School Quiz Program

Charlie Cheung

Submission Date: xx/xx/xxxx

Ccc15@aber.ac.uk

Table of Contents

[Introduction 2](#_Toc162700486)

[Analysing The Problem 2](#_Toc162700487)

[Use Case Diagram 2](#_Toc162700488)

[Design 3](#_Toc162700489)

[Class Diagram 3](#_Toc162700490)

[Initial Class Diagram 3](#_Toc162700491)

[Final Class Diagram 3](#_Toc162700492)

[SchoolQuizApplication 4](#_Toc162700493)

[Testing 5](#_Toc162700494)

[Test Table 5](#_Toc162700495)

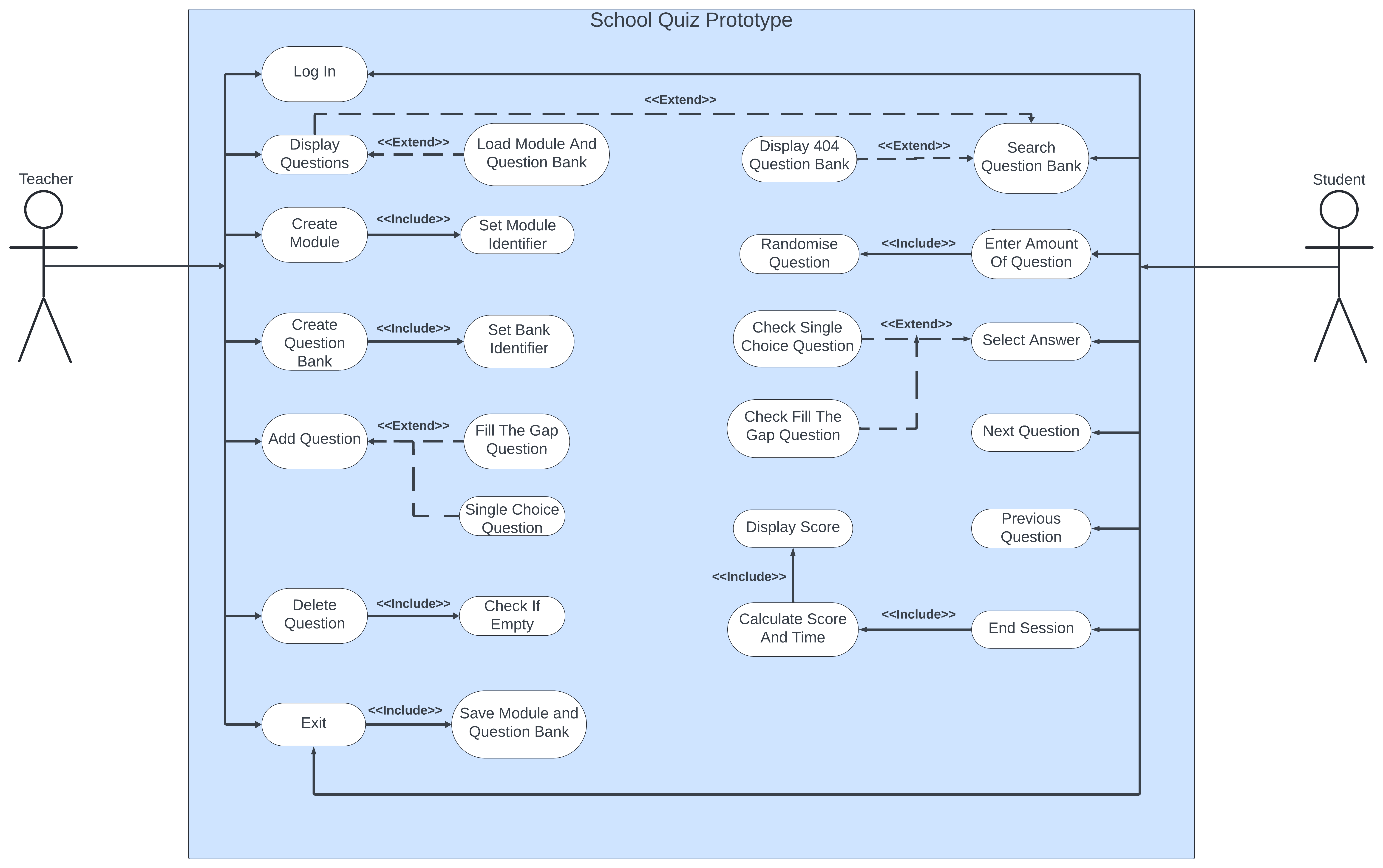
[Evaluation 5](#_Toc162700496)

# Introduction

## Analysing The Problem

For my assignment I was briefed, to create a prototype quiz program for teachers to create sets of questions banks which are associated with a specific module, questions such as a picking the correct answer from a set of potential answers and fill in the blanks of a sentence. Which are designed to evaluate students’ knowledge of a module. The program will be interacted through a text-based menu user interface, designed for teachers and students to navigate the program. Teachers should be able to edit/remove questions banks and modules. Students can partake a quiz from a question bank by searching a specific module identifier, created by the teacher. After finishing the quiz or quitting at any point, the student will be displayed a score of how well they performed. Questions will be saved and loaded from files to allow quizzes to be saved and use later when rebooting the quiz program. Other features will be implemented such as questions will be given at random and students can move back and forth of questions they are doing in a quiz.

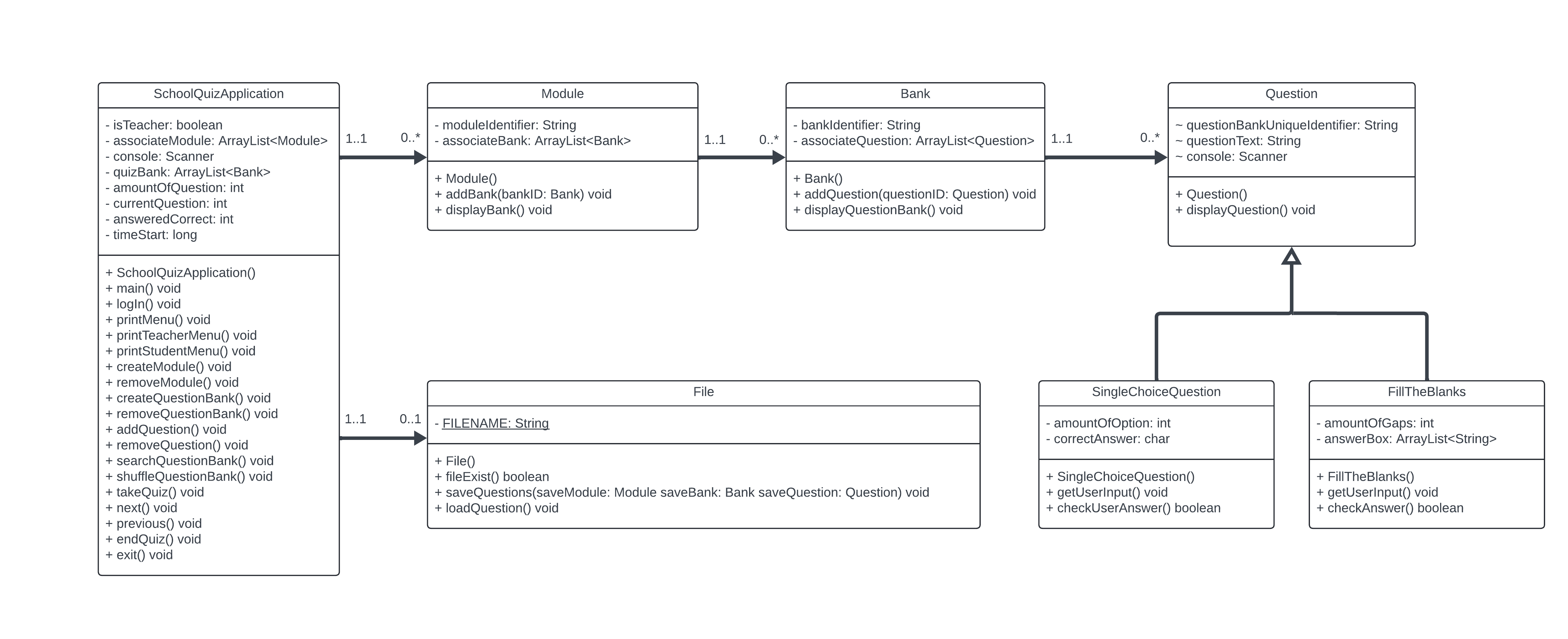
## Use Case Diagram

For my case diagram there will be two actors. A teacher and a student where the teacher can create the quiz and the student can partake the quiz. There will be one system which is the school quiz protype.  


# Design

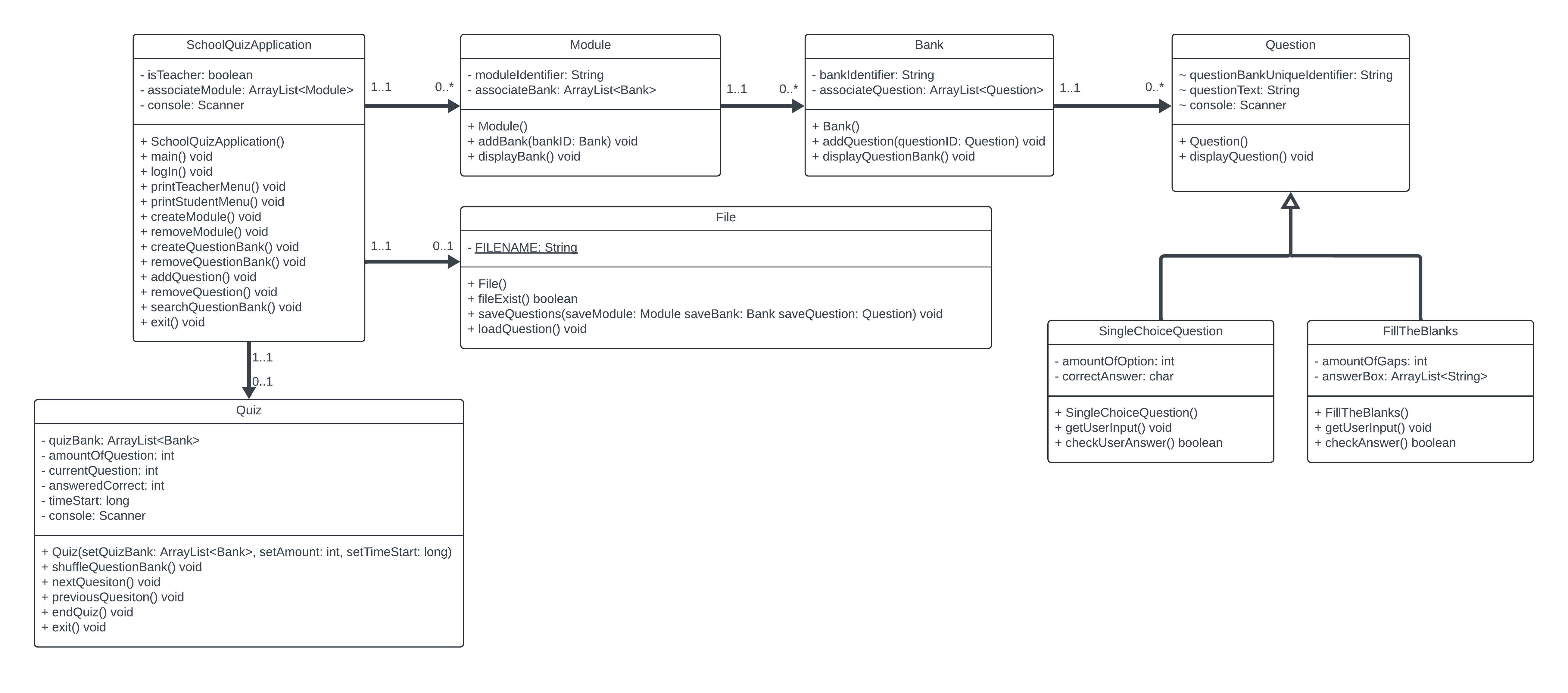
## Class Diagram

### Initial Class Diagram



Before I started creating my program, I’ve made a class diagram to give an insight of how my program will be structured and how my program will work. The code that I will be creating will follow this structure, but it is subject to change over time as there will be oversights to predicting what methods and attributes needed for the final program. Beneath will display an updated class diagram.

### Final Class Diagram



### SchoolQuizApplication

SchoolQuizApplication class is what the user will interact with, majority of the time, it will be designed in mind to display the textual user interface for the user to navigate through the program therefore it will also be the hub.

SchoolQuizApplication() – Constructor to instantiate the main program

main() – Used to start the program

logIn() – Used to identify the user as a teacher or student and give them corresponding print menu function, for example a teacher user will be displayed a teacher menu (printTeacherMenu()) and students will be displayed a student menu (StudentMenu())

printTeacherMenu() – Display a menu to the teacher with options such as create/remove module, question bank and question. Program will then check user input and run the picked option.

printStudentMenu() – Display a menu to the student with options such as searching for a question bank via a specific module identifier or changing user type to teacher or exiting the program.

createModule() – Function used to instantiate a module object with a module identifier string unique of its kind and be 7 alphanumeric long in a specific format.

removeModule() – Function which remove a module object from the associateModule ArrayList which holds all the module.

createQuestionBank() – Function used to instantiate a question bank object which is tied to a specific module identifier, also having its own unique bank identifier with maximum of 15 alphanumeric long.

removeQuestionBank() – Function remove the question bank from the associateBank ArrayList only if its empty.

addQuestion() – Function which instantiate one of the two objects, a SingleChoiceQuestion or a FillTheBlanks object which will inherit from the parent class (Question). The type of object depends what question type the teacher wants.

removeQuestion() – Function used to remove a question object from the associateBank ArrayList which holds all question from a specific question bank.

searchQuestionBank() – Function designed for the student to search for a specific question bank, in order to start a quiz session. In order to start a question bank unique identifier would be needed, which is the module identifier and bank identifier combined. After which it will instantiate the QuizSession class, passing the question bank as an argument.

exit() – Function which is used to exit program.

### QuizSession

QuizSession is a class designed for the student when they want to take a quiz. I’ve decided to have the QuizSession separate from the SchoolQuizApplication because the quiz attributes and methods doesn’t get used until after the student searches for the question bank.

QuizSession() – Constructor to instantiate the quiz. A argument will be needed from the SearchQuestionBank(), which will be stored temporarily in the quizBank object. It will also ask the user the amount of question they desire and starts the timer of the quiz.

shuffleQuizBank() – Will grab the quizBank object and will use a .shuffle() command which will randomise the order of questions.

nextQuestion() – Student can choose to move the to next question by plus one of the element of the quizBank object as long it doesn’t skip the current question which is the question waiting for the student to answer.

previousQuestion() – Student can move back to any answered question by minus one of the element of the quizBank object.

endQuiz() – Student will be shown the time took, score percentage and the amount of unanswered question, after which it will end the session of the quiz. Can be used at any time of the session.

### Module

Module class is designed to have a specific module identifier and have an associateBank ArrayList used to tie zero to many bank object to itself.

Module() – Constructor to instantiate the module object with it own unique ID and give it a blank ArrayList.

addBank() – Pass a bank object argument into the function which adds the bank object in associateBank ArrayList which links all question bank to the module.

displayModule() – Used to display all question banks via its bank identifier within the module object.

### Bank

Bank is a class designed to hold question objects which are tied to the bank identifier.

Bank() – Constructor to instantiate the bank object, designed to hold an ArrayList of question objects and set the bank identifier.

addQuestion() – Pass an argument of a question object which will add the object to associateQuestion ArrayList.

displayBank() – Displays all questions identifiers that is within the ArrayList of associateQuestion

### Question

Question is a superclass which acts as a foundation to all types of question should have a question Text

Question() – Constructor to instantiate the question object with a unique identifier and question itself.

### SingleChoiceQuestion

SingleChoiceQuestion() – Constructor which inherits from question class.

showQuestion() – Show the question in a class specific format.

getuserInput() – Gets user input from the question.

checkUserInput() – Checks if the user input matches with the answer.

### FilleTheBlanks

FillTheBlanks() – Constructor which inherits from the question class.

showQuestion() – Show the question.

getUserInput() – Get user input.

checkuserAnswer() – Checks if user input matches with answer.

### File

File() – Constructor to instantiate the file object which contains the file name.

fileExist() – Checks whether the a file save already exist with the same name.

saveQuestion() – Saves all module, bank and questions objects into a file.

loadQuestion – Load all module, bank and question into an object.

# Testing

## Test Table

# Evaluation

qinggyun