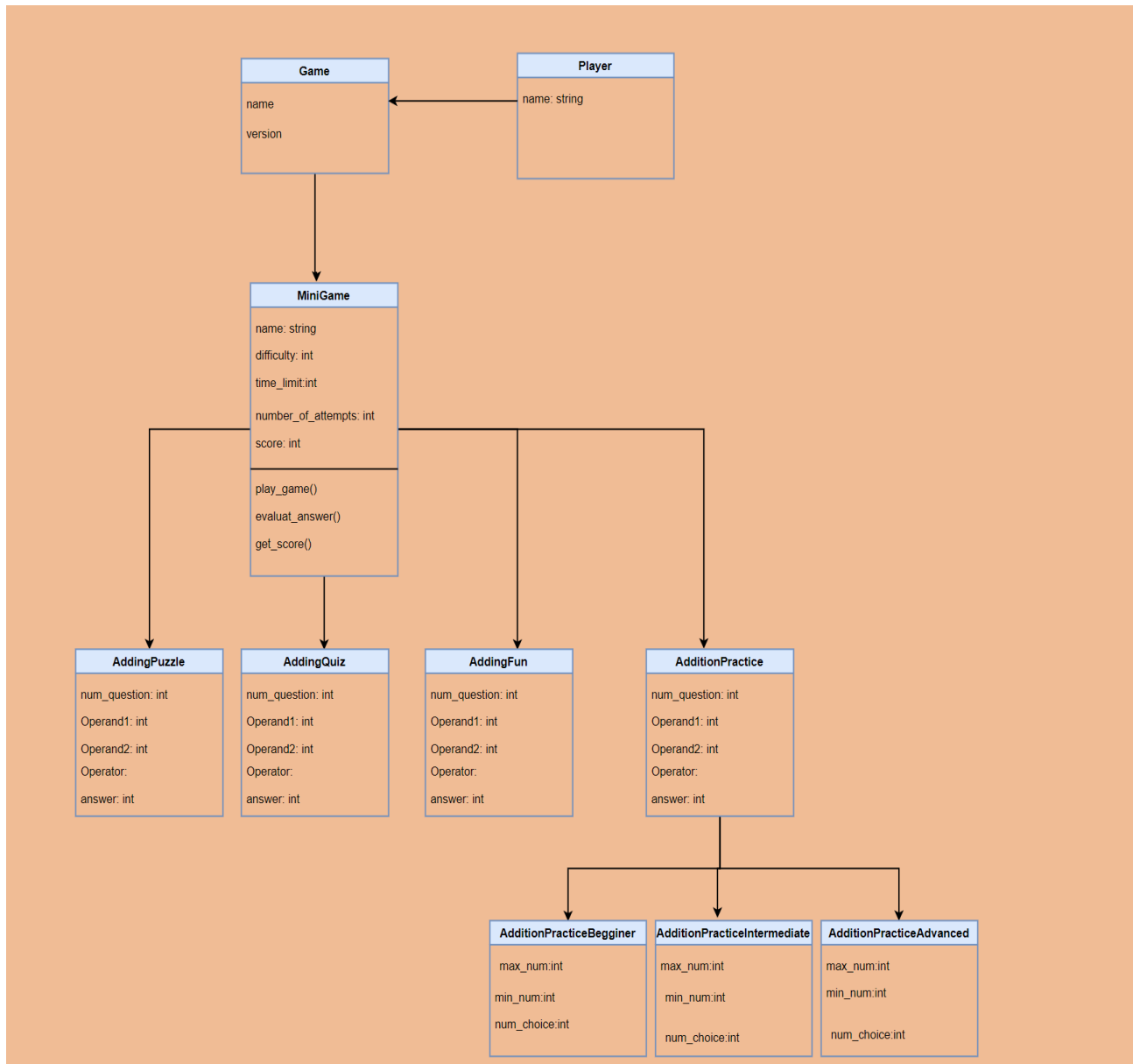
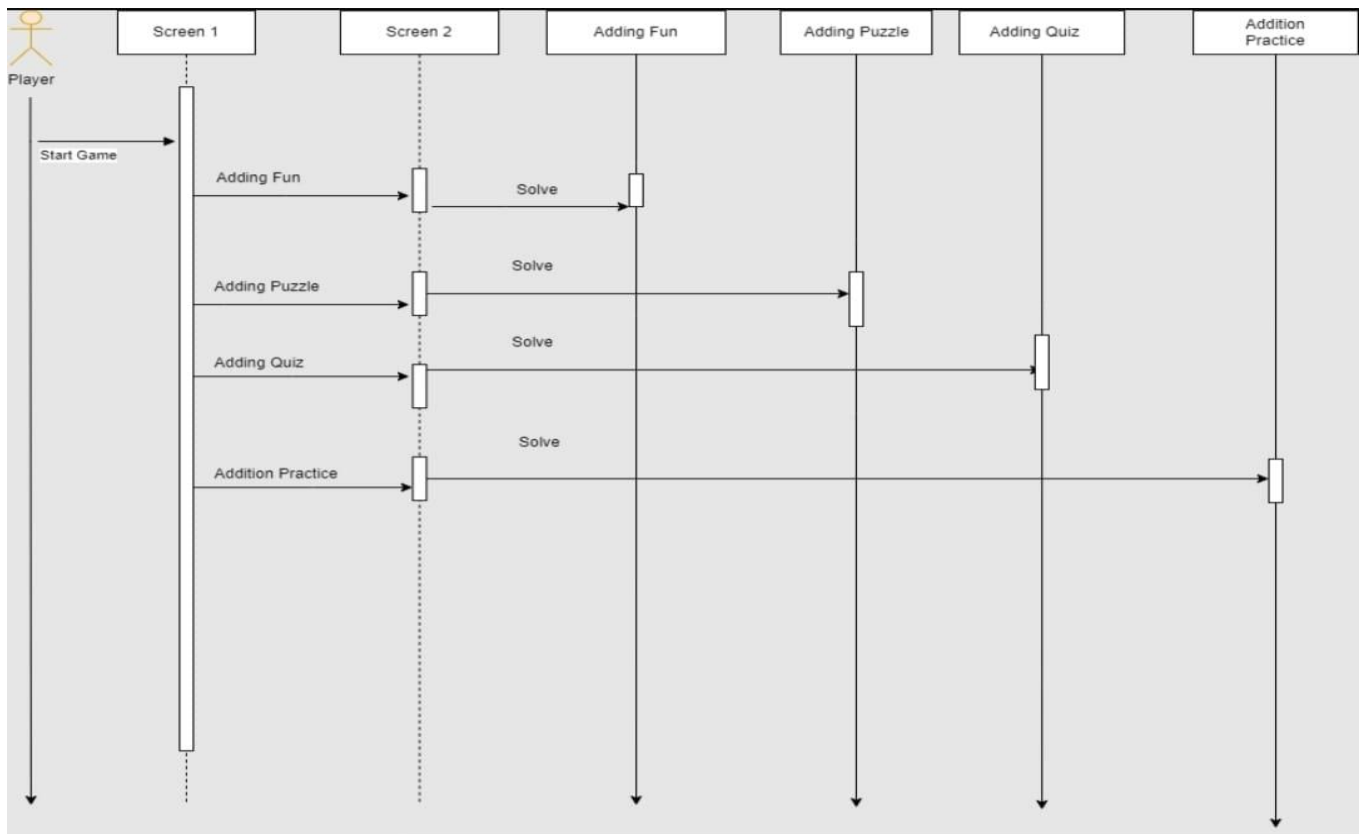


Figure 20: UML Class Diagram 4

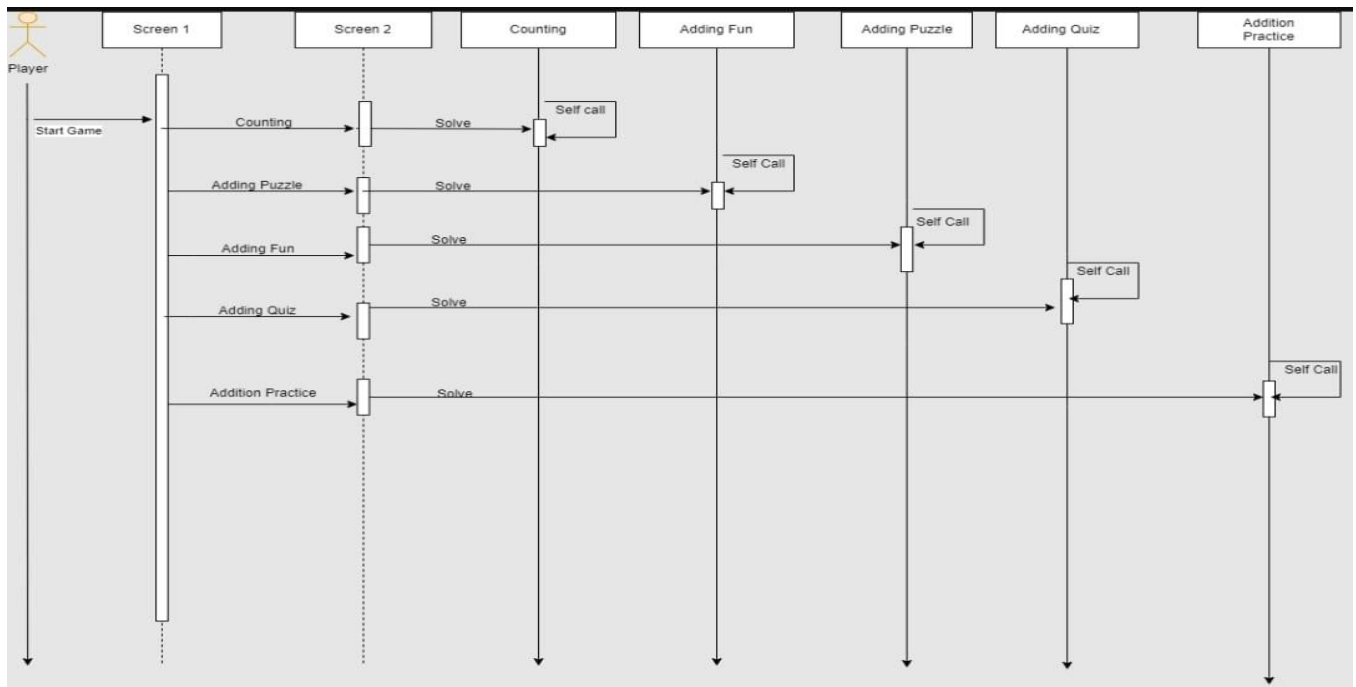
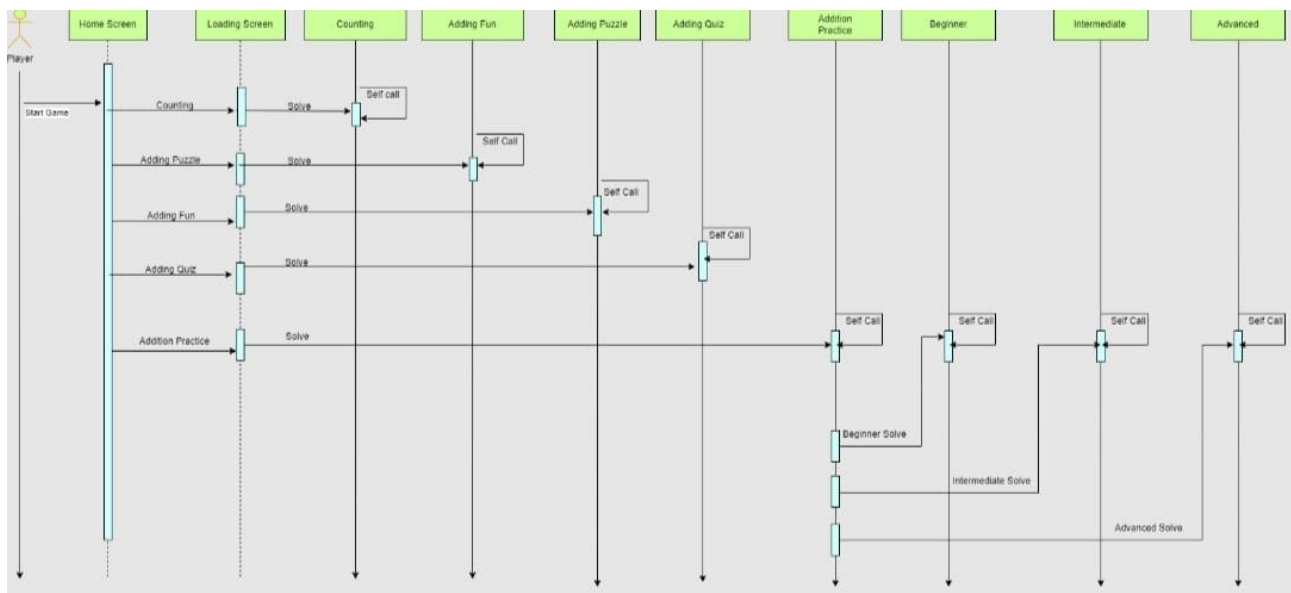
## C. UML Sequence Diagram

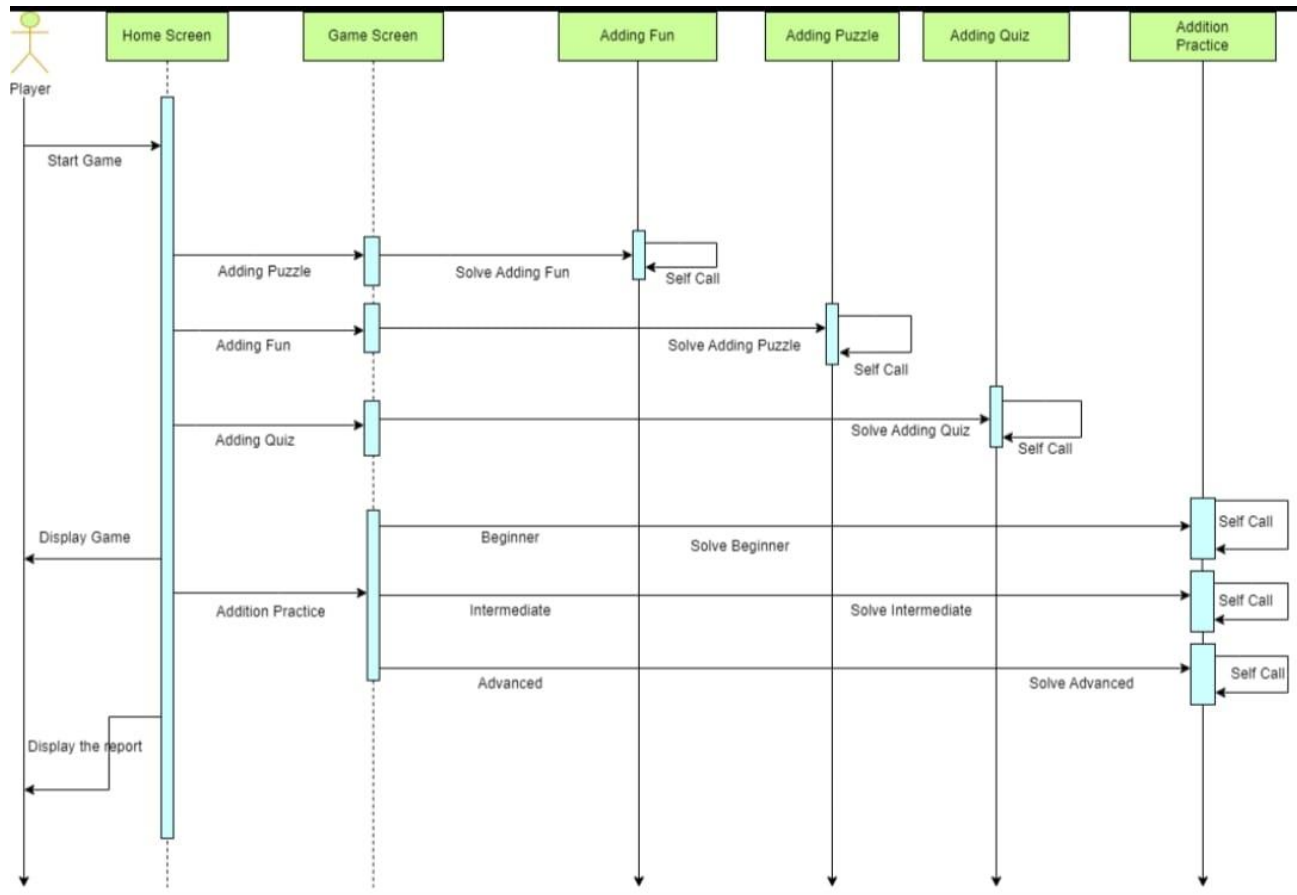
UML Sequence diagrams illustrate the sequences of messages between objects in an interaction. A sequence diagram consists of a group of objects that are represented by lifelines, and the messages that they exchange over time during interaction.



**Figure 21:** UML Sequence Diagram 1



**Figure 22: UML Sequence Diagram 2**

**Figure 23:** UML Sequence Diagram 3**Figure 24:** UML Sequence Diagram 4

## Appendix C: To Be Determined List

N/A