Table of Contents

1.0 Task A - Building Graphic User Interface (GUI) with Event Handlings

1.1 Stage 1: Project Requirements, Class Diagram and GUIs

1.1.1 Analysis of Kid`s Mathematics Application Program Requirements

This is an Input Output Processing (IPO) chart used to summarize the basic requirements of Kid's Mathematics Application program based on the assignment requirements.

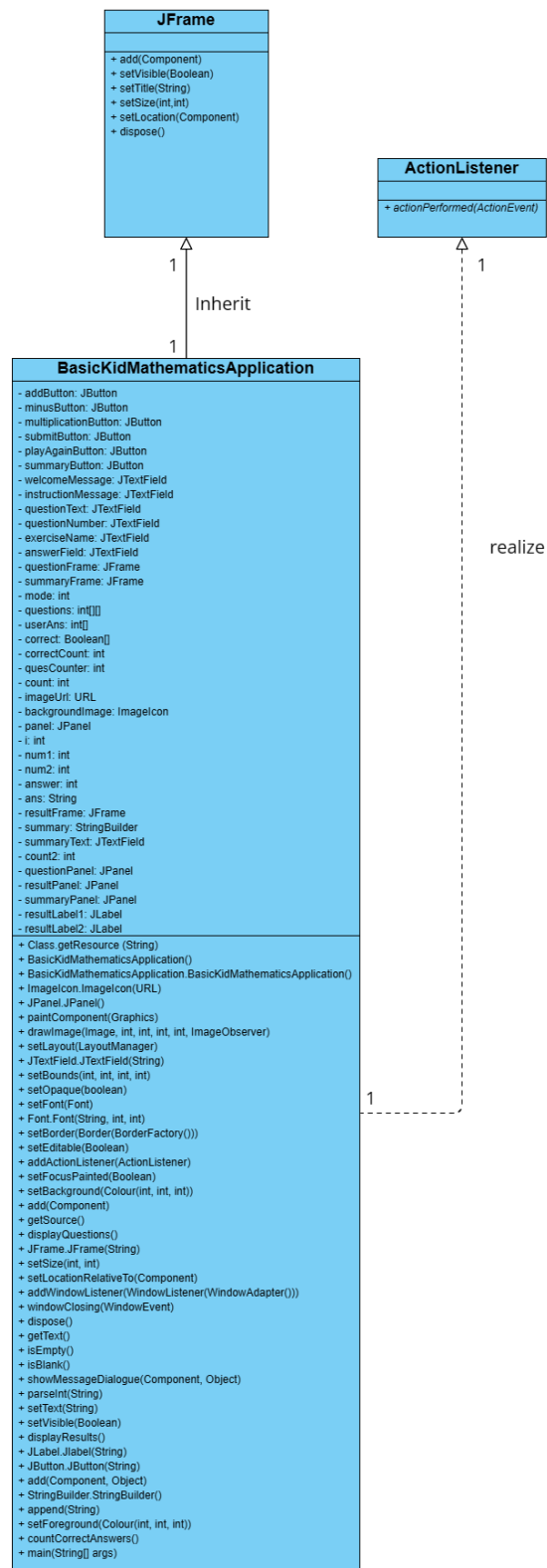| Input	Processing	Output
userSelection Array of userAnswers userSelection2	Get user's selection for selecting games (Addition/Minus/Multiplication) Generate random numbers for the math games questions Loop the generate display of questions process five times according to their selection on games and store the questions and correct answer into an array, then ask for user to input answers and store into an array, check user's answer and store correct/false into an array Validate user's answer as it needs to be numeric, if user key in invalid answer, system will prompt invalid input and ask user to key in the answer again If user choose to close the input message dialogue, the system will close the question and back to the main menu After the loop, display total number of correct answers and selection for user to choose to see summary or play again If the user chooses to see summary, display all 5 questions and shows whether the user's answers are correct or wrong; If the user choose to play again, direct back to the main menu for user to choose math games End the program when user close the program window	math question using random numbers total number of correct answers array of questions and answers array of user's answers array of correct/false

1.1.2 Class Diagram

JFrame

+ add(Component)
+ setVisible(Boolean)
+ setTitle(String)
+ setSize(int,int)
+ setLocation(Component)
+ dispose()

ActionListener

+ *actionPerformed(ActionEvent)*

1

Inherit

1

1

realize

BasicKidMathematicsApplication

- addButton: JButton
- minusButton: JButton
- multiplicationButton: JButton
- submitButton: JButton
- playAgainButton: JButton
- summaryButton: JButton
- welcomeMessage: JTextField
- instructionMessage: JTextField
- questionText: JTextField
- questionNumber: JTextField
- exerciseName: JTextField
- answerField: JTextField
- questionFrame: JFrame
- summaryFrame: JFrame
- mode: int
- questions: int[][]
- userAns: int[]
- correct: Boolean[]
- correctCount: int
- quesCounter: int
- count: int
- imageUrl: URL
- backgroundImage: ImageIcon
- panel: JPanel
- i: int
- num1: int
- num2: int
- answer: int
- ans: String
- resultFrame: JFrame
- summary: StringBuilder
- summaryText: JTextField
- count2: int
- questionPanel: JPanel
- resultPanel: JPanel
- summaryPanel: JPanel
- resultLabel1: JLabel
- resultLabel2: JLabel

+ Class.getResource (String)
+ BasicKidMathematicsApplication()
+ BasicKidMathematicsApplication.BasicKidMathematicsApplication()
+ ImageIcon.ImageIcon(URL)
+ JPanel.JPanel()
+ paintComponent(Graphics)
+ drawImage(Image, int, int, int, int, ImageObserver)
+ setLayout(LayoutManager)
+ JTextField.JTextField(String)
+ setBounds(int, int, int, int)
+ setOpaque(boolean)
+ setFont(Font)
+ Font.Font(String, int, int)
+ setBorder(Border(BorderFactory()))
+ setEditable(Boolean)
+ addActionListener(ActionListener)
+ setFocusPainted(Boolean)
+ setBackground(Colour(int, int, int))
+ add(Component)
+ getSource()
+ displayQuestions()
+ JFrame.JFrame(String)
+ setSize(int, int)
+ setLocationRelativeTo(Component)
+ addWindowListener(WindowListener(WindowAdapter()))
+ windowClosing(WindowEvent)
+ dispose()
+ getText()
+ isEmpty()
+ isBlank()
+ showMessageDialogue(Component, Object)
+ parseInt(String)
+ setText(String)
+ setVisible(Boolean)
+ displayResults()
+ JLabel.Jlabel(String)
+ JButton.JButton(String)
+ add(Component, Object)
+ StringBuilder.StringBuilder()
+ append(String)
+ setForeground(Colour(int, int, int))
+ countCorrectAnswers()
+ main(String[] args)

Figure 1 Class Diagram

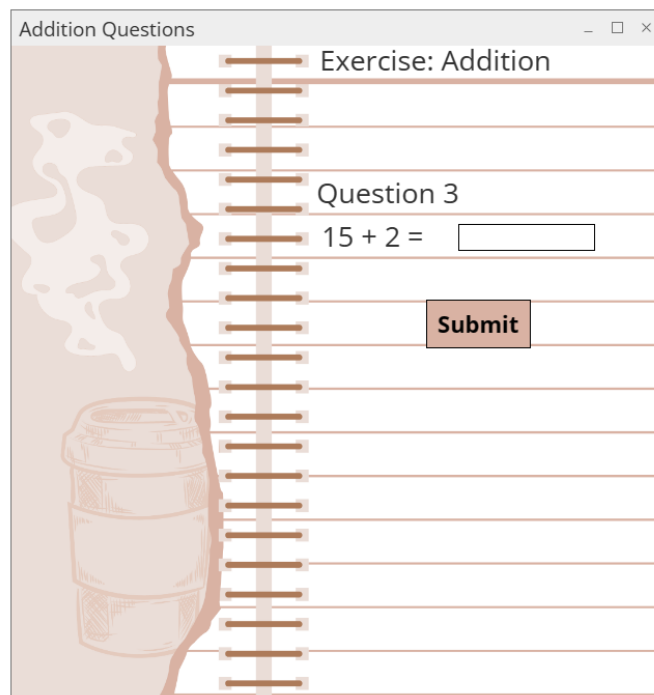**1.1.3 Sketch of GUIS**

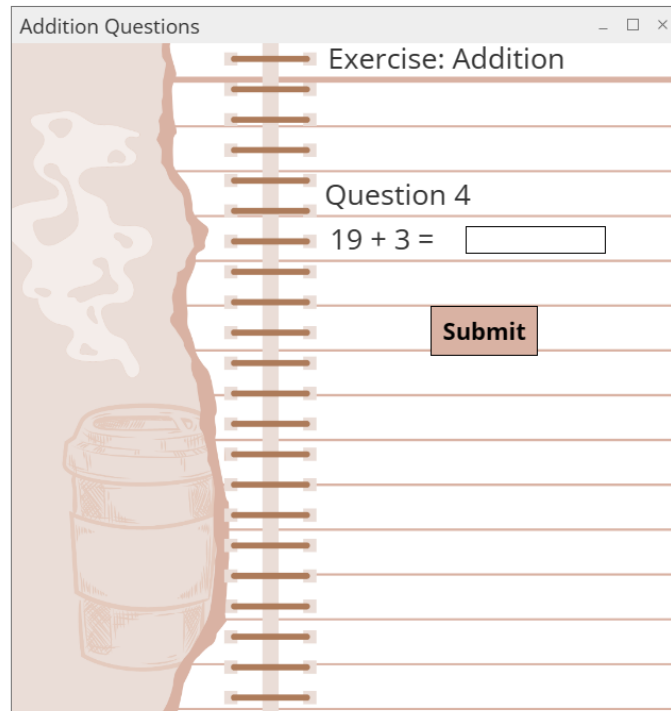*Figure 1.1.1* Select Game Mode

Figure 1.1.2 Addition – Question 1

Figure 1.1.3 Addition – Question 2
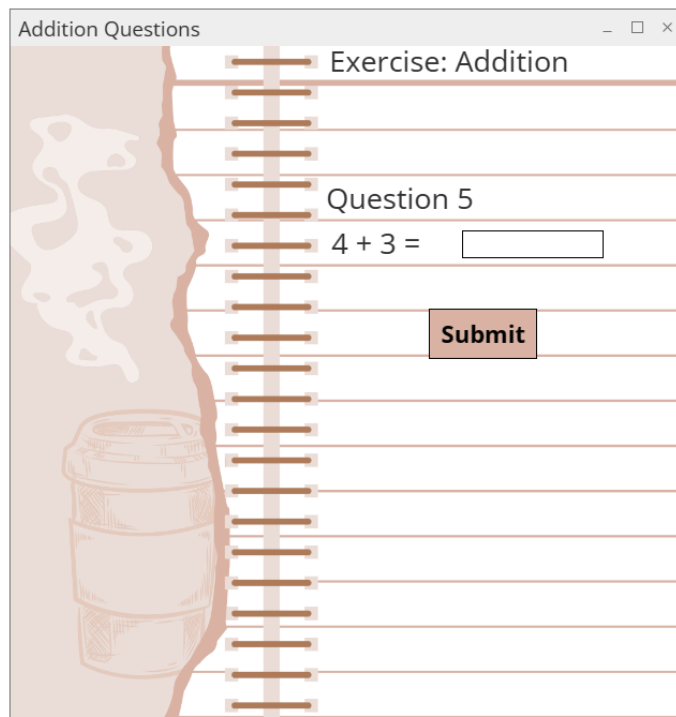
Figure 1.1.4 Addition – Question 3

Figure 1.1.5 Addition – Question 4

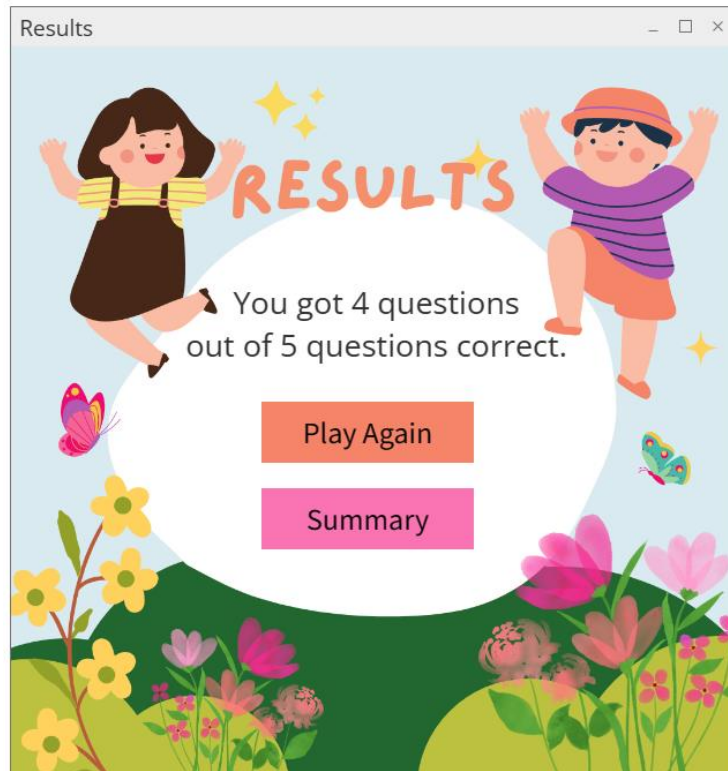*Figure 1.1.6* Addition – Question 5

Figure 1.1.7 Addition – Results
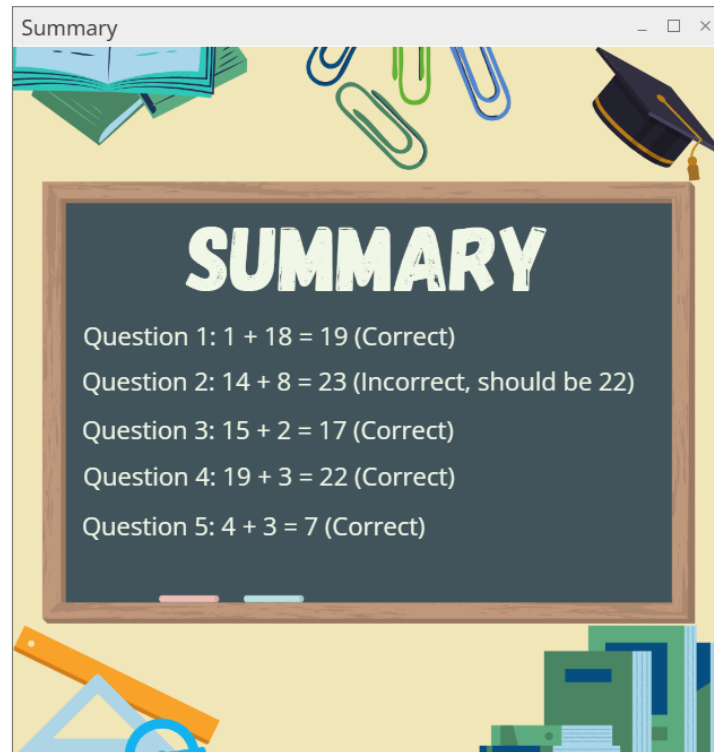
Figure 1.1.8 Addition – Summary
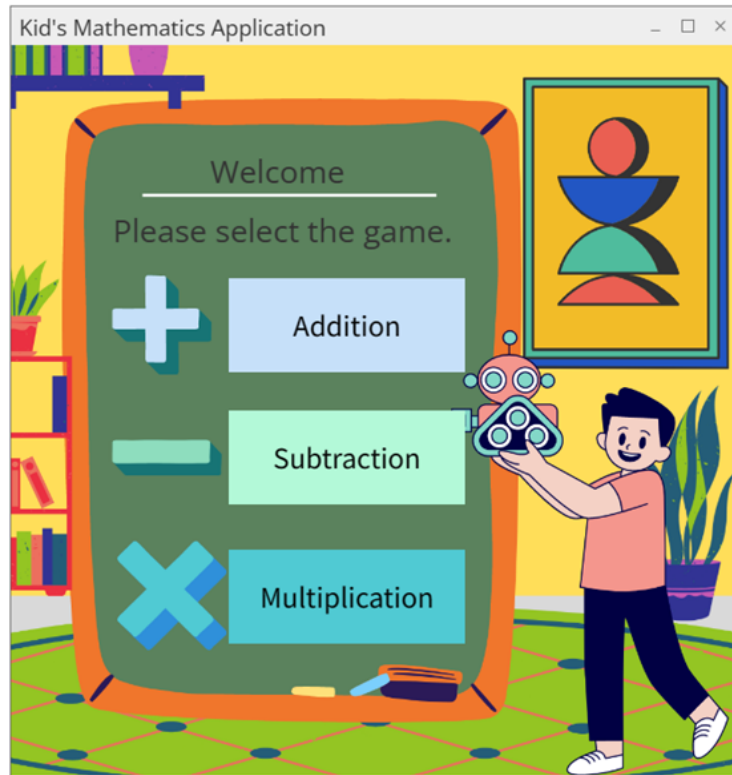
Figure 1.1.9 Addition - Play Again
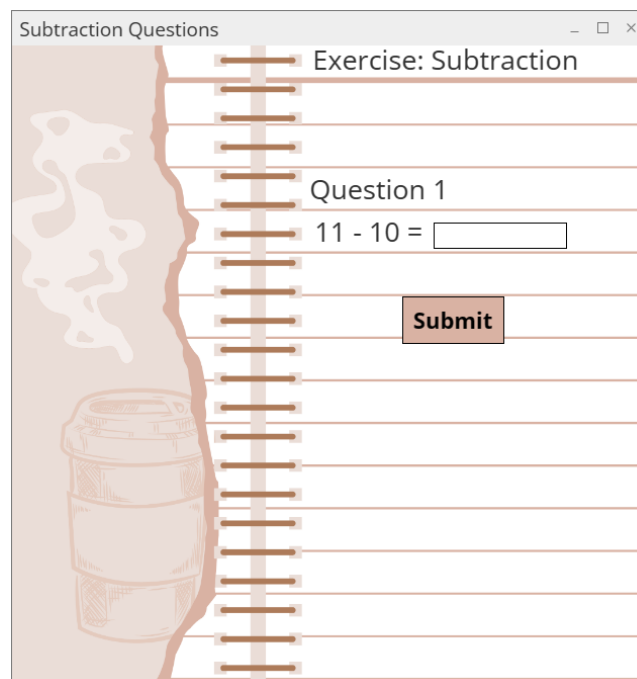
Figure 1.1.10 Subtraction – Question 1

Figure 1.1.11 Subtraction – Question 2

Figure 1.1.12 Subtraction – Question 3

Figure 1.1.13 Subtraction – Question 4

Figure 1.1.14 Subtraction – Question 5
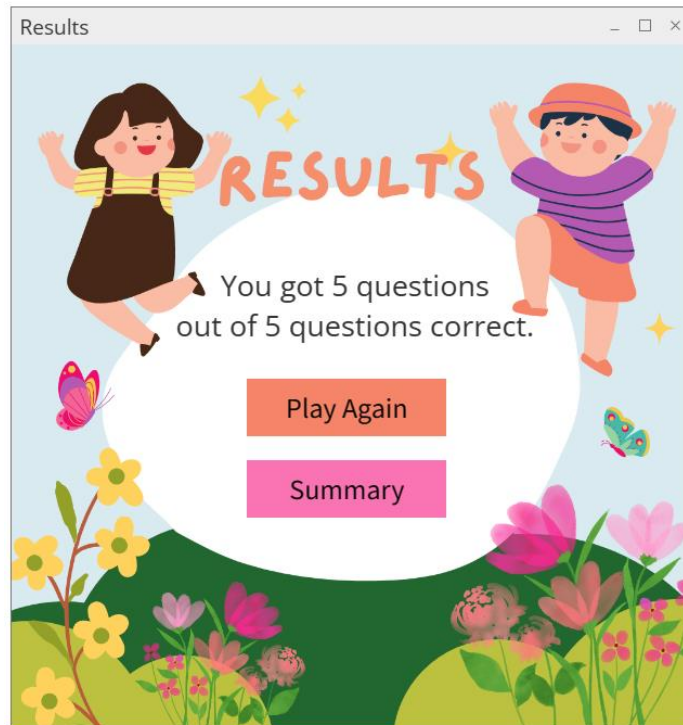
Figure 1.1.15 Subtraction – Results
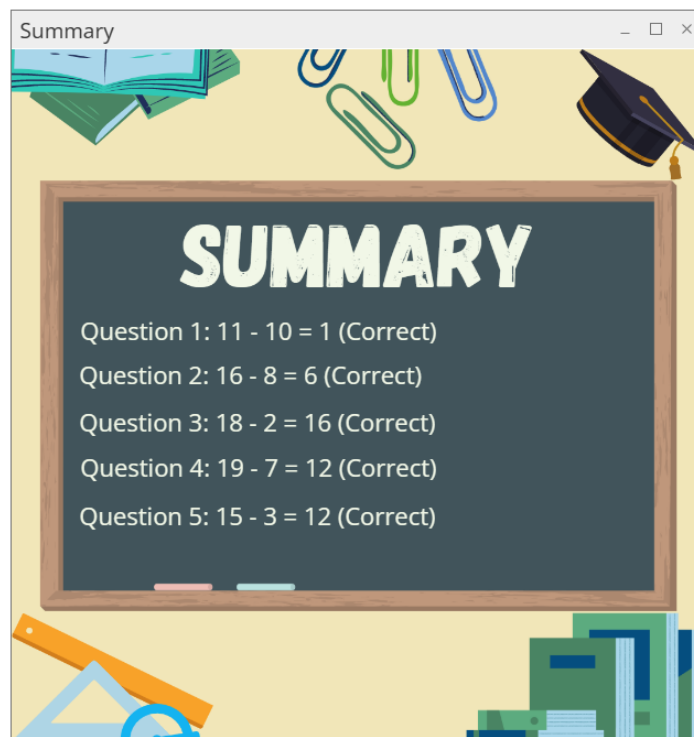
Figure 1.1.16 Subtraction – Summary
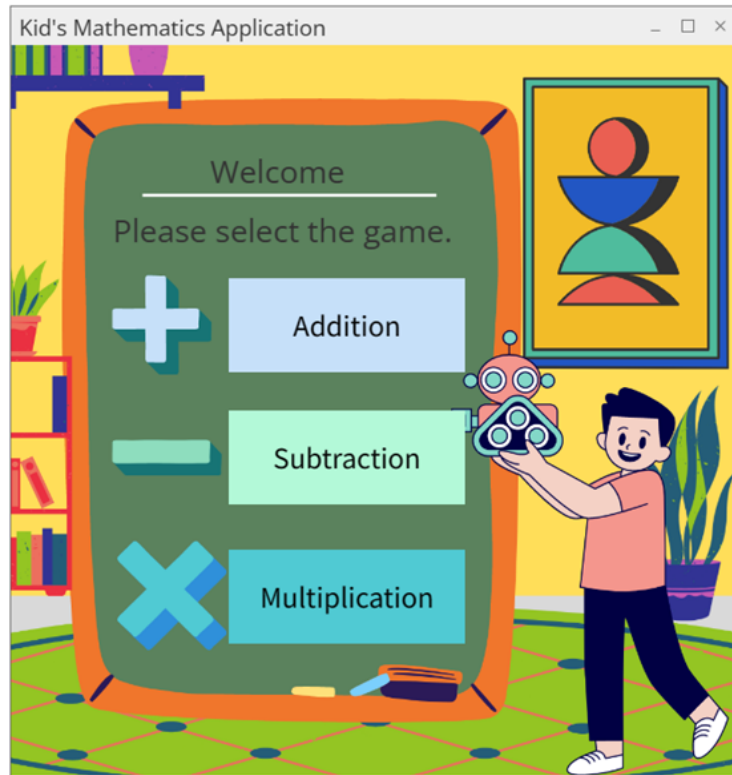
Figure 1.1.17 Subtraction – Play Again
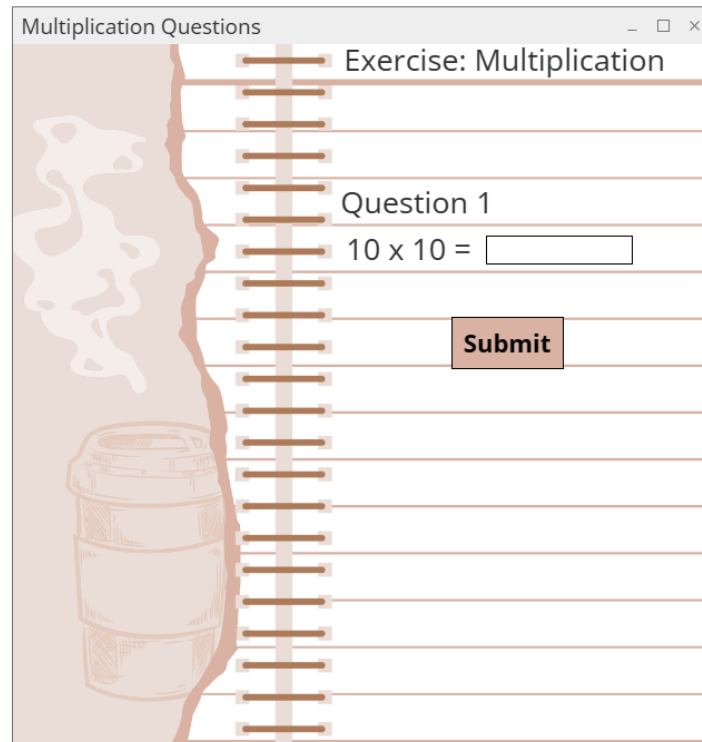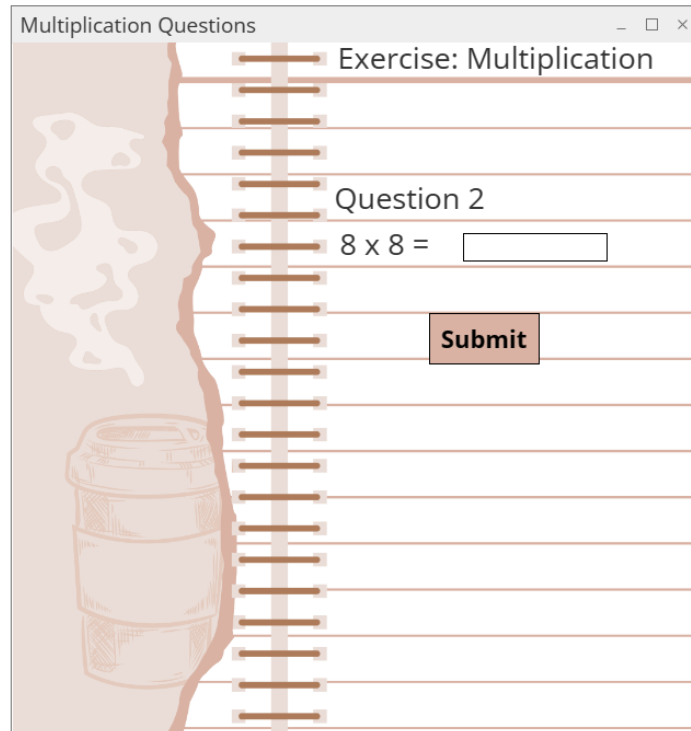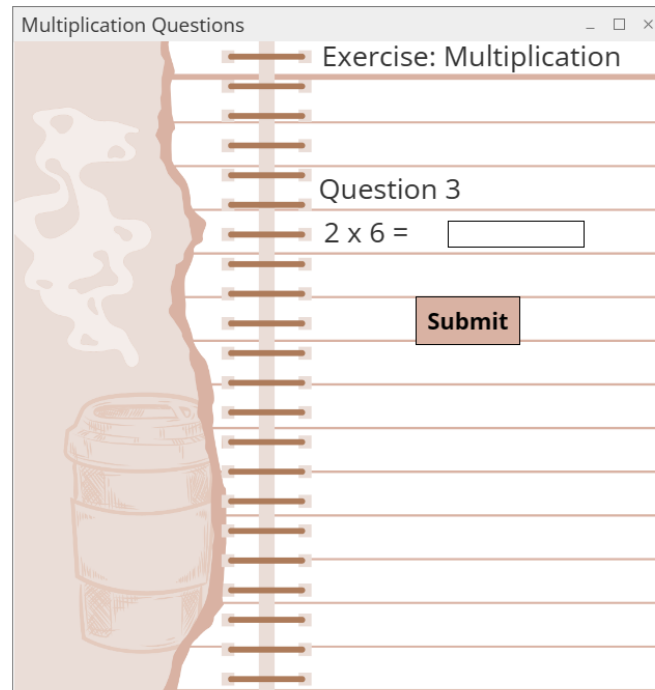
Figure 1.1.18 Multiplication – Question 1
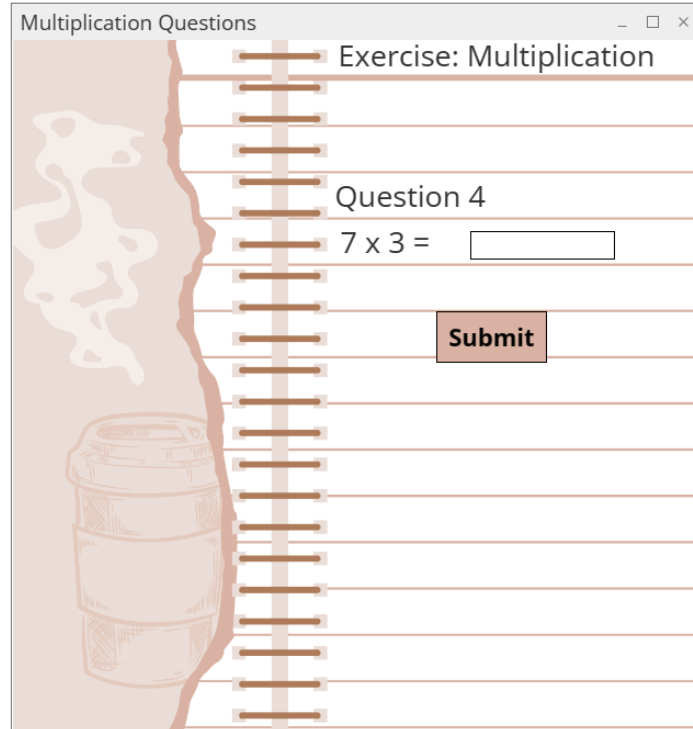
Figure 1.1.19 Multiplication – Question 2
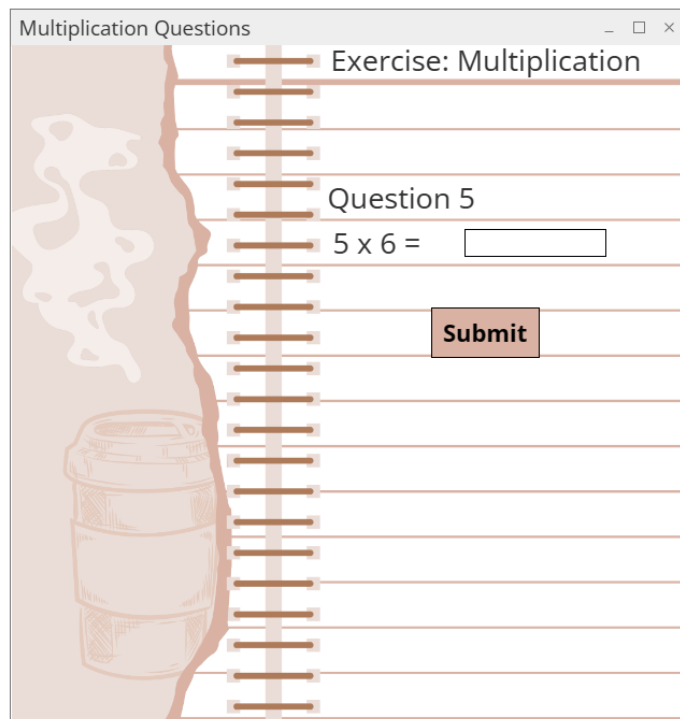
Figure 1.1.20 Multiplication – Question 3

Multiplication Questions _ □ ×

Exercise: Multiplication

Question 4

7 x 3 = [   ]

Submit

Figure 1.1.21 Multiplication – Question 4

Multiplication Questions _ □ ×

Exercise: Multiplication

Question 5

5 x 6 = [   ]

Submit

Figure 1.1.22 Multiplication – Question 5
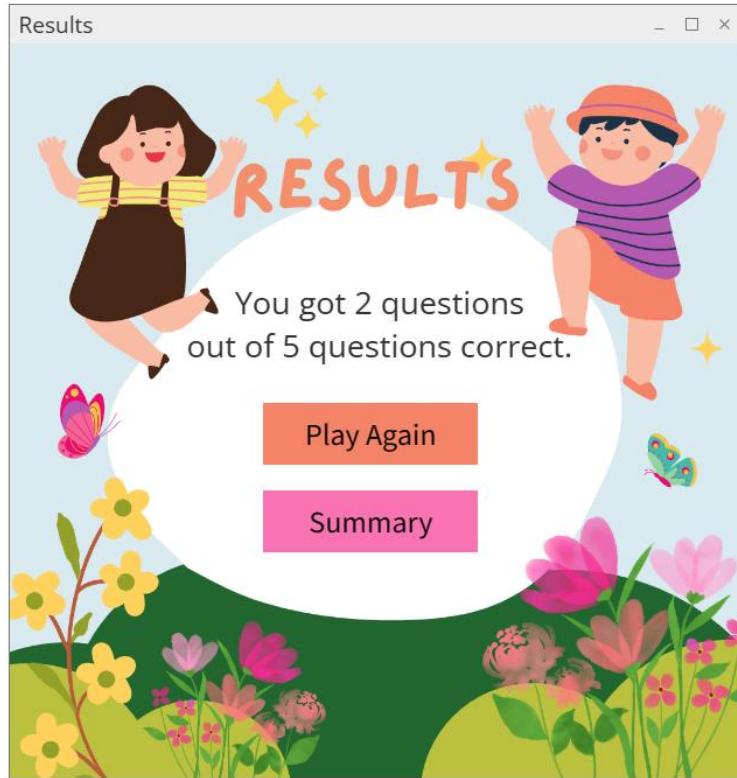
Figure 1.1.23 Multiplication – Results
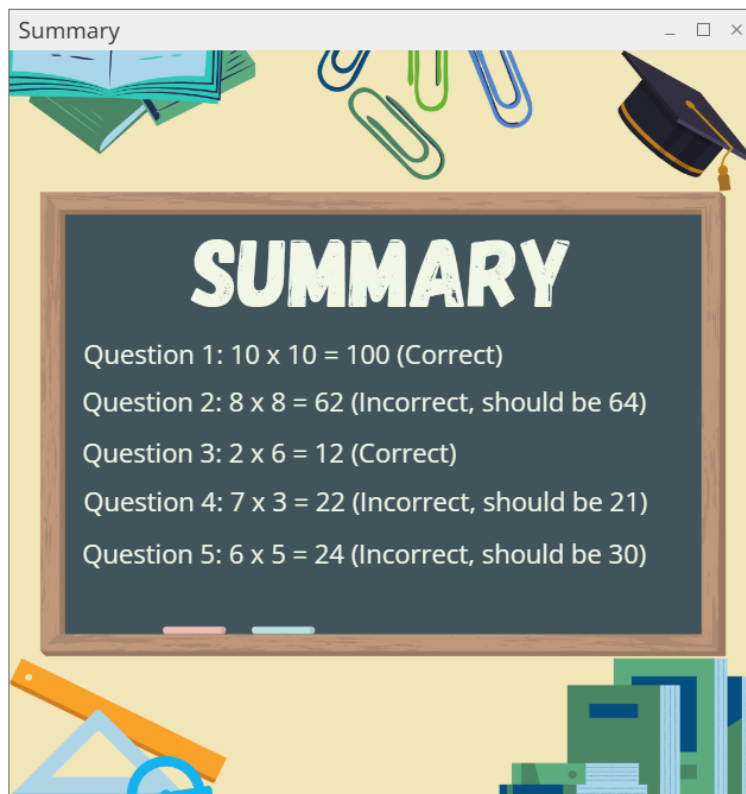
Figure 1.1.18 Multiplication – Summary

1.2 Stage 2: A Basic Working Version

Figure 1.2.1 Kid's Mathematics Application Main Menu
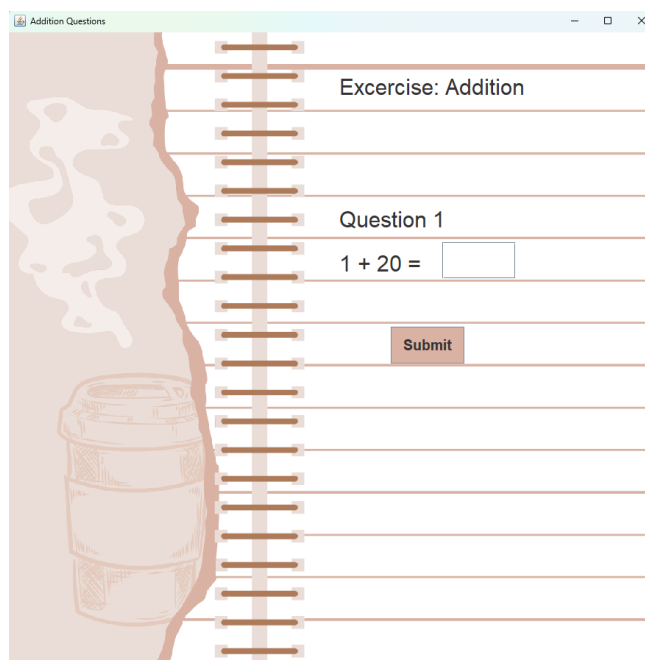
Figure 1.2.2 Addition
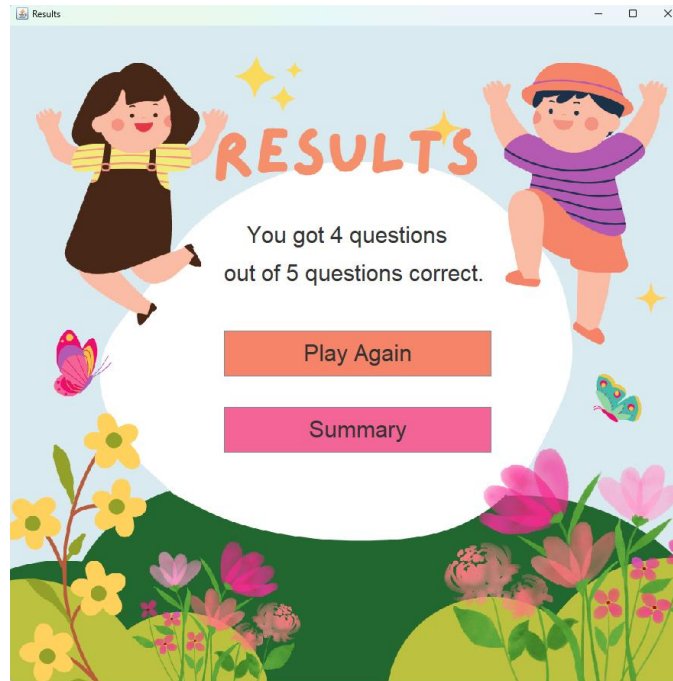
Figure 1.2.3 Result
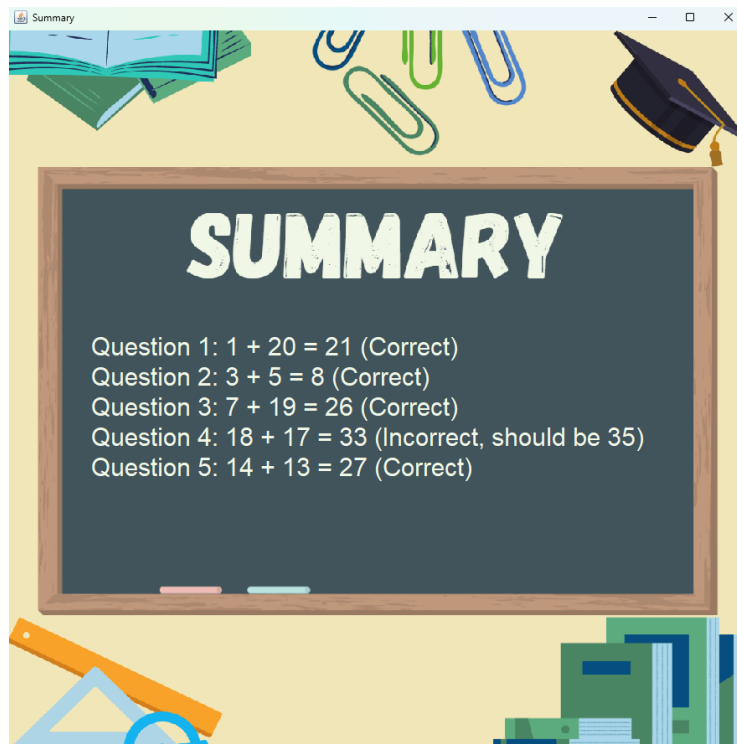
Figure 1.2.4 Summary

Figure 1.2.5 After Clicking Play Again
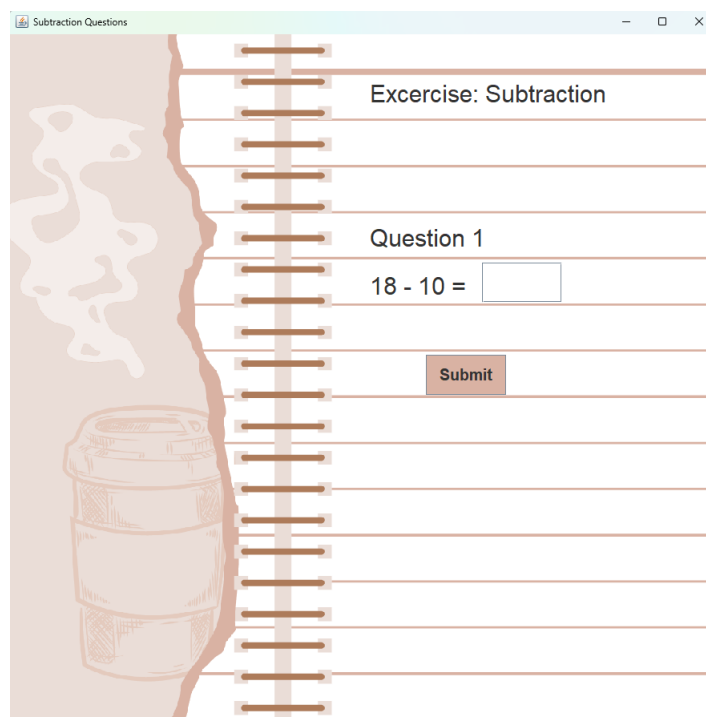
Figure 1.2.6 Subtraction

Figure 1.2.7 Multiplication

Figure 1.2.8 Validation - Invalid Input (Empty)
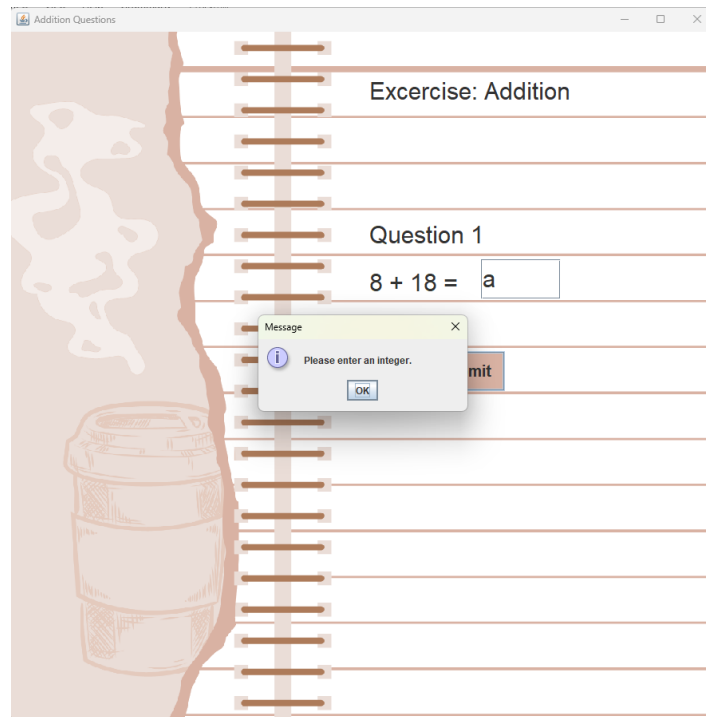
Figure 1.2.9 Validation - Invalid Input (Not Integer)
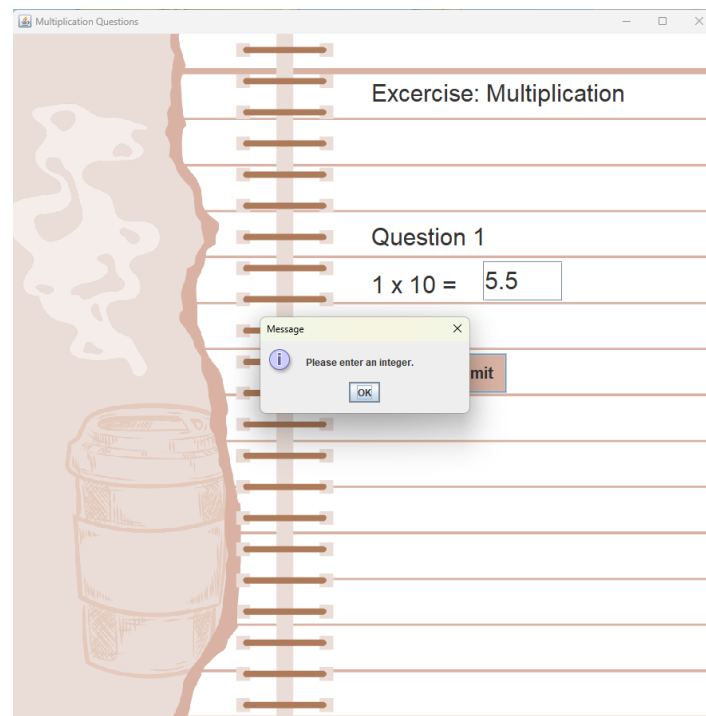
Figure 1.2.10 Validation - Invalid Input (Not Integer)
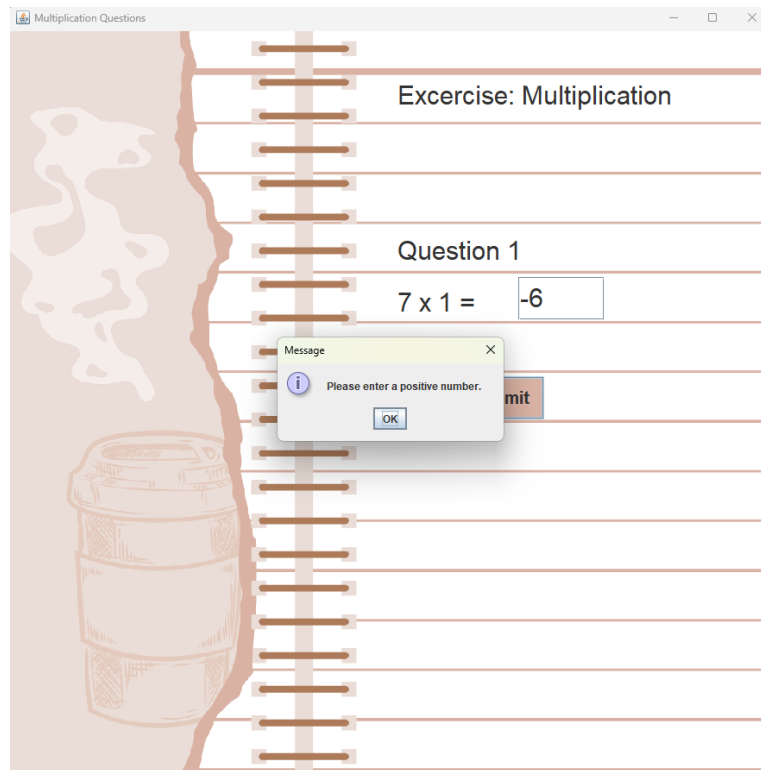
Figure 1.2.11 Validation - Invalid Input (Not Positive Number)

1.3 Stage 3: Enhanced Code and Innovations

For the enhanced version for Kid's Mathematic Application, an input name feature has been added. After running the program, before showing the main menu for user to select game mode, it first shows the enter name frame for user to key in their name and click on the submit as shown in *Figure 1.3.1*.

Figure 1.3.1 Enter Name Menu

This enter name feature will validate on the user's input for the name. After clicking the submit button, it will validate if the name is empty or contain special character/number in the name and ask user to enter again as shown in *Figure 1.3.2*, *Figure 1.3.2.*

Figure 1.3.2 Validation – Invalid Name (Empty)
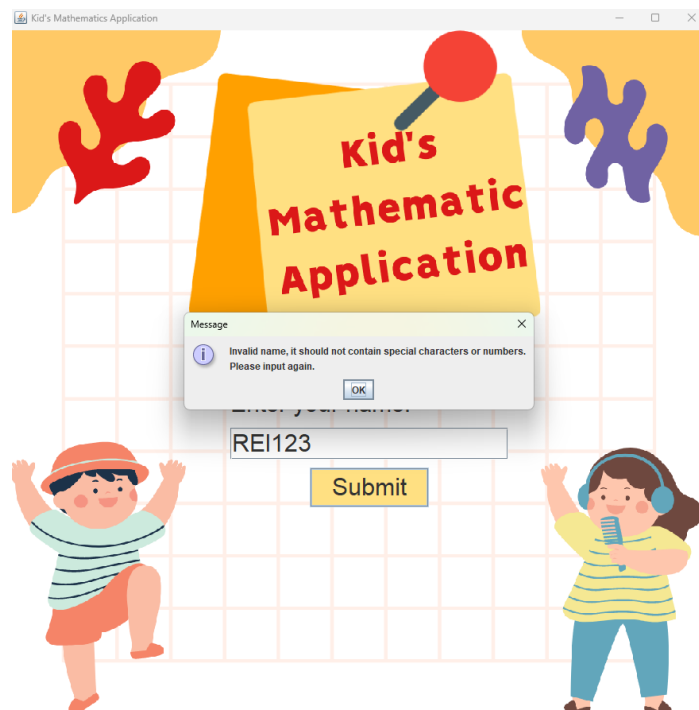
Figure 1.3.3 Validation – Invalid Name (Contain Special Character/Number)

Moreover, after entering a valid name, the program will show user's input name on the welcome message of the main menu as shown in *Figure 1.3.4*.

Figure 1.3.4 Main Menu

In addition, the user's input name will also shown in the questions display frame which make it looks like an exercise with the student's name as shown in *Figure 1.3.5*, *Figure 1.3.6*, *Figure 1.3.7*.

Figure 1.3.5 Addition

Figure 1.3.6 Subtraction

Figure 1.3.6 Multiplication

Furthermore, number pad for user input has also implemented in the enhanced version of Kid's Mathematics Application. The number pad consists of 0 – 9 number with clear and submit button. After clicking the number button, it will appear beside the question to indicate the user's answer. The maximum input is set to be 3 as the greatest number for answer will be 100 (Multiplication: 10x10=100) as shown in *Figure 1.3.7*. The clear button is used to clear the input as user might accidentally clicked on the number while the submit button is to let user to submit their answer for each question and direct to next question as shown in *Figure 1.3.8*, *Figure 1.3.9*.

Figure 1.3.7 When Number is Clicked

Figure 1.3.8 When Clear is Clicked

Figure 1.3.9 When Submit is Clicked

The program will validate user's answer as well after clicking the submit button. If the user does not input any number or clear the answer before clicking the submit button, it will prompt user to enter an answer and ask user to enter answer again for the same question as shown in *Figure 1.3.10*.
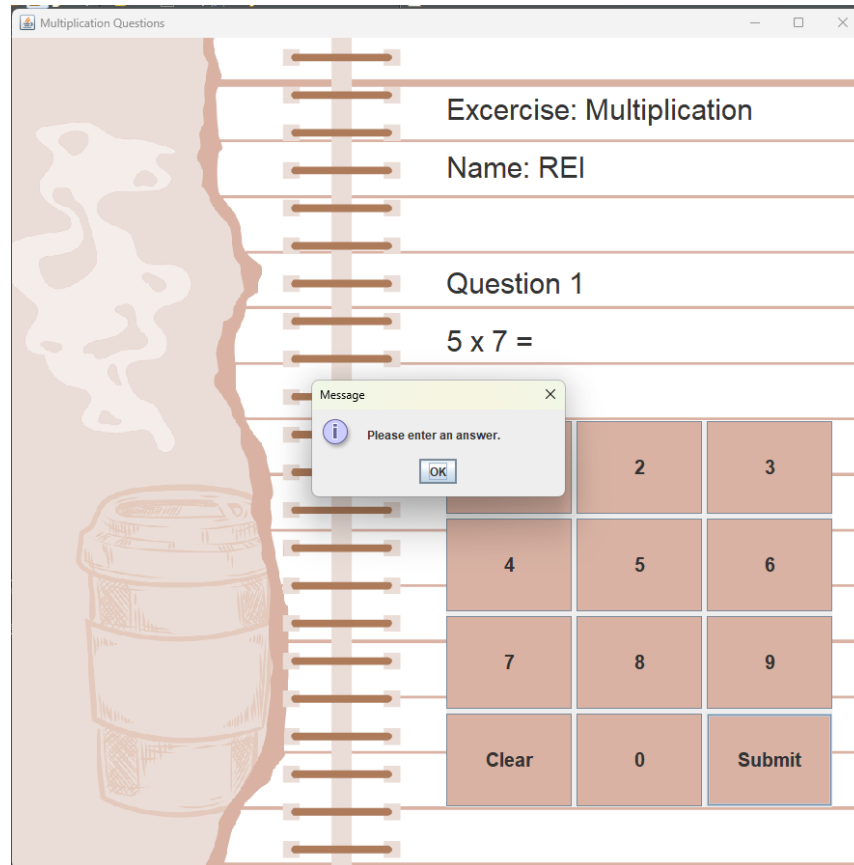
Figure 1.3.10 Validation – Empty Answer

In short, overall improvement or innovation that have added into the Kid's Mathematics Application project in this stage includes the asking name feature, number pad feature and design on the main menu and question display frame which is displaying user's name on welcome message and displaying name on question display frame which makes it look like a real exercise book. In addition, validation is added to the new features as well, such as validate user's name and validate user's input answer.

1.4 Stage 4: Final Report

1.4.1 Introduction of the prototype model

Kid's Mathematic Application is a math game application that designed for children. The system consists of three game mode, which is, addition game, subtraction game and multiplication game. Each game will generate 5 questions and ask user to input the answers. Addition game will generate addition questions with the numbers between 1–20 (eg: 18+15=?). Subtraction game will generate subtraction questions with numbers between 11-20 minus on numbers between 1-10 (eg: 19-10=?). Multiplication game will generate multiplication questions with numbers between 1-10 (eg: 10x10=?). The system will validate the user's answers every time they clicked on the submit button to check so that the answer should not be empty or non-numeric. If user submitted an invalid answer, the system would prompt the error and ask user to enter again for the same question. After user answered 5 questions, the system will display the results, which is the total numbers of correct answers (eg: 4 out of 5 corrects) and will show two options, which is summary button and play again button. If summary button is clicked, the system will show all 5 questions with their answer and shows whether they are correct on that question or not, if they got that question wrong, the system would display the correct answer beside the incorrect message. If the user clicked on the play again button, the system would direct back to the main menu and let the user to select on game mode. If the user clicked on the close window button, the system will eventually close. Since it is an application designed for children, and the math game is considered very easy and only suitable for children below 9 years old, the potential user would only be children.

1.4.2 Explanation on UML Class Diagram and Design of GUIS

As mentioned in **1.4.1 Introduction of the prototype model**, the potential user of Kid's Mathematic Application would be children, the designs for GUI should be as interesting and cute as it could be. The reason is that, if the user interface designed in the way for adults, children will find it very complicated and uninterested on playing with the application. Suitable amount of picture element that are with the illustration style intended for children added into the application would make the system seems to be more children-friendly and interesting to play with. For example, for every frame of the application, there are cute background images to make sure the application does not seem to be tedious as shown in the screenshot in **1.2 Stage 2 A Basic Working Version**. Moreover, there are also picture elements that can understand the button without having to read on the word. For example, beside the addition, subtraction and multiplication button, there are picture of (+, -, x) with respect to their button text as shown in *Figure 1.4.1*. As children would not like to read many words, having those images can let children to understand the system without having to read words. Moreover, the designs of the system's GUI also are as minimalist and aesthetics as it could be, the reason is that, complicated and ugly user interface would make children lose interest on exploring and playing with the application. In the class diagram in **1.1.2 Class Diagram**, a lot of operations are included in the system, so that it is able to customise the system as the desired output, which included the background, bigger size font, colourful buttons and so on. Moreover, there are also methods with their respective task to accomplish such as displayQuestions(), displayResults(), countCorrectAnswers(), it can maximize the reusing of codes as well as making the codes more organized.

Figure 1.4.1 Picture elements

1.4.3 Explanation on the Program Main Function

The program randomize numbers by using the Java built-in functions, which is random() operation in Math class as shown in *Figure 1.4.2*. The *20 meaning that the range of the randomize number is 20 while the +1 is where the range start, in this case, it is start from 1 and to 20.

```java
if (e.getSource() == addButton) //if addition game is clicked
{
    mode=1; //set game mode to 1
    for(int i=0;i<5;i++) { //loop 5 times to generate 5 questions
        // Generate 2 random numbers for addition
        int num1 = (int) (Math.random() * 20) + 1; //generate random number from 1-20
        int num2 = (int) (Math.random() * 20) + 1; //generate random number from 1-20
        int answer = num1+num2; //calculate correct answers
        //Store questions and correct answers into array
        questions[i][0] = num1;
        questions[i][1] = num2;
        questions[i][2] = answer;
    }
```

Figure 1.4.2 Randomize numbers

Next, the program perform error checking by using try catch and java built-in function isEmpty() and isBlank(). First, the program will try to catch the error while performing the parseInt(String), which is try to parse a string into integer for further operation, if the string contains non-number, it will occur error where the catch(NumberFormatException exception) will catch it, and it will prompt user to enter an integer, after that ask user to input again for the same question. In addition, it will also check if the user does not enter anything and clicked on submit button by using the isEmpty() and isBlank() operation, if one of the condition is true, it will ask user to key in an answer before clicking the submit button, and then ask the user to key in again the answer for the same question as shown in *Figure 1.4.3*, *Figure 1.4.4*.

```java
public void actionPerformed(ActionEvent e) { //when submit button is clicked
    if (quesCounter <= 4) { //only run when user have not finish the question
        try
        {
            String ans = answerField.getText();
            if (ans.isEmpty()||ans.isBlank()) { //check if answer is empty
                JOptionPane.showMessageDialog(questionFrame, "Please enter an answer.");
                return; //if is empty, back to the question
            }
            userAns[quesCounter] = Integer.parseInt(ans); //parse int user answer from string
            if (questions[quesCounter][2] == userAns[quesCounter]) {//check if the answer is correct
                correct[quesCounter] = true; //if correct store as true
            } else {
                correct[quesCounter] = false;//else wrong store as false
            }

            quesCounter++; //count the question
            if (quesCounter == 5) { //after user finish 5 questions
                displayResults(); //display the results
                questionFrame.dispose(); //close the question frame
            }
            if (quesCounter <= 4) { //if user have not finish 5 questions, display question
                switch(mode) //dislay question according to game mode
```

Figure 1.4.3 isEmpty(), isBlank()

```java
                {
                case 1:
                    questionNumber.setText("Question "+(quesCounter+1));
                    questionText.setText(questions[quesCounter][0] + " + " + questions[quesCounter][1] + " = ");
                    break;
                case 2:
                    questionNumber.setText("Question "+(quesCounter+1));
                    questionText.setText(questions[quesCounter][0] + " - " + questions[quesCounter][1] + " = ");
                    break;
                case 3:
                    questionNumber.setText("Question "+(quesCounter+1));
                    questionText.setText(questions[quesCounter][0] + " x " + questions[quesCounter][1] + " = ");
                    break;
                }
            answerField.setText(""); //reset the text field everytime when finishing one question
        }
        }catch(NumberFormatException exception) //catch non numeric input
        {
            JOptionPane.showMessageDialog(questionFrame, "Please enter an integer.");
            answerField.setText(""); //reset the text field
            return; //if is empty, back to the question
        }
    }
}
```

Figure 1.4.4 try catch(NumberException exception)

The system creates GUI by initializing and declaring the component such as JLabel, JButton, JTextField, JPanel and JFrame and added into the panel and frame and add action listener to the JButton so that it can function as user clicked on it and set the frame visible as shown in *Figure 1.4.5*, *Figure 1.4.6*, *Figure 1.4.7*, *Figure 1.4.8*, *Figure 1.4.9*, *Figure 1.4.10*.

```java
{
case 1:
    questionFrame = new JFrame("Addition Questions");
    break;
case 2:
    questionFrame = new JFrame("Subtraction Questions");
    break;
case 3:
    questionFrame = new JFrame("Multiplication Questions");
    break;
}
```

Figure 1.4.5 Declaring JFrame

```java
// create a panel with the image
JPanel panel = new JPanel() {
    @Override
```

Figure 1.4.6 Initializing JPanel

```java
JLabel resultLabel1 = new JLabel("You got " + correctCount+" questions");
JLabel resultLabel2 = new JLabel("out of 5 questions correct.");
JButton playAgainButton = new JButton("Play Again");
JButton summaryButton = new JButton("Summary");
```

Figure 1.4.7 Initializing JLabel, Jbutton

```java
playAgainButton.addActionListener(new ActionListener() { //add action listener
    public void actionPerformed(ActionEvent e) { //if play again is clicked
        resultFrame.dispose(); //dispose result frame
        if(count>=1) //if summary frame is open already
        {
        summaryFrame.dispose(); //dispose summary frame
        }
        new BasicKidMathematicsApplication(); //call game mode menu
    }
});
```

Figure 1.4.8 Adding Action Listener

```
questionText.setBounds(450, 240, 150, 150); //set size and location
questionText.setFont(new Font("Arial", Font.PLAIN, 30)); //set font size and style
questionText.setBorder(javax.swing.BorderFactory.createEmptyBorder()); //clear border of text field
questionText.setOpaque(false); //set text field to transparent
questionText.setEditable(false); //set text field cannot be edit
questionPanel.add(questionText); //add text field onto panel

JTextField answerField = new JTextField(3); //initialize answer field text field
answerField.setBounds(590, 285, 100, 50); //set size and location
answerField.setFont(new Font("Arial", Font.PLAIN, 30)); //set font size and style
questionPanel.add(answerField); //add text field onto panel

JButton submitButton = new JButton("Submit"); //initialize submit button
submitButton.setFont(new Font("Arial", Font.BOLD, 20)); //set font size and style
submitButton.setBounds(520, 400, 100, 50);
submitButton.setBackground(new Color(217, 178, 163)); //set button colour by RGB
submitButton.setFocusPainted(false); //clear focus painted
questionPanel.add(submitButton); //add submit button onto panel
submitButton.addActionListener(new ActionListener() { //add action listener
```

Figure 1.4.9 Initializing JTextField

```
summaryPanel.add(summaryText); //add text onto panel
summaryFrame.add(summaryPanel); //add panel onto frame

summaryFrame.setSize(900, 900); //set frame size
summaryFrame.setLocationRelativeTo(null); ///center the frame
summaryFrame.setVisible(true); //set frame visible
```

Figure 1.4.10 Adding Component to JPanel and JFrame

1.4.4 Evaluation on the System

I.   Faults & Failures

As sufficient validation is added into the system, and fully debug and tested every run showing that there is no problem on the program, there are no faults or failures of executing the project or any features could not be implemented.

However, during the process of implementing the system, there are several difficulties faced. For example, due to lacking understanding about GUI coding, at first, I could not implement a proper frame for displaying question, as the first approach, I use for loop to initialize and add the component onto the frame, but it will not wait the user to submit answer before going to the next loop. At the moment, I was going to give up on the desired output, instead, using message dialogue to display question. However, having consultation with experienced person, he told me that if we do not do any changes on the interface, the created interface will be forever there. This is when I got enlightened, the loop should be happened at setting the JTextField text for displaying the question instead of creating the interface. Eventually, I got to get the desired output by changing the setText(String) text every time user click on the submit button. Other than that, I do not face any severe failures on implementing the system, all the problems could be solve by reading materials online.

II.   Strengths of the System

One of the strengths of the system is the designs of the GUI is minimalist and aesthetic which include cute illustration that intended for children in the background. This will make the application to be more interesting for child to play along it. They will not feel tedious to explore the application. Moreover, image to represent the function is also implemented in the application. Children will not need to read the text could easily understand how to use the application. In addition, another strength is most of the interaction of the GUI only need mouse clicking, which is very suitable for a child to play with, as they will prefer all interaction to be as easy as only clicking the mouse. For example, number pad is implemented in the answering questions system so that the user only needs to click on the number pad to key in the answer while answering the questions without typing on the keyboard. Furthermore, another strength of the system is the enter name function, after entering the name, the user's name will appear in the welcome message of the main menu as well as under the exercise title of the answering question interface, which make the system even more interesting. In addition, another strength is for the summary display, the system does not show only the correct answer but user's answer as well for them to check if they have careless mistake and to be caution next time. Moreover, another strength of the system included the complete validation to prevent faulty system. For example, validation on user's name as well as the user's answer is provided, it could prevent user from entering blank or invalid name and answer such as name with non-letters or answer with non-numbers.

1.4.5 Conclusion

In conclusion, Kid's Mathematic Application is a math game system designed for children. The GUI designed is aesthetic and minimalist with interesting picture as to be children-friendly user interface. The strengths of the system included cute, interesting GUI design which is suitable for being a kid-based application, images to represent function, mouse-clicking based interaction for most of the interaction and extra features making the system more interesting. Moreover, the strength included system displays user's answer with correct answer as well if they got it incorrect. Furthermore, another strength is sufficient validation to avoid faulty program.

Word Count: 2000

2.0 Task B – Testing and Demonstration

2.1 Test Plan

Selected Function: Addition Function
Test Plan: Black Box Testing

Boundary Value Analysis (BVA) Table:

Input Condition	Valid Equivalence Class	Valid Boundaries	Invalid Equivalence Class	Invalid Boundaries
ans				
content	0-999	0 999	<0 >999	-1 1000
type	integer		non-integer	

As user could only enter the answer by the number pad provided in the display question frame and the limit of entering the answer is only 3-character set in the system, invalid cases such as negative value (<0) or greater than the limit (>999) will not occur, the only invalid answer will occur is blank answer which is not an integer. Hence, the test data will be testing how the system handle the only possible invalid answer which is blank answer.

2.2 Test Data

Test Data:

Step	num1	num2	ans	Operation	Output
1	15	19			Please enter an answer.
2	15	19	34		
3	20	15	35		
4	16	4			Please enter an answer.
5	16	4	20		
6	19	11	999		
7	13	12	25		
8					You got 4 questions out of 5 questions correct.
9				Click on Summary Button	
10					Question 1: 15 + 19 = 34 (Correct) Question 2: 20 + 15 = 35 (Correct) Question 3: 16 + 4 = 20 (Correct) Question 4: 19 + 11 = 999 (Incorrect, should be 30) Question 5: 13 + 12 = 25 (Correct)

2.3 Self-Reflection Report

Overall, I learnt a lot and experienced a lot from this project. This is my first assignment that will require us to develop a Graphical User Interface (GUI). As this is my first time dealing with GUI, there are a lot of difficulties and challenges that I have faced during implementation of Kid's Mathematic Application. At the beginning, when I am sketching the GUI, I have a lot of innovations and creative that I wanted to add into the project, and I am also afraid I could not make it at the end. However, after overcoming such challenges, I eventually get to implement the ideal GUI for the project.

During stage 1, I learnt and understand how to analysis the project's requirements, how to sketch GUI using online tools, how to draw a class diagram in a proper format when I am dealing with the task. Next, another thing I learnt and understand from the project is how to implement the basic GUI, which has the frame, panel, label, button, text field and so on as dealing with the task in stage 2 and stage 3. Then, I learnt how to implement a GUI with event handling which is to add event listener to the GUI so that user can interact with the button. Moreover, I learnt how to customize the component of the GUI, such as the colour by Red, Green and Blue (RGB), the size, the location of the component. I also learnt how to customize the font and the way to display it, such as transparent background and borderless text field. In addition, I also learnt how to direct the user from one frame to another and how to manage the operation when user click to close the program's window. In addition, I learnt how to implement a complete validation for a GUI program which can also prevent the user from making error such as limiting the character of user's input in the JTextField. Moreover, I also learnt how to implement a simple math game on GUI. Furthermore, I have learnt how to explain the code logically so that others could easily understand it. In addition, I have learnt how to evaluate the program such as its strengths. Moreover, I have learnt how to produce a complete test plan with test data to test the program. Last but not least, by completing the task for this project, I have learnt to think the solution of the problem with another way instead of getting into a blind alley.

There are a lot of difficulties and challenges I have faced throughout the process of implementing the Kid's Mathematic Application. The first challenges are at the beginning of the project, I do not have much understanding in Java GUI and as mentioned above, this is my first time implementing a GUI, I do not have experience to deal with the bugs. The progress was very slow at the beginning, to overcome these challenges, I have to read a lot of reference from the online materials to understand how to implement the GUI with the expected outcome of my ideal GUI. Moreover, I also took a long time to deal with the bugs since I do not understand why the error is occurring, hence, to overcome this situation, I have to search online to see if someone encountered the error before and how they fix it. One of the most severe difficulties I have encountered throughout the project is to implement a display question frame, since it needs to loop the display questions and input answers process five times, at mention of loops with fixed condition, the first thing came into my mind is for loop. Hence, at the beginning when I am trying to implement the questions frame, I used for loop to initialize the questions and answers text field as well as the submit button, however, it does not wait the user to click on the submit button before entering the next loop. There is where my idea failed. I try to search online but nothing appears to similar to my situation. At the moment, I was going to give up for implementing the ideal GUI, instead, using a JOptionPane input dialogue to display the questions and ask for user to key in answers which is pretty undesired solution as the input dialogue cannot be customized and the user interface will be very tedious for a child to play with. However, after having consultation with an experience person, he enlightened me by stating out the fact that, after creating a component, if does not do any changes on the component, it will stay there. At the moment, I realized I have fall into a blind spot of thinking as I do not have deep understanding on GUI. Hence, after the conversation, I change the structure of the code into, where every time the user clicks on the

submit button, the program will then only change the text on the text field to the next questions, which make it loop the questions five time without using for loop, instead, a more logical way for a GUI. Other than that, I do not face any other difficulties or challenges that make me as perplex as when dealing with the questions frame problem. All the other challenges are minor issues that I could fix it easily or fix it by reading the solution in the online forum. As a result, I am able to implement the ideal GUI eventually.

In conclusion, it is a very valuable experience on working on this project. I have gained a lot of precious experience and understanding on creating a GUI program with event handling, as well as the knowledges on object-oriented programming, which included inheritance, polymorphism, abstraction and software testing. I also gained a lot of experiences on debugging, and implementing complete validation and I am confident that, I will debug faster in the future by the experienced gain from the assignment. Moreover, I also learnt to overcome challenges while developing a program. Last but not least, I am happy with the product of this project, which is the simple math game with well-designed GUI.