

1) SRS Progress Sheet

a) Jaswanthi Jagarlamudi

03-13-23	<ol style="list-style-type: none">1.Updated System Features 2<ul style="list-style-type: none">● Generate random numbers and validate the answers<ol style="list-style-type: none">1. Description and Priorities2. Stimulus/Response3. Functional Requirements2.Updated UML class, sequence diagrams
03-06-23	<ol style="list-style-type: none">1.Updated System Features 1<ul style="list-style-type: none">● Title / Menu Screen<ol style="list-style-type: none">1. Description and Priorities2. Stimulus/Response3. Functional Requirements2.Updated UML class, sequence diagrams
02-27-23	<ol style="list-style-type: none">1.Updated Hardware interfaces required2.Updated Performance requirements3.Updated class, sequence diagrams
02-20-23	<ol style="list-style-type: none">1.Updated Overall Description2.Added Appendix B: Analysis models<ol style="list-style-type: none">A. UML Use case diagramB. UML Class diagramC. UML Sequence diagram3.Added Initial Screen, Home Screen, Puzzle Screen, Store Screen to the user interface
02-13-23	<ol style="list-style-type: none">1.Updated Introduction<ul style="list-style-type: none">● Purpose● Intended audience and reading suggestions● Product scope2.Updated external interface requirements<ul style="list-style-type: none">● User Interfaces

	<ul style="list-style-type: none">• Software Interfaces
02-06-23	<ol style="list-style-type: none">1.Created initial version of document2.Updated SRS with Table of Contents

Software Requirements Specification

for

Kid Math Game with cat

Version 1.5

Prepared by Team 1 of Software Engineering Class

California State University, San Bernardino

03-13-23

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

Table of Contents

Table of Contents

Revision History

1. Introduction

- 1.1 Purpose
- 1.2 Document Conventions
- 1.3 Intended Audience and Reading Suggestions
- 1.4 Product Scope
- 1.5 References

2. Overall Description

- 2.1 Product Perspective
- 2.2 Product Functions
- 2.3 User Classes and Characteristics
- 2.4 Operating Environment
- 2.5 Design and Implementation Constraints
- 2.6 User Documentation
- 2.7 Assumptions and Dependencies

3. External Interface Requirements

- 3.1 User Interfaces

Error!	Bookmark	not	defined.	1.1	Initial	Screen	4
4							
				3.1.2	Main Screen		4
				3.1.3	Adding Puzzle Screen		4
				3.1.4	Adding Fun Screen		5
				3.1.5	Adding Quiz Screen		5
				3.1.6	Addition Practice Screen		5
				3.1.7	Store Screen		5
				3.2	Hardware Interfaces		5
				3.3	Software Interfaces		5
				3.4	Communications Interfaces		5
				4.	System Features		5
				4.1	Title/Menu Screen		6
				4.1.1	Description and Priorities		6
				4.1.2	Stimulus/Response		6
				4.1.3	Functional Requirements		7
				4.2	Generate random numbers and validate the answers		7
				4.2.1	Description and Priority		7
				4.2.2	Stimulus/Response		8
				4.2.3	Functional Requirements		8
				5.	Other Nonfunctional Requirements		8
				5.1	Performance Requirements		9
				5.2	Safety Requirements		9
				5.3	Security Requirements		9
				5.4	Software Quality Attributes		9
				5.5	Business Rules		9

6. Other Requirements	10
Appendix A: Glossary	11
Appendix B: Analysis Models	12
A. UML Use Case Diagram	13
B. UML Class Diagram	14
C. UML Sequence Diagram	15
Appendix C: To Be Determined List	

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
Jaswanthi Jagarlamudi	02-13-23	1.Updated Introduction <ul style="list-style-type: none"> Purpose Intended Audience and Reading Suggestions Product Scope 2.Updated External Interface Requirements <ul style="list-style-type: none"> User Interfaces Software Interfaces 2) Unity 3) Visual Studio 4) C# and .NET 	1.1
Jaswanthi Jagarlamudi	02-20-23	1.Updated Overall Description <ul style="list-style-type: none"> Product Perspective Product Functions User classes and characteristics Operating Environment 2.Added Appendix B: Analysis Models <ul style="list-style-type: none"> 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 class,sequence diagrams	1.3
		1.Updated System Features 1 <ul style="list-style-type: none"> Title / Menu Screen 	

Jaswanthi Jagarlamudi	03-06-23	1. Description and Priorities 2. Stimulus/Response 3. Functional Requirements 2.Updated UML class,sequence diagrams	1.4
Jaswanthi Jagarlamudi	03-13-23	1.Updated System Features 2 • Generate random numbers and validate the answers 1. Description and Priorities 2. Stimulus/Response 3. Functional Requirements 2.Updated UML class,sequence diagrams	1.5

1. Introduction

a) 1.1 Purpose

The game application “Kid Math Game With Cat”, is an attractive, interactive and entertaining game. This game is created on the Unity Hub tool. Kids can be able to enhance their mathematical skills by playing this game with fun. It provides learning in counting, addition methods. This game provides a challenging and engaging educational experience for kids.

b) 1.2 Document Conventions

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

c) 1.3 Intended Audience and Reading Suggestions

The Software Requirement Specification provides the way for the user to verify that the game developed is coordinated with the original idea. To completely understand and review the project from initial stage to developed, the content was created in portions and can be therefore read such as described. 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 Part 5.

d) 1.4 Product Scope

The game “Kid Math Game with cat” aids in teaching kids the counting, addition concepts. The mixed math style makes the concept understandable to players of any skill level. Additionally this game offers a fun exercise to kids where kids can be able to learn the concept of addition with different items. Hints are also introduced to the users to make them understand the concept and solve the level in a better way.

e) 1.5 References

- Software Engineering: A Practitioner’s Approach
- IEEE Template: <https://goo.gl/nsUFwy> for System Requirements Specification documentation
- GitHub page:
- NuGet: <https://www.nuget.org/> package manager designed to enable developers to share reusable code.

i)

2. Overall Description

f) 2.1 Product Perspective

The game is an identical replica of the original kid math game app that is available in the playstore. The primary goal of this game is to make the mathematical concept of addition simpler and more enjoyable for kids to learn. This game provides us with the learning experience.

g) 2.2 Product Functions

Adapting kids to learn basic math addition calculations by implementing kid math game for the questions asked based on the difficulty level and thereby pointing towards answers. Implementing high quality cartoon animations wherever necessary for the effectiveness of kid's learning. The following is a summary of the major features implemented in the game. Based on the necessity of the game to function this is separated into categories.

- Title / Menu Screen : This is the application's initial viewable screen, which includes buttons for a new player, volume control
- Creating characters with names : Different characters are created with names to help us move from one phase to another phase.
- Adjust questions based on difficulty level : There will be a screen where the game redirects us to the next levels based on our performance from easy to medium and hard.
- Generate questions and validate the answers with reactions : Adding puzzles are generated here and the characters created helps us to validate the answers.
- Collecting stamps, stickers and toys : During the game different toys and stickers can be collected while moving to different levels.
- Kid learning progression track : For every game the kid plays, his/her data is recorded, a certificate is generated which tracks the development of his/her performance.

h) 2.3 User Classes and Characteristics

The control scheme is designed to be intuitive and the game play is fair enough to be understood for 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.

i) 2.4 Operating Environment

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

j) 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.

k) 2.6 User Documentation

While this app is the most simplest and the minimalistic one, end users will not face much difficulty while using the app as we have not implemented any complex operations on the app that uses huge compatibility. However, kids who are about to start their learning journey, we will provide them with a tutorial video on how to use the app.

ONLINE HELP:

- A step by step tutorial video on how to navigate on the application.

l) 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 released on market. As of now we have not noticed any dependencies and assumptions 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.

m) 3.1 User Interfaces

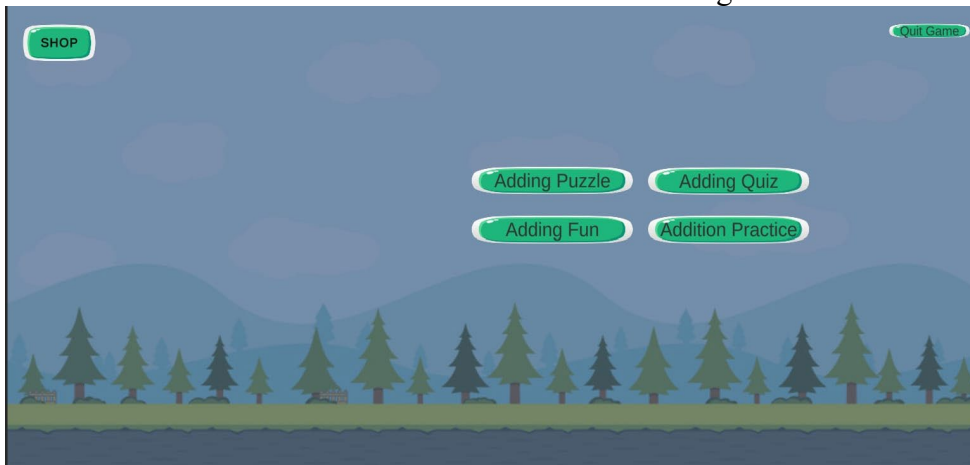
3.1.1 Initial Screen

This is the initial screen of the game and it redirects to the main screen.



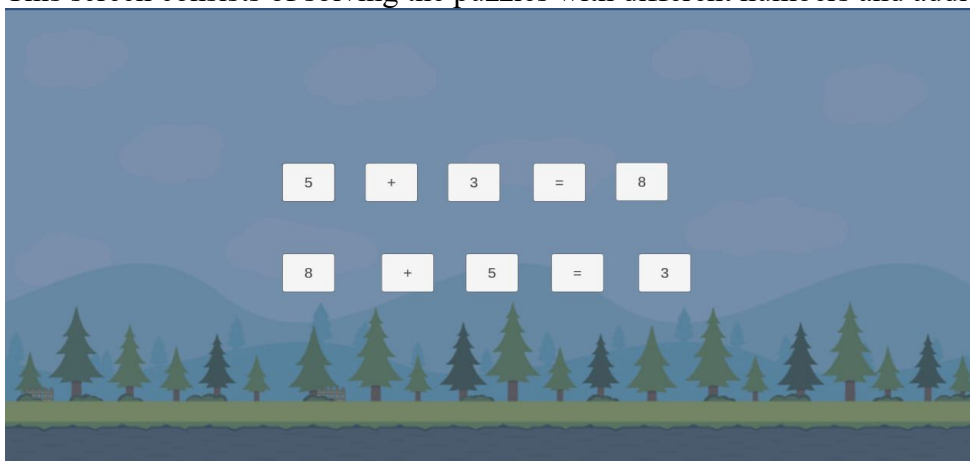
3.1.2 Main Screen

This screen consists of adding puzzles, adding fun, adding quiz, and adding practice buttons where the kid can be redirected to different scenarios according to his choice.



3.1.3 Adding puzzle Screen

This screen consists of solving the puzzles with different numbers and addition operators.



3.1.4 Adding Fun Screen

This screen consists of adding numbers using different entities.

3.1.5 Adding Quiz Screen

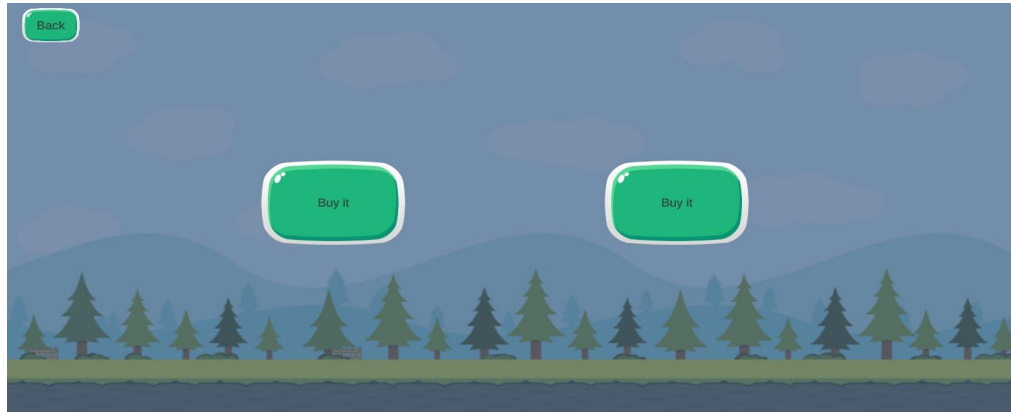
This screen consists of addition quiz with different numbers.

3.1.6 Addition Practice Screen

This screen consists of different levels beginner, intermediate and advanced where the kid can be redirected to solve the adding practice examples according to his level of progress.

3.1.7 Store Screen

Whenever clicked on the shop button it redirects to the store screen where the kid can buy a list of items.



n) 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.

o) 3.3 Software Interfaces

3.1.1 Unity

- For the project, a 32-bit personal version of the Unity game engine is utilized.
- The program should not require any of the features from the personal edition.
- The user interface of the Connectome application will be powered by Unity game engine. Although the application won't have any elements for gaming, the framework makes it simple to interact with the kid math game.

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

- C# is the only programming language used across the application's backend
- Visual Studio 2015 is used to compile all C# source code. C# development utilizes the NET framework version 3.5 due to limitations imposed by the Unity game engine.

p) 3.4 Communications Interfaces

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

4. System Features

q) 4.1 Title/Menu Screen

r) 4.1.1 Description and Priority

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

s) 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 to different functionalities of the game, whether it could be adding puzzle, adding quiz, adding fun and addition practice.

t) 4.1.3 Functional Requirements :

RFQ-1 : The title / 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 retired to the main screen.

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

RFQ-4 : If the player completes the game, it will take to the rewards page and then returns the player to the menu screen.

u) 4.2 Generate random numbers and validate the answers**v) 4.2.1 Description and Priority :**

The player after going to the menu screen navigates to the adding puzzle screen where questions regarding the addition practice are displayed in the questionnaire board and here the player must choose an answer and drag the appropriate answer to the correct place. If the response is accurate, an animation of a cat will appear to show that it is accurate; otherwise it prompts the player to let them know that it is a wrong response. If the answer is accurate it moves to the next level.

w) 4.2.2 Stimulus/Response Sequences :

Step 1 : The player navigates to the adding puzzle screen from the main menu screen.

Step 2 : In the adding puzzle screen we can see a questionnaire 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 questionnaire board.

Step 4 : In the answer board random numbers are generated to select 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 displays as incorrect if it is wrong.

x) 4.2.3 Functional Requirements :

RFQ-1 : The cat should appear and prompt the answer as 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 prompt as incorrect answer.

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.

y)

5. Other Nonfunctional Requirements

z) 5.1 Performance Requirements

Taking into account the capability of modern smartphones and android operating systems, performance shouldn't 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 functioning will be simple enough and easy to understand. The graphics won't be extremely complex to avoid slowing down the system.

aa) **Safety Requirement**

bb) **Security Requirements**

cc) **Software Quality Attributes**

dd) **Business Rules**

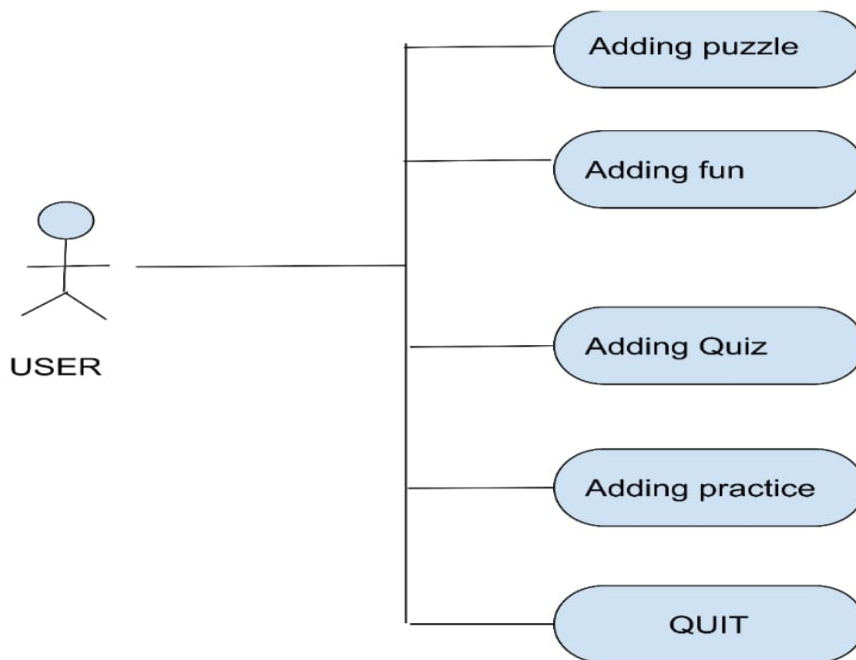
6. Other Requirements

Appendix A: Glossary

Appendix B: Analysis Models

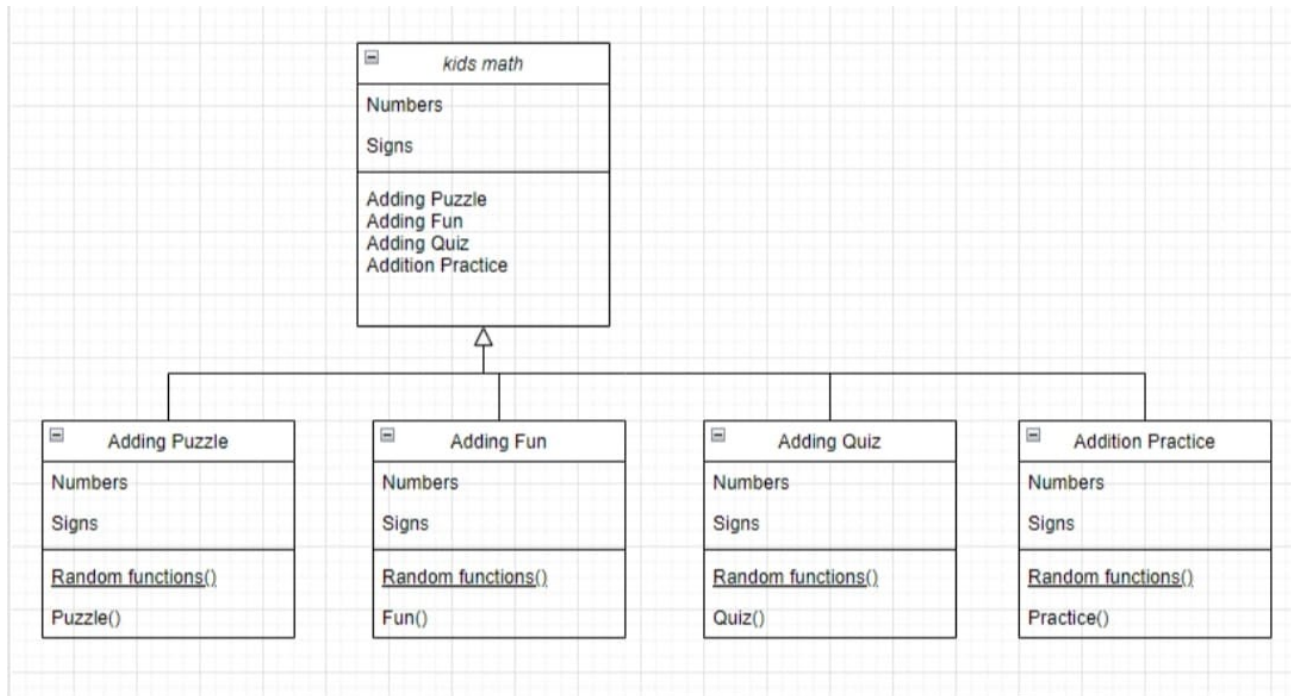
A. UML Use Case Diagram:

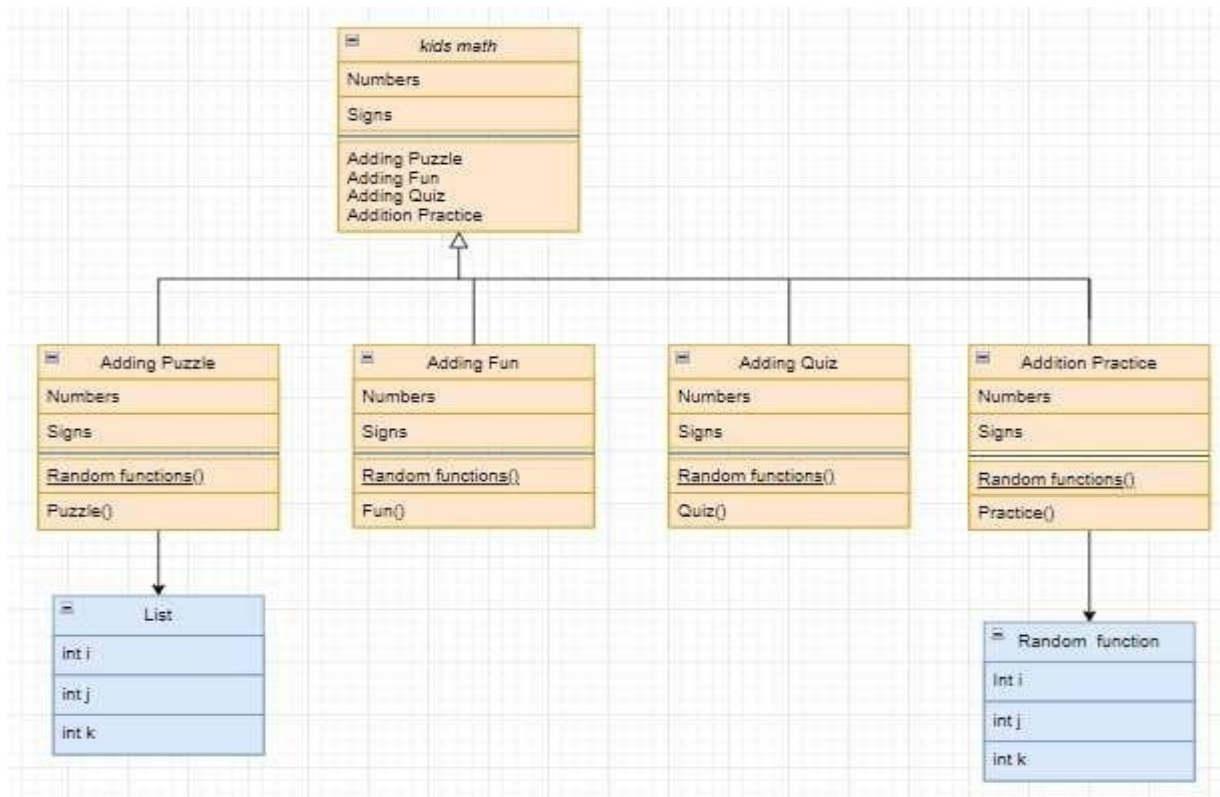
Use case diagrams are the diagrams which are used to show the relation between actors and their interactions. A use case diagram shows various use cases and different types of users the system has and will be accompanied by other types of diagrams as well.



B. UML Class Diagram:

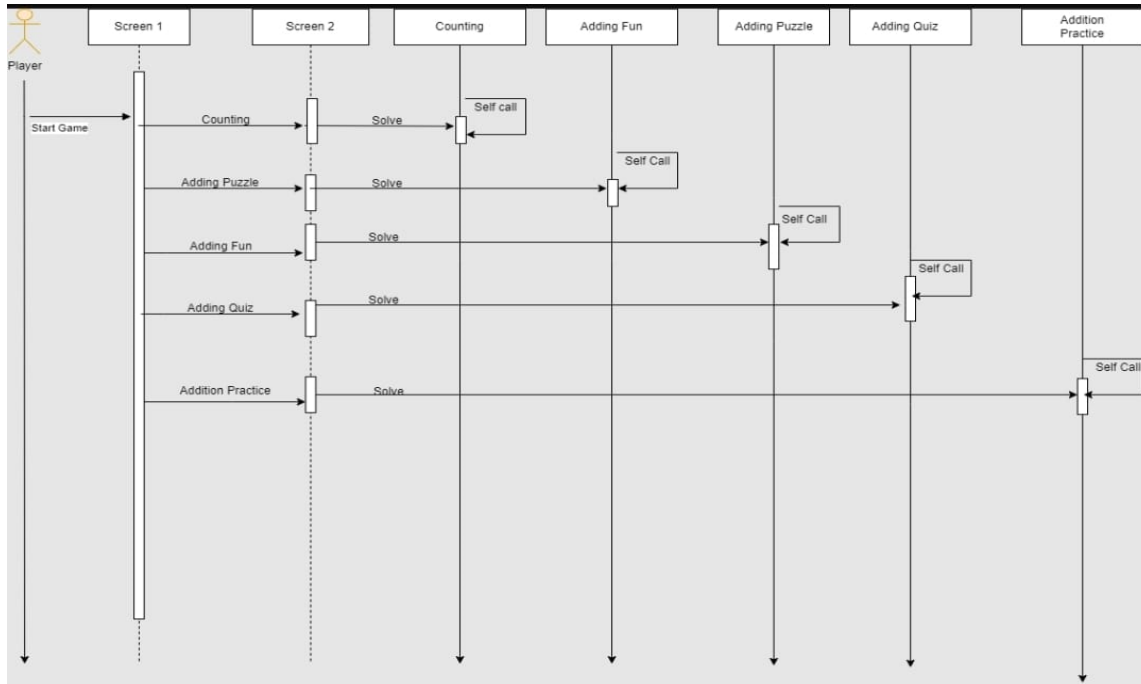
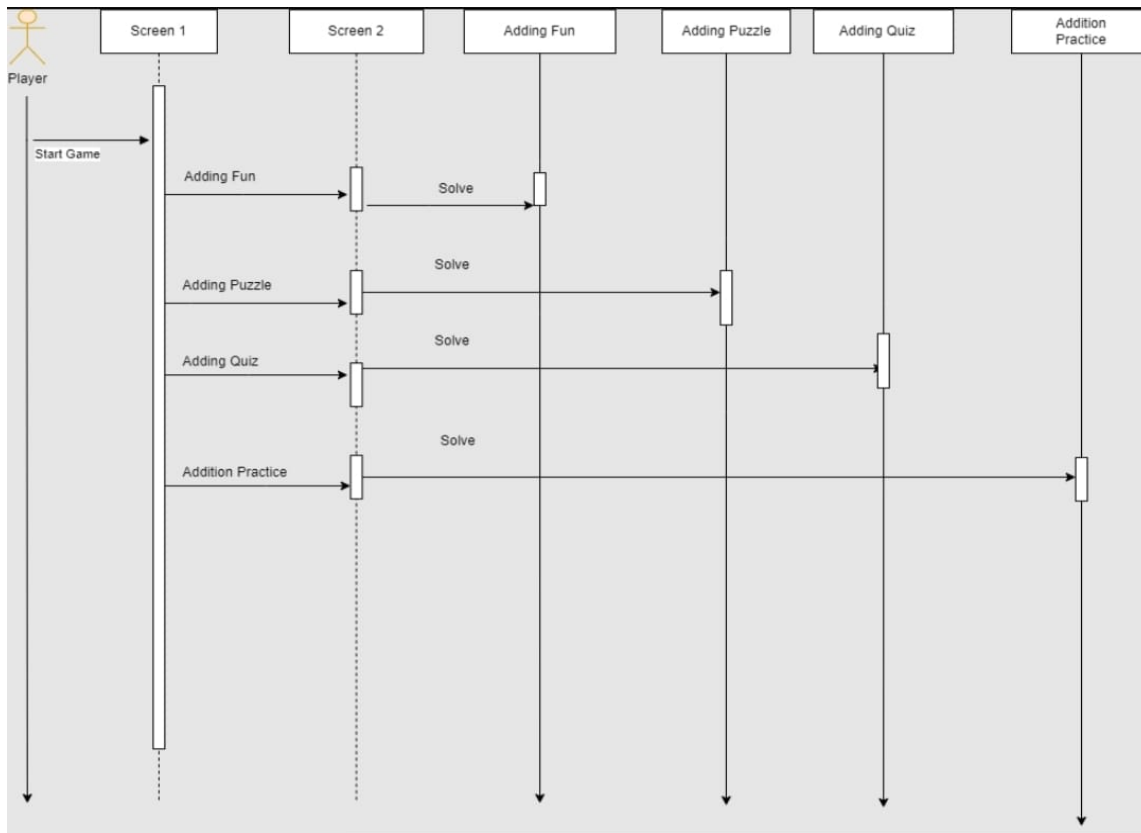
UML diagram describes a system by visualizing the different types of objects within a system and the kinds of static relationships that exist among them. It also illustrates the method operations and attributes of the classes.





C. UML Sequence Diagram

UML Sequence diagram illustrates the sequence 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.



Appendix C: To Be Determined List