A close up of a sign

Description automatically generated

**Fundamental Java Project: Quiz Manager**

Technical Guide

By

PRASHANTH RATHINAVEL

July 04, 2019

# 

# **Introduction**

# The abbreviation of the project title is

# e-qms – Electronic-Quiz Management System

The objective of this project is to build and deliver a Quiz Management System through a console application (Eclipse IDE). This application comprises of two user types. They are,

1. Teacher/ Admin

The user will be able to login as a Teacher/ Admin and perform the following functions in this application.

* Create Question
* Update Question
* Delete Question
* Search Question

1. Student

The user will be able to login as Student and take the quiz i.e., answer the sequence of questions (open type) under each topic (created by the teacher/ admin). Upon successful completion of answering all the questions in a topic, the user will be provided with the final score.

## **Project Scope**

1. To display/ publish the open type questions under the selected topic in a spontaneous/ continuous pattern in the student menu i.e. in a sequence
2. Automatic computation of the score upon completion/ answering all the open type questions
3. Creation of Data Access Object with CRUD functions i.e. create, read, update and delete are the four basic functions
4. Create a configurable properties file for the application

## **Software Requirements**

The following software specifications are required to execute and experiment the functions of the application.

|  |  |  |
| --- | --- | --- |
| 1 | Front End | Not Available |
| 2 | Back End | Java 8 |
| 3 | Database | H2 Console |
| 4 | Version Control | GIT |
| 5 | Testing | Manual Testing |
| 6 | Tools | Eclipse IDE |
| 7 | SQL Syntax | * CREATE TABLE … TABLENAME – Create Tables * INSERT … INTO … VALUES– Insert records into tables * UPDATE … SET … WHERE - Update existing value with new values * DELETE FROM … WHERE - Delete an existing record |

# **Hardware Requirements**

The following hardware specifications are required to install and setup the software requirements to build and run the application.

|  |  |  |
| --- | --- | --- |
| 1 | Operating System | Windows, Mac OS X, Linux |
| 2 | RAM | 256 MB (min.) |
| 3 | Hard Disk | 500 MB (min.) |
| 4 | Processor | 64-bit, four-core, 2.5 GHz min. per core |

**Project Specifications**

|  |  |
| --- | --- |
| 1 | Create a list of questions in sequence (i.e. quiz) under different topics provided by the user (teacher/ admin) |
| 2 | Create tables in the h2 database to store the data values |
| 3 | The user (student) should be able to answer the questions created under the different topics |
| 4 | Automatic evaluation of open questions and displaying the computed score at the end of the quiz |
| 5 | The options for user (teacher/ admin) to create, update, delete and search the questions using the CRUD functions (create, read, update and delete) |
| 6 | The different types of questions to be used are   * Associative Questions * Multiple Choice Questions * Open Questions |

## **Acronyms and Abbreviations**

|  |  |
| --- | --- |
| **Acronyms** | **Meaning** |
| CRUD | **Create**, **read**, **update** and **delete** functions that are implemented in relational database applications |
| GUI | **Graphical User Interface** – The front end interface through which the user can interact with the application. |
| DAO | **Data Access Object** Service class for the application to communicate with the H2 database |
| UML | **Unified Modeling Language** is a standard way to visualize the design of a system |
| OQ | **Open Questions** are provided with free-text field response and without any predefined answers |
| JDBC | **Java Database Connectivity** is an application programming interface (API) for the programming language Java |

**System Specifications**

**Functional Requirements**

**Main Window:** The following code snippet while executed will display the main menu with the Teacher/ Admin and Student menu.

A screenshot of a social media post

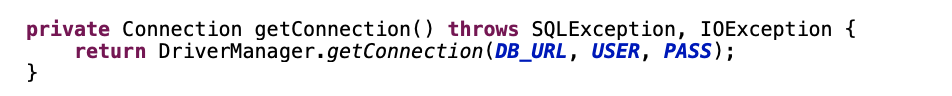
Description automatically generated

**Student Window:** The following code snippet when executed will display the sequence of questions (open type) and once the user completes answering all the questions, the respective score will be displayed.

A screenshot of a social media post

Description automatically generated

**Database Connectivity:** The following code snippet will allow the application to connect with h2 database and its respective tables.



**Functionalities of Admin Window:**

**Create Question:** The method used in the below code snippet will allow the teacher/ admin to create open type questions one by one under specific topic along with its difficulty level and answer. The records are stored in the h2 database tables.

A screenshot of a cell phone

Description automatically generated

**Update Question:** The method used in the below code snippet will allow the teacher/ admin to update the existing open type questions under respective topic. The records in the tables are updated with new values using SQL queries.

A screenshot of a cell phone

Description automatically generated

**Delete Question:** The method used in the below code snippet will allow the teacher/ admin to delete an existing open type questions under the given topic. The records in the tables are deleted using SQL queries.

A screenshot of a social media post

Description automatically generated

**Search Question:** The method used in the below code snippet will allow the teacher/ admin to search/ display all the open type questions provided the topic. The records in the tables are retrieved and displayed using SQL queries.

A screenshot of a cell phone

Description automatically generated

**Non-functional Requirements**

* The program should be able to read a configuration property set from a file on the filesystem which will avoid hardcoded parameters.
* The data values are stored in the tables (rows, columns) in the h2 database.

# **User Interface Design**

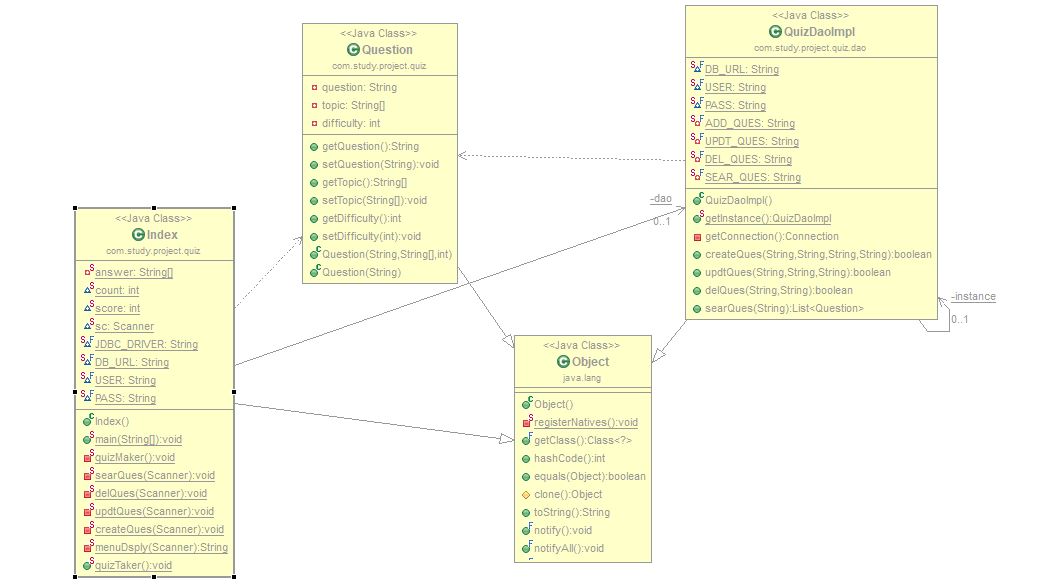
Refer to the User Guide.

**Future Enhancements**

1. Design and development of GUI using (HTML, CSS, JS and jQuery)
2. Altering the tables or Creating new tables to store and retrieve associative and multiple choice questions in the application
3. Creating sign/ signup form the different users to access the application using their credentials (username, password)
4. Options to export the results/ score in different file formats (.xlsx, .pdf, .docx)

**Appendix**

**UML CLASS DIAGRAM**



**References**

<https://thomas-broussard.fr/work/java/courses/project/fundamental.xhtml>