****

***Department of Engineering***

***School of Informatics & Engineering***

***Institute of Technology Blanchardstown***

***Dublin 15***

**Software Design & Quality**

**7th of October 2016**

# Table of Contents

[Table of Contents i](#_Toc463433223)

[Table of Figures i](#_Toc463433224)

[Introduction 1](#_Toc463433225)

[Problem Statement 1](#_Toc463433226)

[Software Analysis 1](#_Toc463433227)

[Use Case Analysis 1](#_Toc463433228)

[Fully Dressed Use Case Model 1 1](#_Toc463433229)

[Fully Dressed Use Case Model 2 1](#_Toc463433230)

[Conceptual Model 1](#_Toc463433231)

[UML Class Conceptual Model 1](#_Toc463433232)

[Modelling Assumptions 1](#_Toc463433233)

[Behaviour Analysis 1](#_Toc463433234)

[High-level sequence diagram 1](#_Toc463433235)

[High-level Activity diagram 1](#_Toc463433236)

[High-level State diagram 1](#_Toc463433237)

[Software Design 2](#_Toc463433238)

[Interaction Analysis 2](#_Toc463433239)

[Detailed Sequence Diagrams 2](#_Toc463433240)

[Refined Class Diagram 2](#_Toc463433241)

[Implementation Recommendations & Notes 2](#_Toc463433242)

# Table of Figures

[Figure 1 1](#_Toc463433941)

[Figure 2 1](#_Toc463433942)

[Figure 3 1](#_Toc463433943)

# Introduction

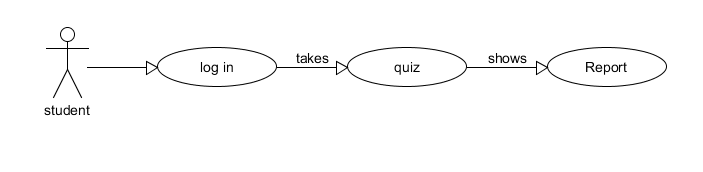
In this task we are given a problem statement. In the statement given, it explains “to design the software for a multiple Choice Quiz application to aid students taking the module “Software Design & Quality”. We are asked to design and show a simple use case analysis and a conceptual model for this problem.

# Problem Statement

You are required to develop a C++ Multiple Choice Quiz application to aid students taking the module “Software Design & Quality” understand agile software development methodologies (specifically Kanban & SCRUM) and hence study for their January exams. Users will be presented with a randomly chosen set of 10 questions from a question bank and be presented with a choice of solutions. The user must then choose a correct answer and move to the next question. The application will generate a report of how well the student did after the quiz has completed. The student’s result is saved to their profile. The application can manage many student profiles and can generate an overall class report. The quiz may be administrated by an admin user. An admin user has full access to all student’s attempts, but a student user can only see their own attempts as such Student’s must login to their profile before starting the quiz. Students are allowed have multiple attempts which are all persistently saved against their individual profile. Questions and solutions are saved in an XML file which is read by the quiz application. You will need to research Agile developmental lifecycles to define your questions and you will need to investigate how to parse XML using C++

# Software Analysis

### Fully Dressed Use Case Model 1

****

#### Fully Dressed Description

#### Primary actor: student.

**Goal in context**:

**Level**: User level

**Stakeholders and Interests**:

User: wants to take quiz and see resulting report

Quiz: Shows quiz and generates student result in report.

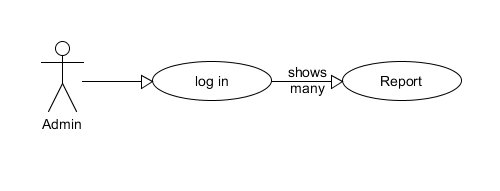
**Precondition**: Quiz is ready.

**Minimum Guarantee**: User sees quiz questions.

**Success Guarantees**: user has taken quiz and weather station is updating value to display new temperature.

**Trigger**: Student logs in.

### Fully Dressed Use Case Model 2



#### Fully Dressed Description

**Primary actor**: Admin

**Goal in context**: Admin sees all reports of student’s results from quiz.

**Level**: Admin level

**Stakeholders and Interests**:

Admin: wants to see students quiz result.

Quiz: stores student’s result’s.

**Precondition**: quiz is ready.

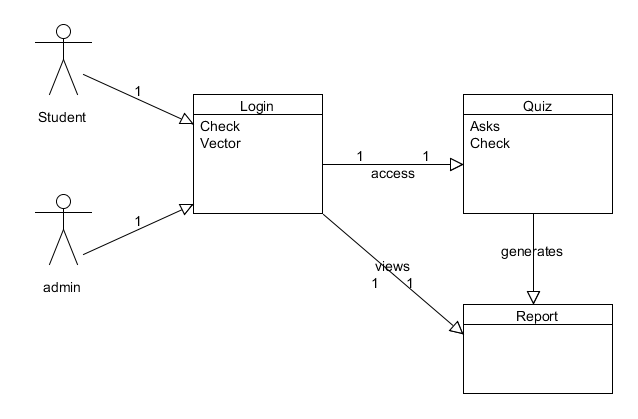
**Minimum Guarantee**: Admin sees all student’s results in report.

**Success Guarantees**: Admin has viewed report

**Trigger**: admin logins

## Conceptual Model

### UML Class Conceptual Model



### Modelling Assumptions

* Quiz class takes in questions + answers from the XML file.
* Only one student at a time using the system.
* Only one report per student after quiz.
* One report is added to the overall reports.

**System Design Observations and Recommendations**

* One student can only view their own report.

## Behaviour Analysis

### High-level sequence diagram

### High-level Activity diagram

### High-level State diagram

### High-level State diagram

# Software Design

## Interaction Analysis

### Detailed Sequence Diagrams

#### Sequence 1

#### Sequence 2

#### Sequence 3

#### Etc.

### Refined Class Diagram

### Implementation Recommendations & Notes