**COEN 275 Object-Oriented Analysis,**

**Design and Programming**

**Quiz Application**

**Requirement Analysis**

**Contents**

**1. Application Requirements ............................................................................................................. 3**

1.1 Overview of the application requirements ........................................................................................ 3

1.2 Functional workflows and steps ........................................................................................................ 3

**2. Object Oriented Features .............................................................................................................. 4**

2.1 Object Oriented Design ...................................................................................................................... 4

2.2 Inheritance ......................................................................................................................................... 4

2.3 Polymorphism ..................................................................................................................................... 4

2.4 Abstraction .......................................................................................................................................... 4

1. Application Requirements
   1. Overview of the application requirements

* Decent UI.
* Allows the user to login as user or admin.
* Leaderboard
* Show score as a percentage
* Quiz timer
* Randomize questions
* Results page after the quiz with the updated leaderboard.
* Each quiz should have at least 10 questions.
* The admin should be able to
* Perform CRUD operations with the questions.
* Change the quiz timer value.
  1. Functional workflows and steps
* To use the application, the teacher/student must first register by providing a username and password, and then be redirected to the login page to confirm that credentials have been validated and authenticated.
* To login as a student, click on student on the welcome page and type in your credentials. After clicking the login button, if the authentication is successful then the user can access the student dashboard.
* The student dashboard displays an attend quiz button that allows students to take quizzes, a view results button that displays the leaderboard and a quiz status button that shows all the quiz questions as well as a logout button at the bottom of the page.
* To login as a teacher, click on the teacher on the welcome page and type in your credentials. After clicking the login button, if the authentication is successful then the user can access the teacher dashboard.
* The teacher dashboard displays a create quiz button that allows them to create a quiz, a quiz status button that can be used to view all questions, a remove quiz button that can be used to delete the quizzes, a view result button to check the leaderboard and a logout button at the bottom.
* To login as admin the user does not have to register, they can just enter the admin details (username: admin, password: pass)
* The admin can view and edit student/teacher info, check quiz status, edit quiz and logout.

1. Object Oriented Features

2.1 Object Oriented Design

The whole design of the program is built on object-oriented concepts, where state and functionalities are logically separated into different classes and objects depending on their importance and behavior, with no functions available outside of class. When using

The Java Swing application follows an object-oriented design pattern, with all child UI components inheriting from a single parent class of the window frame that launches the Java Swing application. Different UI components exhibit different features such as inheritance, abstraction, and polymorphism.

2.2 Inheritance

Common properties from the general base class Event are transferred via inheritance to more specialized derived classes like student\_homepage, teacher\_homepage and admin\_home. These classes can then add their own set of methods and properties in addition to the inherited ones and can even override some methods depending on how they are implemented.

* 1. Polymorphism

The use of polymorphism allows for the association of suitable runtime methods with the object for which they are called. For instance, any CRUD operation methods on the Event class will call the appropriate CRUD methods on the student, teacher or any of the other classes depending on which object is triggered.

* 1. Abstraction

By exposing just an application interface through the Event class and keeping the implementation specifics in the subclass, abstraction is utilized to mask the specific implementation.