

**Software Analysis**

***Department of Engineering***

***School of Informatics & Engineering***

***Institute of Technology, Blanchardstown***

***Dublin 15.***

**Software Design & Quality**

# 1 Introduction

# 2 Problem statement

# 3 Use Case Analysis

**USE CASE 1**

The user is presented with an interface where they are asked to input their login credentials – username and password. The system verifies that that username and password are correct and determines whether the user has access as either a student or as an administrator.

**USE CASE 2**

The student has successfully logged into the system and is presented with the option to take a quiz. When the student starts the quiz, they are then given a series of 10 randomly chosen multiple-choice questions that must be answered. Once the student has completed the quiz, the quiz report is displayed and the results are saved to the student’s profile. The student can then exit back to the student menu screen.

**USE CASE 3**

When an admin selects the option to view a student profile, they are prompted to enter the username of the student profile they wish to view. In the student profile, they can view the results of the students’ previous quiz attempts.

When the student selects the option to view the student profile, they are brought directly to their own profile only. In their own profile, they can view the results of the their previous quiz attempts.

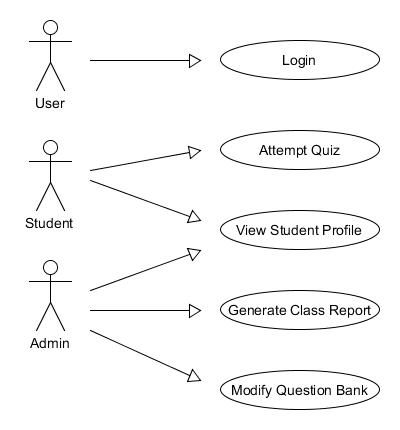
**USE CASE 4**

The admin has the option to generate a class report. When selected, the report will display the results of each student and the average of their combined grades.

**USE CASE 5**

The admin has the option to modify existing questions and add additional questions to the question bank.

## 3.1 Use Case model

****

### 3.1.1 Description

**USE CASE 1**

#### Fully-dressed Description

Primary actor: User.

Goal in context: Logging into the system.

Level: User level.

Stakeholders and Interests:

Student: wants to access their profile and take quiz.

Preconditions: The system must be able to verify the credentials.

Minimum guarantee: The user will be prompted to try again, if login attempt is unsuccessful.

Success guarantee: User has successfully logged into their system account.

Trigger: User selects the sign in button.

Main success scenario:

1. User credentials verified.
2. User logged in to correct account.

Extensions:

1. User does not enter any details, selecting the option to sign in, results in prompt to enter correct credentials.

Frequency of use: Several times per day.

**USE CASE 2**

#### Fully-dressed Description

Primary actor: Student.

Goal in context: Taking a quiz.

Level: Student level.

Stakeholders and Interests:

Student: wants to take quiz and view results.

Preconditions: Questions must be randonly generated upon start of quiz.

Minimum guarantee: Student is able to take the quiz.

Success guarantee: Student can complete the quiz and view the result.

Trigger: Student selects the option to begin quiz.

Main success scenario:

1. Student is permitted to take the quiz.
2. Student answers all 10 questions.
3. Upon quiz completion, student is presented with quiz report.
4. Student may exit back to menu screen after quiz report is displayed.

Extensions:

1. Quiz report is saved to student’s profile.

Frequency of use: Several times per day.

**USE CASE 3**

#### Fully-dressed Description

Primary actor: Admin and Student.

Goal in context: Viewing student profile.

Level: Student and Admin level.

Stakeholders and Interests:

Student: wants to view previous quiz results.

Admin: wants to view specific students’ results.

Preconditions:

1. Student must be enrolled to class.
2. Admin has to know each student’s number.

Minimum guarantee: Profiles must be accessible.

Success guarantee:

Student is able to view their own profile.

Admin can search for and view a specific student profile.

Trigger: Student and Admin select the option to view profile.

Main success scenario:

1a. Student clicks on their profile.

1b. Student is presented with their profile, including all previous quiz results.

2a. Admin clicks on student profile and is prompted to enter student number.

2b. Admin is presented with specified students profile and can view their previous quiz results.

Extensions:

1. Admin enters incorrect student details, is then promted to re-enter student details.

Frequency of use: Several times per day.

**USE CASE 4**

#### Fully-dressed Description

Primary actor: Admin.

Goal in context: Generating quiz report.

Level: Admin level.

Stakeholders and Interests:

Admin: wants to create a class report.

Preconditions: Student profile must be available to combine into class report.

Minimum guarantee: Admin has option to generate class report.

Success guarantee: Class report is generated and displayed for the Admin to view.

Trigger: Admin selects ‘create class report’ option.

Main success scenario:

1. Admin selects option to generate class report.
2. All student profiles are combined into a class report.
3. Class report is displayed.

Extensions: No students enrolled in class so report impossible to create

Frequency of use: Once per week.

**USE CASE 5**

#### Fully-dressed Description

Primary actor: Admin.

Goal in context: Admin able to add, delete and modify questions in the question bank.

Level: Admin level.

Stakeholders and Interests:

Admin: Wants to edit questions in question bank.

Preconditions: Admin must be able to access question bank.

Minimum guarantee: Admin able to access questions in the question bank.

Success guarantee: Admin able to add, delete and modify questions in the question bank. All changes made to the question bank can be saved.

Trigger: Admin selects the ‘modify questions’ option.

Main success scenario:

1. Admin selects the ‘modify questions’ option.
2. Admin is brought to the questions editing page.
3. Admin is then able to add, delete and modify questions in the question bank.
4. Admin is able to save changes made to the question bank.
5. Admin can then select the option to return to the menu screen.

Frequency of use: Several times per day.

# 4 Conceptual Model

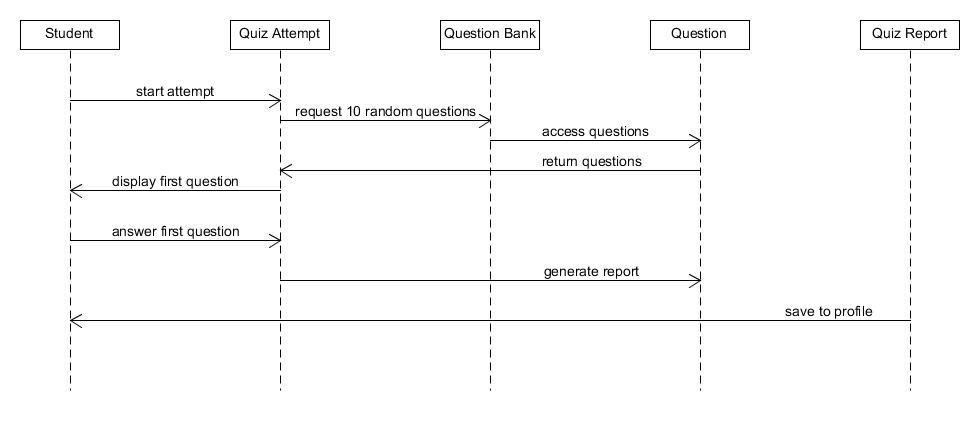
## 4.1 UML Class Conceptual Model

## 4.2 Modelling Assumptions

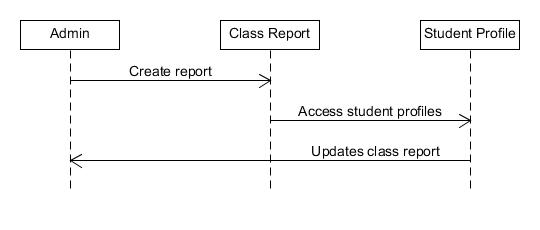
# 5 Behaviour Analysis

## Sequence diagram

* Student taking quiz

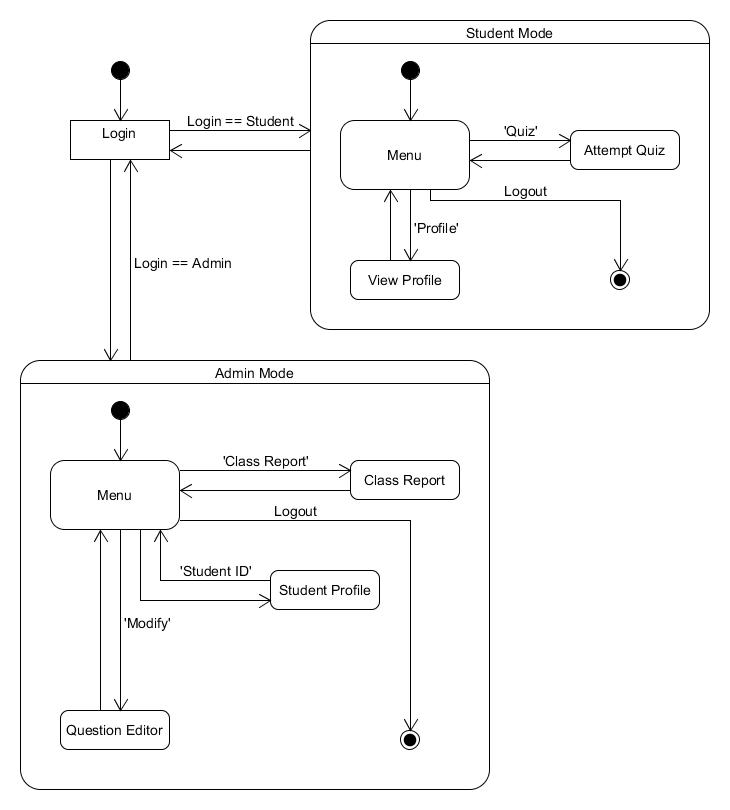


* Admin creating report



## Activity diagram

## State diagram



# 6 System Design Observations and Recommendations

# 7 Software Design

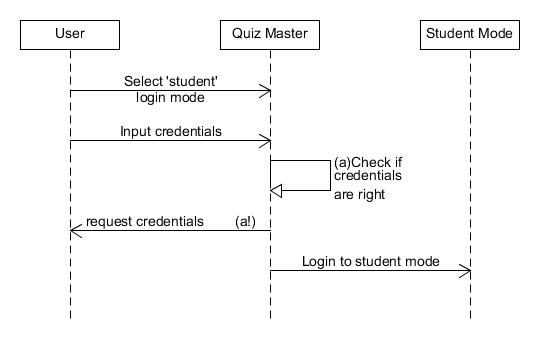
## Interaction Analysis

### Detailed Sequence Diagrams

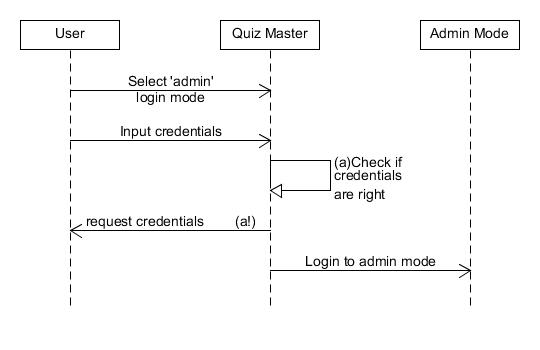
* Student taking quiz

### F:\YR3\Programming\PortableGit\3stooges\Peter\student taking quiz refined.jpg

* Student mode login



* Admin mode login



### Refined Class Diagram

### Implementation Recommendations and Notes