Semester Project Report

Project Title	Quiz Game Project

Member	Name of Student	Roll No
1	Abubakar Saeed	225338

Deadline	17th July 2023, till 11:59 PM by each group via email and other members in Cc
Submission	Submit both (.doc & .PDF) file without modifying the document formatting
Subject	Subject of email must be updated according to project title
Print	Also submit hard (printed) copy of report till 17th July 2023 during lecture
Email	waqarhgcuf@gmail.com

Department of Computer Science
Institute of Arts and Sciences (Chiniot Campus)
Govt College University Faisalabad

Declaration

I hereby declare that the project work entitled "*Quiz Game Project*" submitted as a semester project for programming fundamentals *(CSI-302)*, is my own work, based on personal study, research, and sources used in its preparation, whether they be books, articles, reports, and lecture notes.

I also certify that this project work is submitted in the partial fulfillment of the requirements to pass the PF course and has not previously been submitted for assessment in any academic capacity and not copied/plagiarized in part/whole work of other persons. In case of any violation and negligence, I shall be fully responsible to face any kind of consequences in terms of grades.

Member	**Name of Student**	**Signature**
1	Abubakar Saeed	

Dated: 17-07-2023

Project Introduction

Background/Need of Project

The quiz game project aimed at developing an interactive quiz game that tests the knowledge of players on various topics. The quiz game will feature multiple questions, and players will earn points for correct answers.

Goals & Objectives

- To test and enhance player's knowledge in various subjects, including Calculus, PF, DLD, Discrete Math and Pak Study.
- The game will store and display the high scores of players in order.
- An immersive quiz experience with three dynamic difficulty levels.
- Players will earn points for correct answers, and their result card will be displayed at the end of each round and their result will also be generated in txt file with player name.

Scope of Project

The scope of the quiz game project includes developing an interactive quiz game that tests player's knowledge on various topics, implementing a user-friendly interface, and providing feedback and scoring to players.

Timeframe

	Description of Work (As per four deliverables)	Start and End dates
Phase 1	Algorithm, flowchart and implemented interface module, research for question loading Module.	10/5/23-21/5/23
Phase 2	Algorithm, flowchart of question loading module	21/5/23-28/5/23
Phase 3	Implemented question loading module, research for gameplay module, player information module	28/5/23-4/6/23
Phase 4	Algorithm, flowchart and implemented gameplay module, player information module	4/6/23-25/6/23
Phase 5	Algorithm, flowchart and implemented quiz difficulty module, high score module, result generating module	25/6/23-12/7/23

Target Users

The target users for the quiz game project are anyone who enjoys playing quiz games and wants to test and enhance their knowledge in various subjects.

Endorsements

Hafiz Muhammad Jalees, Lecturer in Psychology, Government Graduate College, Shorkot City.

Mr. Muhammad Hussain Simab, Senior teacher in Government of Special Education Faisalabad.

Saleem Akhtar, Lecturer in English, Higher Education Department. PhD Scholar TUF University.

Project Documentation

1. Algorithms

(Quiz Game Interface Module)

```
1. Start
2. Declare choice
3. do
    3.1. Display menu
    3.2. Input choice
        3.2.1. Case 1
            3.1.2.1. Display subject list
            3.1.2.2. Input choice
            3.1.2.3. Start various subject quiz based on user choice such as 1, 2, 3 …
        3.2.2. Case 2
            3.2.2.1. Display high score
        3.2.3. Case 3
            3.2.3.1. End quiz
4. while (choice!=3)
6. end dowhile
7. End
```

(Question Loading Module)

```
1. Start
2. Declare questions, options, correctOptions, questionNumber, line, i=0
3. Read the question data from file
4. while (line in question data from file)
    4.1. question.add (line)
    4.2. while (i<4)
        4.2.1. options.add (line)
        4.2.2. i++
    4.3. endwhile
    4.4. correctOptions.add (line)
    4.5. questionNumber++
5. endwhile
6. End
```

(Gameplay Module)

```
1. Start
2. Declare userOption, score, i
3. Read the correctOption and questionNumber from question loading module
4. randomShuffle (questionNumber)
5. Initialize i ← 0
6. for (i<questionNumber)
    6.1. Display question, option
    6.2. do
        6.2.1. Input userOption
        6.2.2. If (userOption < 'A' || userOption > 'D') then
            6.2.2.1. Print “invalid input please try again”
    6.3. while (userOption < 'A' || userOption > 'D')
    6.4. end dowhile
    6.5. If (userOption==correctOption) then
        6.5.1. Print “correctAnswer”
        6.5.2. score++
    6.6. else
        6.6.1. Print “incorrect Answer”
    6.7. endIf
    6.8. i++
7. endfor
8. Display result card
9. End
```

(Player Information Module)

```
1. Start
2. Declare playerName, rollNo, isAlphabetic, isRollInteger
3. Initialize isAlphabetic, isRollInteger ← false
4. do
    4.1. Input playerName
    4.2. If (playerName contain integers) then
        4.2.1. isAlphabetic ← false
        4.2.2. Print “invalid input”
    4.3. else
```

```
            4.3.1. isAlphabetic ← true
        4.4. endIf
5. while (!isAlphabetic)
6. end dowhile
7. do
        7.1. Input rollNo
        7.2. If (rollNo contain alphabets) then
            7.2.1. isRollInteger ← false
            7.2.2. Print “invalid input”
        7.3. else
            7.3.1. isRollInteger ← true
        7.4. endIf
8. while (!isRollInteger)
9. end dowhile
10. End
```

(High Score Module)

```
1. Start
2. Declare playerScores, playerRecord, line, highScores
3. Save player result in highScoreFile
4. Read result from highScoreFile
5. while (line in highScoreFile)
        5.1. playerRecord$_0$ ← line
        5.2. playerRecord$_1$ ← line
        5.3. playerRecord$_2$ ← line
        5.4. playerRecord$_3$ ← line
        5.5. playerRecord$_4$ ← line
        5.6. playerRecord$_5$ ← line
        5.7. playerScores.add (playerRecord)
6. endwhile
7. If (playerRecord$_3$ >score in data file) then
        7.1. highScores ← playerRecord$_3$
8. endIf
9. Display playerScores, highScores (in descending order)
10. End
```

(Quiz Difficulty Module)

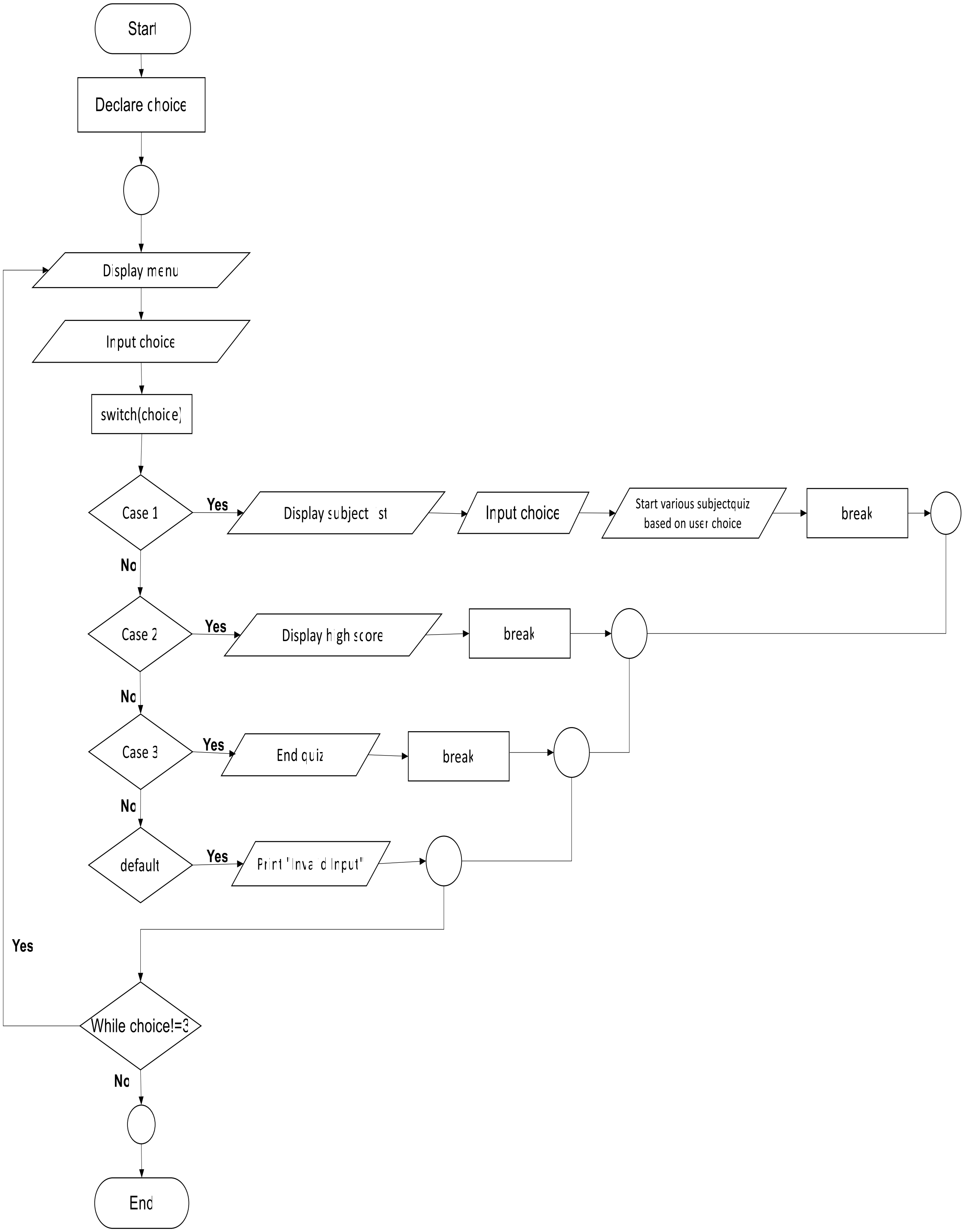
```
1. Start
2. Declare difficultyChoice, fileName, quizDifficulty
3. do
    3.1. Display difficulty menu
    3.2. Input difficultyChoice
        3.2.1. Case 1
            3.2.1.1. fileName ← (Read quiz file from easy name folder)
            3.2.1.2. quizDifficulty ← Easy
        3.2.2. Case 2
            3.2.2.1. fileName ← (Read quiz file from medium name folder)
            3.2.2.2. quizDifficulty ← Medium
        3.2.3. Case 3
            3.2.3.1. fileName ← (Read quiz file from hard name folder)
            3.2.3.2. quizDifficulty ← Hard
        3.2.4. Case 4
            3.2.4.1. Return to subject selection
4. while (choice!=4)
5. end dowhile
6. End
```

(Result Generating Module)

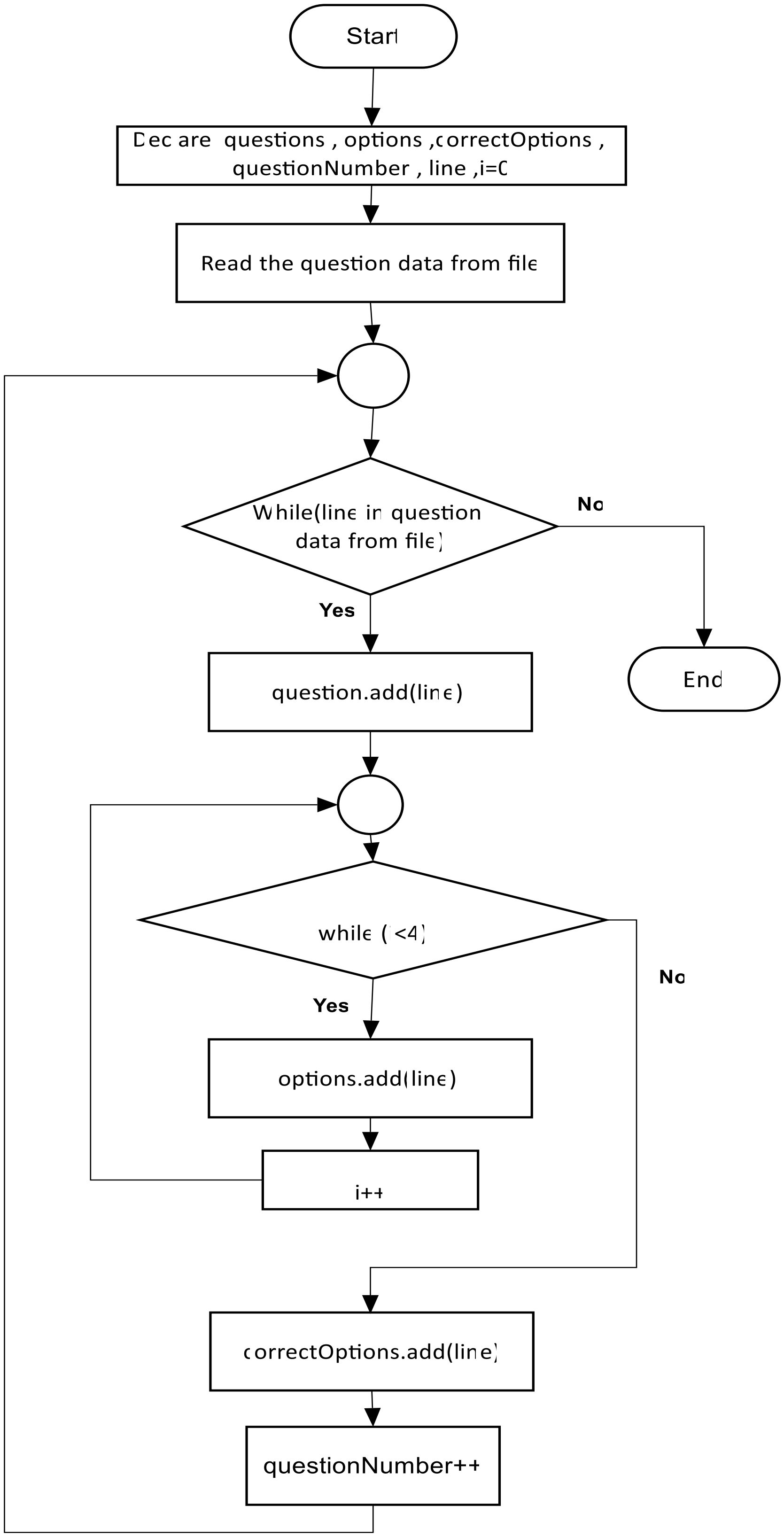
```
1. Start
2. Declare studentResult
3. Read playerInformation, score, questionNumber, quizDifficulty and quizSubject from
   respective modules
4. studentResult ← Set fileName with playerName
5. Save result card in file
6. End
```

2. Flowcharts

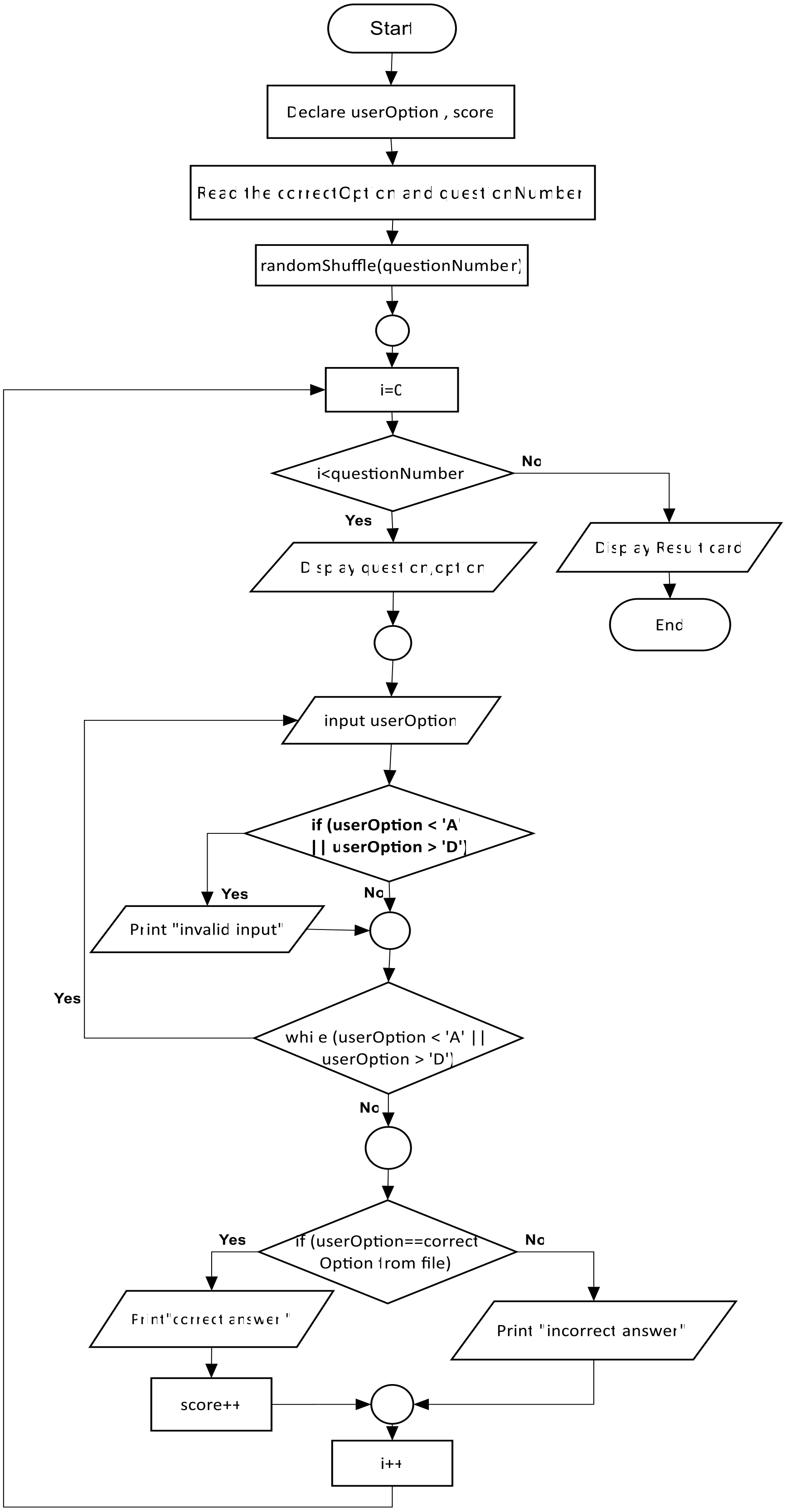
(Quiz Game Interface Module)



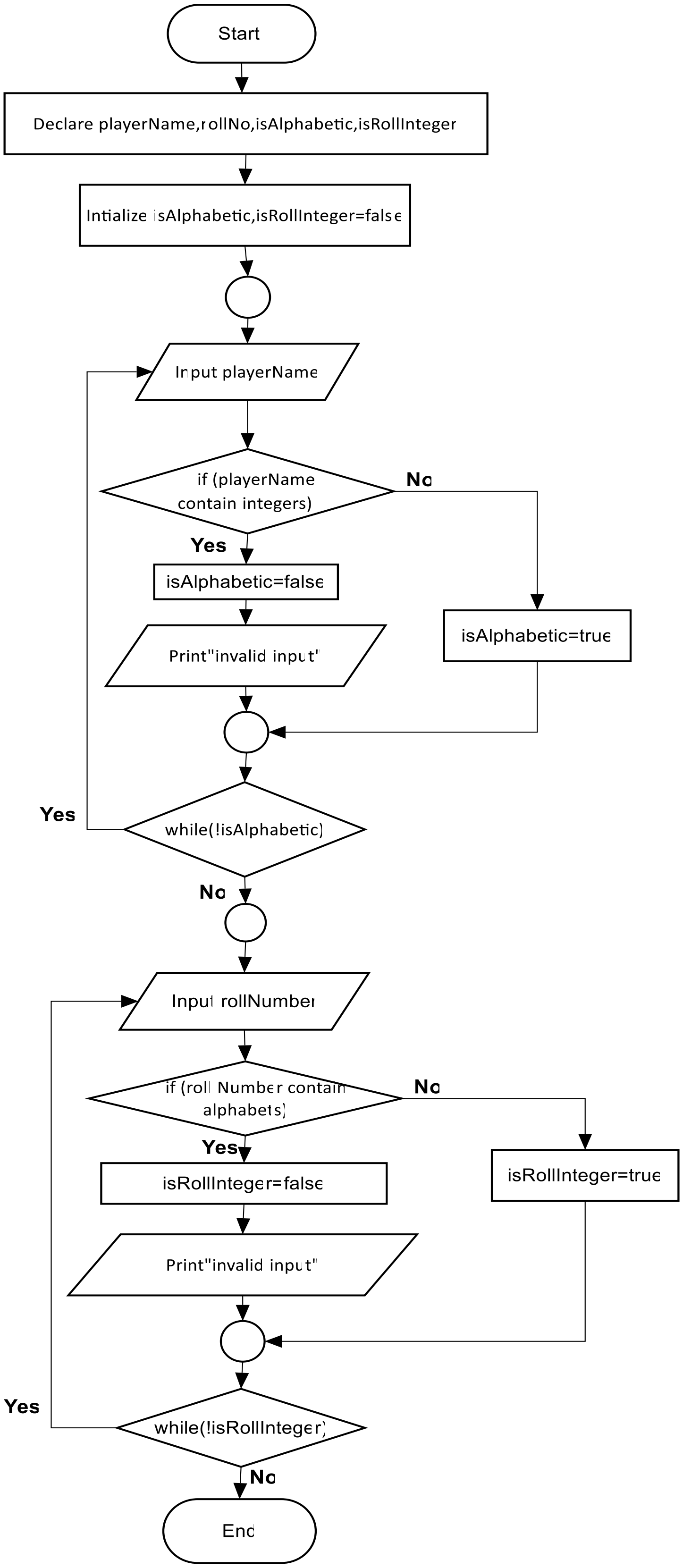
(Question Loading Module)


Start
Declare questions , options ,correctOptions , questionNumber , line ,i=0
Read the question data from file
While(line in question data from file)
No
Yes
End
question.add(line)
while (<4)
Yes
No
options.add(line)
i++
correctOptions.add(line)
questionNumber++

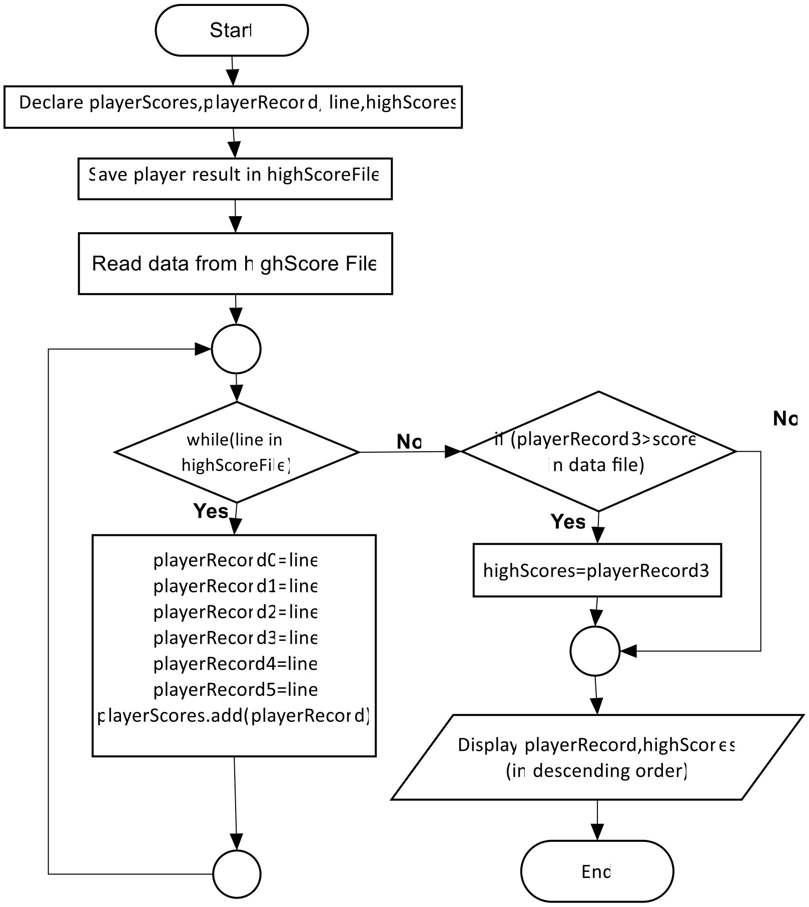
(Gameplay Module)



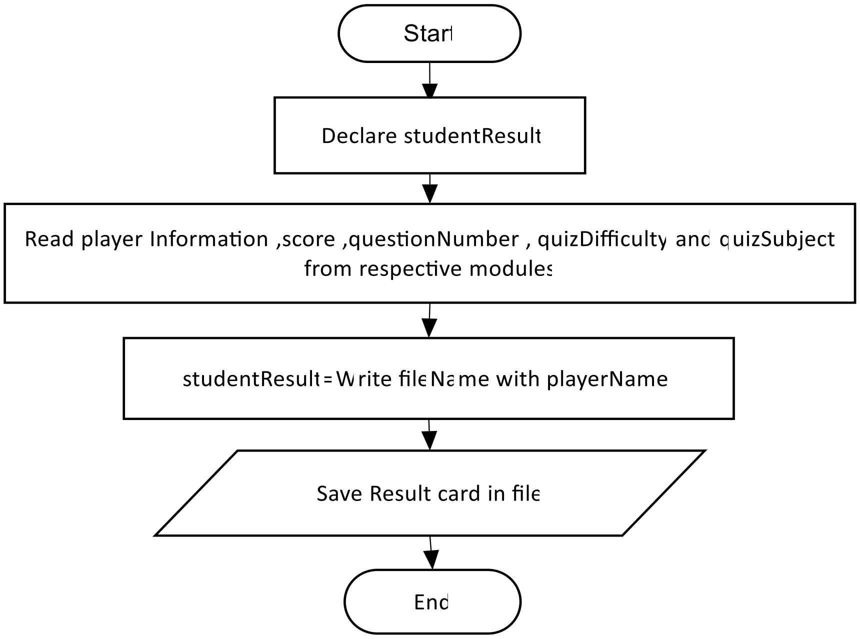
(Player Information Module)



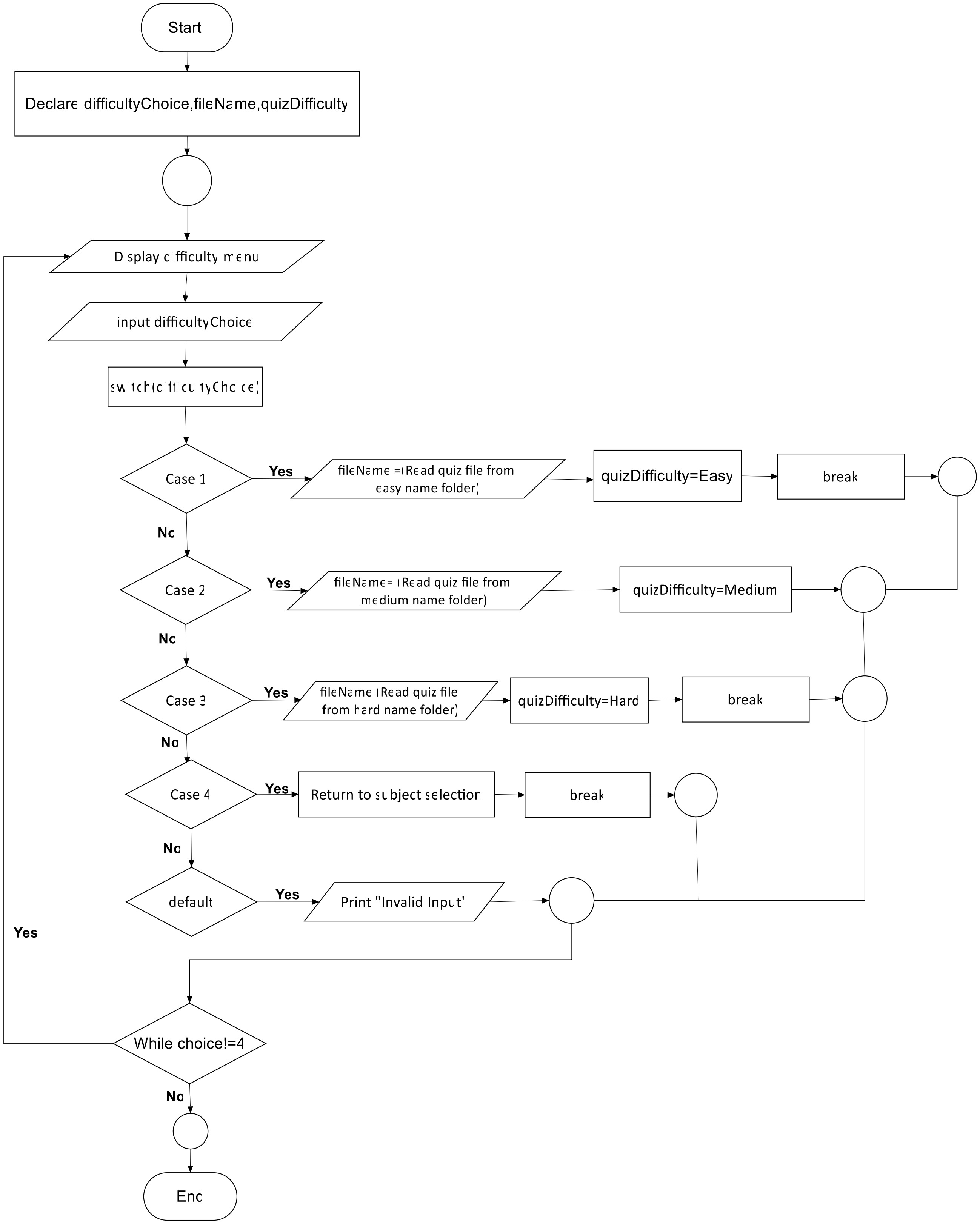
(High Score Module)


Start
Declare playerScores,playerRecord, line,highScores
Save player result in highScoreFile
Read data from highScore File
while(line in highScoreFile)
No
if (playerRecord3>score in data file)
No
Yes
playerRecord0=line
playerRecord1=line
playerRecord2=line
playerRecord3=line
playerRecord4=line
playerRecord5=line
playerScores.add(playerRecord)
Yes
highScores=playerRecord3
Display playerRecord,highScores (in descending order)
End

(Result Generating Module)


Start
Declare studentResult
Read player Information ,score ,questionNumber , quizDifficulty and quizSubject from respective modules
studentResult=Write fileName with playerName
Save Result card in file
End

(Quiz Difficulty Module)



3. Implementation

Code has been send via email.

4. Discussion

The goal of the quiz game project is to provide an engaging and educational experience for players, customizable difficulty levels, and a comprehensive result card report generated in a file, challenging their knowledge and promoting learning in an enjoyable manner.

5. Limitation

The limitations to deploy this project include factors such as platform compatibility (ensuring the game works on multiple devices and operating systems). Additionally, resource constraints, such as time, budget, may also impact the deployment of the project.

6. Future Work

In the future, we will improve the performance of quiz game project by optimizing the code, implementing exception handling, we'll ensure efficient execution, minimize resource consumption and adding extra features to enhance the user experience.