Software Requirements Specification

for

Quiz Generator

Version 1.0 approved

Prepared by: <Authors>

<Team name>

10/3/2018

Table of Contents

Table of Contents ... ii
Revision History ... ii
1. Introduction ... 1
1.1 Purpose ... 1
1.2 Document Conventions ... 1
1.3 Intended Audience and Reading ... 1
1.4 Product Scope ... 1
1.5 References ... 2
2. Overall Description ... 2
2.1 Product Perspective ... 2
2.2 Product Functions ... 2
2.3 User Classes and Characteristics ... 2
2.4 Operating Environment ... 3
2.5 Design and Implementation Constraints ... 3
2.6 User Documentation ... 3
2.7 Assumptions and Dependencies ... 3
3. External Interface Requirements ... 3
3.1 User Interfaces ... 3
3.2 Hardware Interfaces ... 4
3.3 Software Interfaces ... 4
3.4 Communications Interfaces ... 4
4. System Features ... 4
4.1 Create and Display Quizzes ... 4
4.2 Answer Quiz Questions ... 5
4.3 Display Quiz Grades ... 5
4.4 Retake Previous Quizzes ... 6
4.5 Prompt to Take More Quizzes ... 6
4.6 Store Quiz Scores ... 7
4.7 Delete Chosen Past Quizzes ... 7
4.8 Create Comprehensive Quizzes ... 8
4.9 Login System ... 8
5. Other Nonfunctional Requirements ... 9
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 ... 10
Appendix B: Analysis Models ... 10
Appendix C: To Be Determined List ... 11

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of Quiz Generator is to allow students to study Software Engineering course materials quickly and effectively through a computer-based tool that randomly generates repeatable quizzes for the user. We aim to have Mississippi State University students enrolled in the *Introduction to Software Engineering* course use Quiz Generator to help them prepare for quizzes, exams, and related testing material. Quiz Generator's ultimate goal is to achieve a higher average GPA for students enrolled in *Introduction to Software Engineering* course.

1.2 Document Conventions

This document was written in *Arial* with a font size of 11 for basic text. Header text is written in *Times* and is also bolded with a font size of 17. Sections 1-5, Introduction, Overall Description, External Interface Requirements, Systems Features, Other Nonfunctional Requirements, and Other Requirements, are in descending order of abstraction as this document goes over the pieces of Quiz Generator. Each section has a subsection marked by a period followed by another number, for example 1.1, 1.2, 1.3, etc. These subsections go into greater detail about a key part of the larger section to which they belong.

Keywords: Student, Learning, Studying, Software Engineering, Tool, Quiz, Textbook Quiz

1.3 Intended Audience and Reading

The intended audience for Quiz Generator is Students and Teachers who are engaged in attending *Introduction to Software Engineering* course. This Quiz Generator is also accessed by developers who are building the Quiz Generator application.

The SRS document gives a broad description of Quiz Generator. In the later section it summarizes the Overall description of Quiz Generator, External Interface Requirements, System Features and Nonfunctional Requirements. If you are interested in more detail, then the reader can view the associated sub sections of each topic.

For a full understanding of Quiz Generator, it is recommended to read the full document, but if the reader is already familiar with Quiz Generator they are free to read the document as they see necessary.

1.4 Product Scope

Quiz Generator is aimed at Mississippi State University students enrolled in *Intro to Software Engineering* course. It is to be used as a study tool to help increase comprehension of course materials and ultimately achieve a higher grade in the course.

1.5 References

No external resources have been used.

2. Overall Description

2.1 Product Perspective

The Quiz Generator is a new standalone program to meet the needs of students and instructors. Students will always need study materials, and this Quiz Generator was developed to fill the need for a study tool for software engineering students. This Quiz Generator is one-of-a-kind in that it is produced by and for software engineering students.

2.2 Product Functions

This program will take in the user’s inputs for desired chapters and the desired number of questions for each chapter. Then the program will choose the proper number of questions from the chapters and display them to the user in a random order. The program will also check the user’s input character against the character corresponding to the correct answer. The program will display whether or not the user was correct. If a user inputs an irrelevant character, the question will re-prompt the user for a valid input. In addition, the program will display the final quiz score as well as the correctness of the user’s answers for each individual question.

When creating quizzes, the program will have the ability to compare the specifications of each individual quiz to see if the user’s input matches that of a previously created quiz. If a matching quiz is found, the program will prompt the user to see if they want to take said previously created quiz.

To perform all these functions, the program requires systems for documenting user input and checking whether that input matches the correct answer. In addition, the program requires a system for saving user quiz scores, as well as a system for relating the quiz score to the quiz taken.

2.3 User Classes and Characteristics

User Class **Student:**

Characteristics:
Undergraduate college students with moderate software expertise, using the
software to study. This class is the largest and most frequent program users. They are very important to satisfy.

Permissions:
Take quizzes, view own quiz scores, manage own quiz scores

User Class **Instructor:**

Characteristics:
College instructors with technical expertise. These users will be using the software to view student progress. There will be few instructors, but at the least there will be one. Instructors will be relatively infrequent and are not the primary users of the software. They are important to satisfy.

Permissions:
View and manage all student scores, manage students' quizzes

2.4 Operating Environment

The software will operate on PC devices, on Windows 7 and newer Windows operating systems. The program will be written in Python and must have access to reading and writing files to computer memory. The program will run with a software library called TKinter for graphical user interfaces.

2.5 Design and Implementation Constraints

The design and implementation of the software solution will be constrained by the limitations of the Python coding language as well as the associated TKinter library. All data must be able stored and read as plain text from a file, limiting the types of characters and the complexity of the data stored.

2.6 User Documentation

A written guide detailing step-by-step user interaction with the software will be provided within the software itself. A separate document will be provided to instructors, as their interactions with the software will be different from the students'.

2.7 Assumptions and Dependencies

This software assumes the Python coding language has the capabilities to take, process, and catalogue user input. In addition, the software assumes the TKinter library for Python will provide sufficient capability and control in creating a graphical user interface. The software is dependent on the ability to store and access a text file on the computer's memory, and that this file will only be altered by the program.

3. External Interface Requirements

3.1 User Interfaces

The user interface will have a logout button present on the top right corner of the window, this button sends the user to the login screen. There will also be a home button on the top left corner of the window, this button sends the user to the quiz creation screen. If there is too much content to be displayed on screen, a scroll bar on the right side will be used to navigate down the page. These user interfaces will be used to log into the program, select quiz attributes to be used when creating the quiz, taking the quiz, and viewing past quizzes taken.

3.2 Hardware Interfaces

There will be no need for additional hardware interfaces. The Windows operating system will handle all interfacing with keyboard, mice, and computer memory devices.

3.3 Software Interfaces

The program will run in a Windows application window. The only interaction between the program and the Windows operating system is through the use of text files on the computer's memory to serve as data storage and recall.

3.4 Communications Interfaces

There are no necessary communications interfaces for this program. Operations all take place locally on the machine.

4. System Features

4.1 Create and Display Quizzes

4.1.1 Description and Priority

This feature will take in the user's inputs for desired chapters and the desired number of questions, then choose the proper number of questions from the chapters and display them to the user in a random order. This is of critical priority.

4.1.2 Stimulus/Response Sequences

The user will be prompted to input desired chapters by their chapter number and to input the desired number of questions to be pulled from each chapter. If a quiz with the same chapters and same number of questions had been taken in the past, the user will be prompted on if they want to take the past quiz or create a new one. Then the appropriate questions will be taken from each chapter and displayed to the user in a random order.

4.1.3 Functional Requirements

The program must have a list of questions for each chapter to pull from. If an improper chapter or number of questions is entered, the program will ask again until a proper input is entered. If the chosen number of questions does not exist for a chapter, the program will notify the user at the top of the quiz.

REQ-1: The program must be able to retrieve questions based on chapter.
REQ-2: The program must be able to check what quizzes have been taken previously.
REQ-3: If user gives a chapter that does not exist, prompt again for desired chapters.

REQ-4: If user gives an unusable number of questions, prompt again for desired number of questions.

4.2 Answer Quiz Questions

4.2.1 Description and Priority

The feature allows users to respond to displayed questions by choosing one answer out of a given set of possible answers. This is of critical priority.

4.2.2 Stimulus/Response Sequences

The user will input a character corresponding their selected answer from the set. Following the input, the correctness of their input should be displayed.

4.2.3 Functional Requirements

The program should check the user's input character against the character corresponding to the correct answer. The program should display whether or not the user was correct. If a user inputs an irrelevant character, the question should re-prompt the user for a valid input.

REQ-1: If the user inputs a relevant character, the correctness of the chosen answer should be displayed.
REQ-2: If the user inputs an irrelevant character, the user should be re-prompted for valid input.

4.3 Display Quiz Grades

4.3.1 Description and Priority

The program should display the user's percentage of correct questions and display the correctness of their input for each question. This is of critical priority. Quiz scores should be stored for future access. This is of high priority.

4.3.2 Stimulus/Response Sequences

Upon the user's quiz completion, the percentage of correct answers versus incorrect answers will be displayed, and the user will then be able to view the correctness of their answers for each question individually.

4.3.3 Functional Requirements

The program requires systems for documenting user input and checking whether that input matches the correct answer. In addition, the program must display the final quiz score as well as the correctness of the user's answers for each individual question.

REQ-1: If the user completes their quiz, the program should display final quiz score.
REQ-2: If the user completes their quiz, the program should display the correctness of the user's answer to each individual question as well as the answer to that question.
REQ-3: The program must be able to store grades associated with quizzes.

4.4 Retake Previous Quizzes

4.4.1 Description and Priority

The user will be presented the option to look at a list of previously taken quizzes. This is of high priority. When a user inputs quiz specifications which apply to a previously created quiz, the program should prompt the user to take said previously created quiz. This feature is of critical priority.

4.4.2 Stimulus/Response Sequences

The program must be able to store past quizzes. The program must be able to take input from the user to check the user's desired chapters and number of questions for that chapter. In addition, the program must have the ability to compare the specifications of each individual quiz to see if the user's input matches that of a previously created quiz. In addition, the program must be able to take input from the user deciding whether to take that previously created quiz.

4.4.3 Functional Requirements

The program must have a list of questions for each chapter to pull from. If an improper chapter or number of questions is entered, the program will ask again until a proper input is entered. If the chosen number of questions does not exist for a chapter, the program will notify the user at the top of the quiz.

REQ-1: If the user inputs specifications of a previously created quiz, they should be able to take same quiz.
REQ-2: If the user's input matches a previously created quiz, they should be able to take a newly created quiz instead.
REQ-3: The program must be able to store past quizzes taken.

4.5 Prompt to Take More Quizzes

4.5.1 Description and Priority

This feature will take in the user's input if they want to take another quiz or stop taking quizzes. This feature will also ask users if they want to take a quiz from the same chapter or specify a new quiz. This is of high priority.

4.5.2 Stimulus/Response Sequences

The user will be prompted to input if they want to stop taking quizzes, if selected the program ends. The user will be prompted to input if they want to retake the quiz or take a new quiz. If a new quiz is chosen, the user is sent to the main screen. If the quiz is retaken, a quiz over the same chapters will be displayed.

4.5.3 Functional Requirements

The program must prompt the user if the user will retake the quiz or not. The program must be able to display a quiz or able to redirect back to the main screen. If the user input is improper then the application should give error message and ask user for valid input.

REQ-1: The user will be asked if they want to take another quiz.
REQ-2: If the user inputs an invalid response then error message will be displayed.
REQ-3: The program must be able to redirect to the main screen.

4.6 Store Quiz Scores

4.6.1 Description and Priority

When users finish their quiz, the score to that quiz will be stored for future access. This is of high priority.

4.6.2 Stimulus/Response Sequences

When a quiz is completed, the program should automatically store the user's percentage score for said quiz.

4.6.3 Functional Requirements

This feature requires a system for saving user quiz scores, as well as a system for relating the quiz score to the quiz taken.

REQ-1: Upon completion of a quiz, the user should be notified that their score has been saved.
REQ-2: Upon completion of a quiz, the user's score should be stored and accessible in the future.

4.7 Delete Chosen Past Quizzes

4.7.1 Description and Priority

This feature will allow the user to delete stored scores for previously taken quizzes. If a quiz's scores have been deleted, the user may remove the quiz from the list of previously taken quizzes. This is of medium priority.

4.7.2 Stimulus/Response Sequences

Upon opening the program, the user should be prompted on whether they would like to manage their scores or take a quiz. When users choose to manage their scores, they should then be able to select specific quizzes and have them deleted. Once deleted, the user is prompted if they want the quiz to be removed from the list.

4.7.3 Functional Requirements

This feature requires a display of past quizzes and their scores. The feature also requires a system for selecting specific quizzes and having the associated scores removed. The program must be able to remove quizzes from the displayed past quizzes.

REQ-1: Upon opening the program, the user should have the option to manage their scores.
REQ-2: Upon the selection and deletion of a quiz score, its data should be removed from the quiz score listings.

REQ-3: Upon deletion, the user should have the option to remove the quiz.

4.8 Create Comprehensive Quizzes

4.8.1 Description and Priority

The feature of this function is to generate the quiz questions from all the chapters. The user will be displayed all the quiz from all chapters. This is medium priority.

4.8.2 Stimulus/Response Sequences

The random quiz generator will ask user if they want to take a comprehensive quiz.

4.8.3 Functional Requirements

In this function the user will be asked if they want to take comprehensive quiz.

REQ-1: This function should be able to ask user if they want to take a comprehensive quiz.

4.9 Login System

4.9.1 Description and Priority

On the program’s main screen, a login system will be displayed for course instructors. The instructor will log in using their school id and the course given password. They will be able to see the quiz scores for a chosen quiz. This is of medium priority. The login system will have students login using their school id. Submitted scores will be stored under the student’s id when viewed by instructor. This is of low priority.

4.9.2 Stimulus/Response Sequences

In this feature the user will be asked if they are a student or instructor, then prompted for the appropriate information. After logging in with id, students may take quizzes. After logging in with id and password, instructors may view quiz scores or take quizzes.

4.9.3 Functional Requirements

The function of this system is to display a login system to the user, take user’s credentials, and grant access to the quiz taking system.

REQ-1: Before opening the quiz generator the user will be prompted for type of user, then associated credentials.
REQ-2: The program must be able to distinguish which users have permissions for certain abilities.
REQ-3: The program must be able to store user credential data.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Operations within the program should prioritize efficiency to maintain program responsiveness. In addition, reads and writes should be kept to a minimum to ensure minimal delays to user interactions. Any delays that must occur should be kept under 100 milliseconds.

5.2 Safety Requirements

We aim to have the use Quiz Generator be a safe and comfortable experience. Quiz Generator does not feature any flashing graphics or other discomfort-inducing qualities. However, since Quiz Generator is a computer program it is recommended that the user be in a sitting position so as to not endanger the machine that Quiz Generator is running on.

5.3 Security Requirements

The students can login to Quiz Generator with their Mississippi State University NetId and Instructor can login by their Course Code. Once logged in students can only see their grade, but instructor will be able to see every student's score for the quizzes. Instructors will use a course password that will be provided for that class, and this password will be hidden from the students.

5.4 Software Quality Attributes

The program must be reliable. When students wish to take their choice of chapters and number of questions in the Quiz, then the program must give all the chapters and number of questions student requested. Only when a chapter does not have enough questions should the program deviate. Any student should be able to view their own quiz grade, and instructors should able to view every quiz grade without any being hidden.

The software should be available and functional when the students or the instructors wish to open it. The software should be able communicate with the database or create a new one if on a new machine.

The Quiz Generator must be robust and handle any possible incorrect inputs. The software should function no matter what inputs are entered.

5.5 Business Rules

This program will only require one user to function. For the best user experience it is asked that user not attempt to edit any files created by the program, as this may cause undesired effects such as loss of functionality.

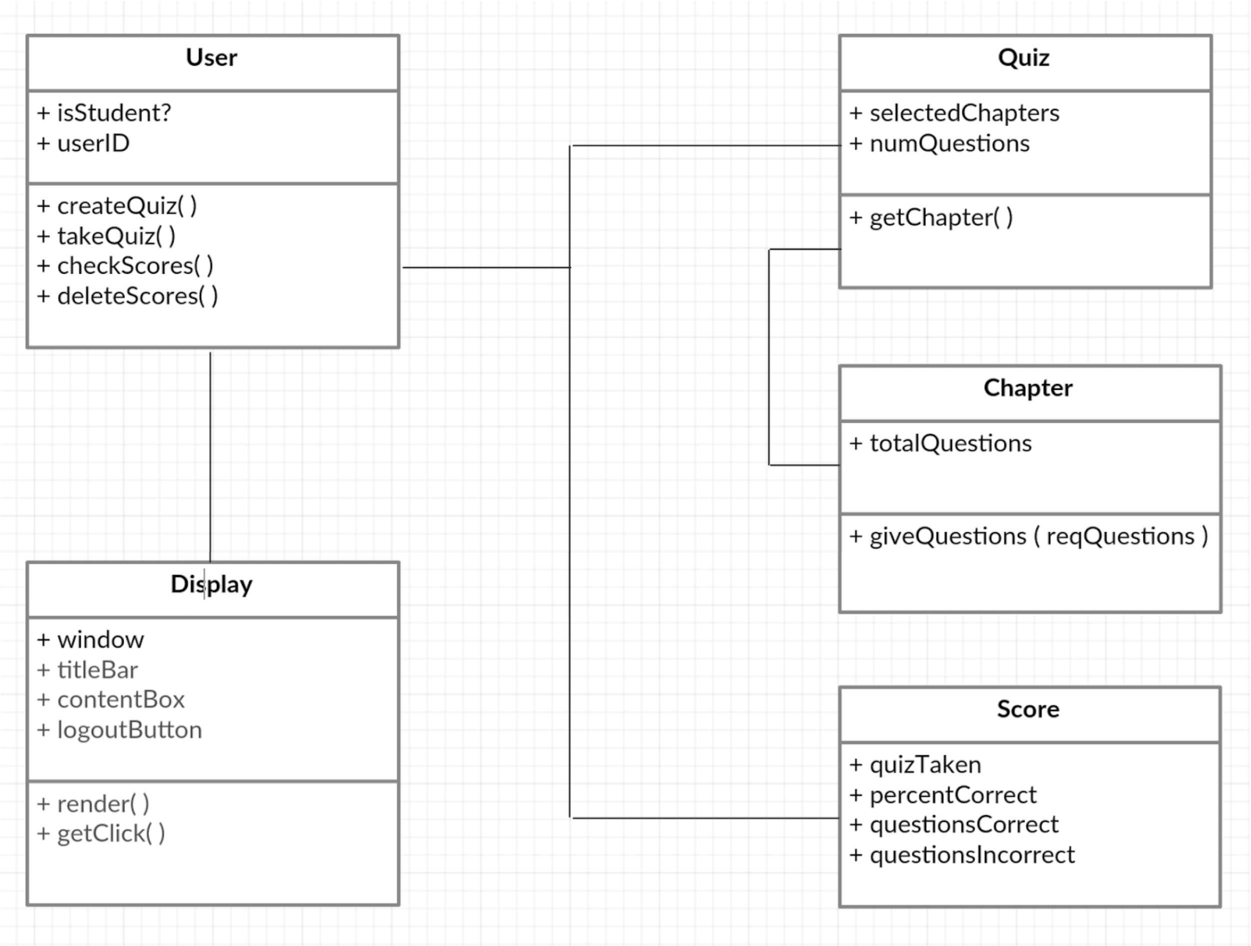
6. Other Requirements

The database must be done using a plain text file. The program will be able to create, read, and modify this document, and use these text files to persist information after the program is closed.

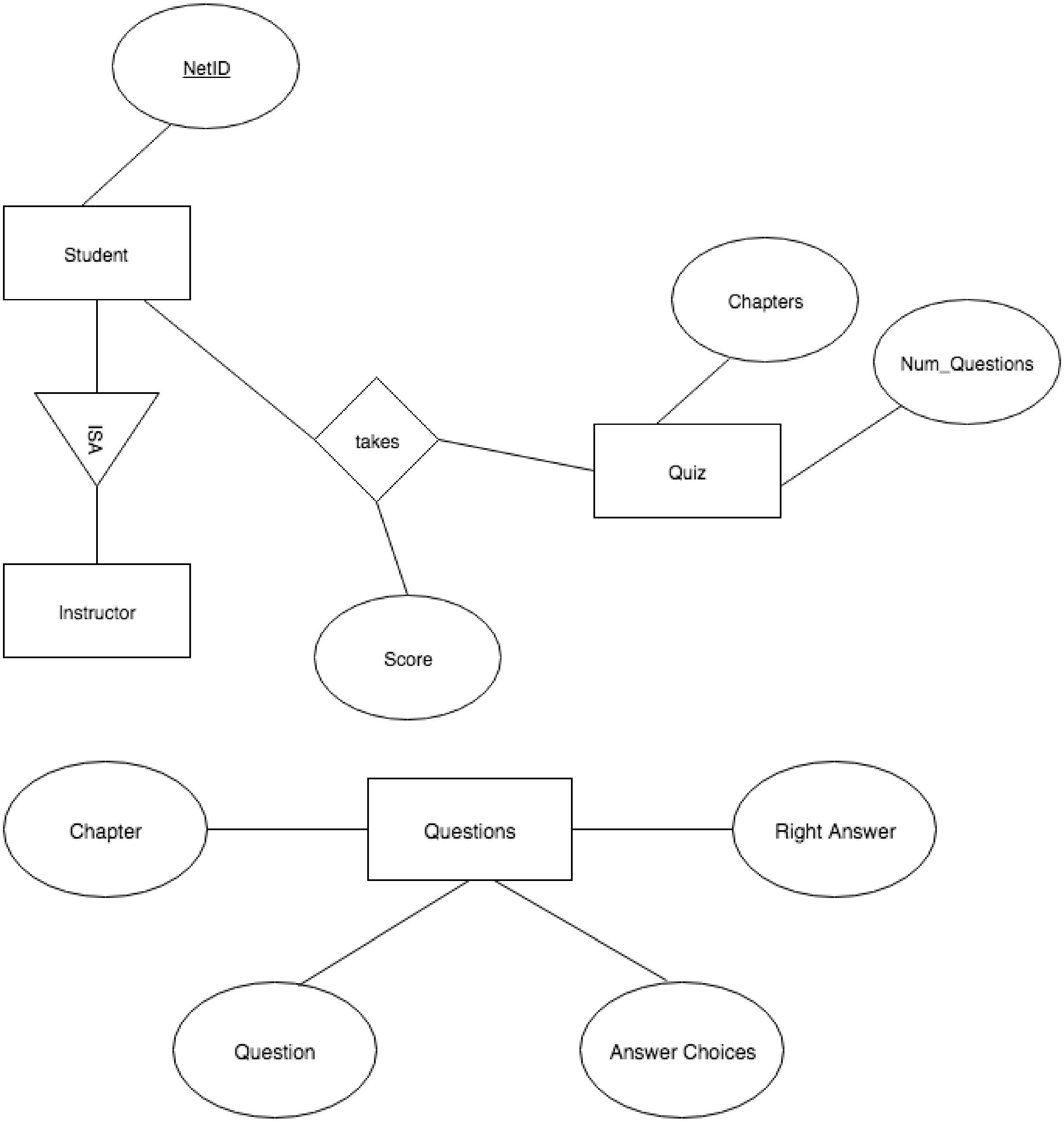
Appendix A: Glossary

Appendix B: Analysis Models

Class Diagram:



Entity-Relationship Diagrams:



Appendix C: To Be Determined List