

# Java Fundamental Project

# Quiz Manager

## Technical Guide

Done By: Nishigandha Shirish Kulkarni  
E-mail: [Nishigandha.sk1995@gmail.com](mailto:Nishigandha.sk1995@gmail.com)



## Table Of Contents

1. Introduction.....	3
1.1 Project Description	
2. Schema.....	3
3. Operational Description.....	4
3.1 Authentication	
3.2 Login	
3.3 Create/Update/Delete the Quiz and Question	
3.4 Evaluation	
4. Configuration Instructions.....	5
4.1 Development Tools	
4.2 Database Configuration	
5. SQL Queries.....	7
6. References.....	9

## 1. Introduction

### 1.1 Project Description

Implement a Quiz Manager Application which aims at managing quiz preparation and Execution with following features:

Admin/Teacher Login:

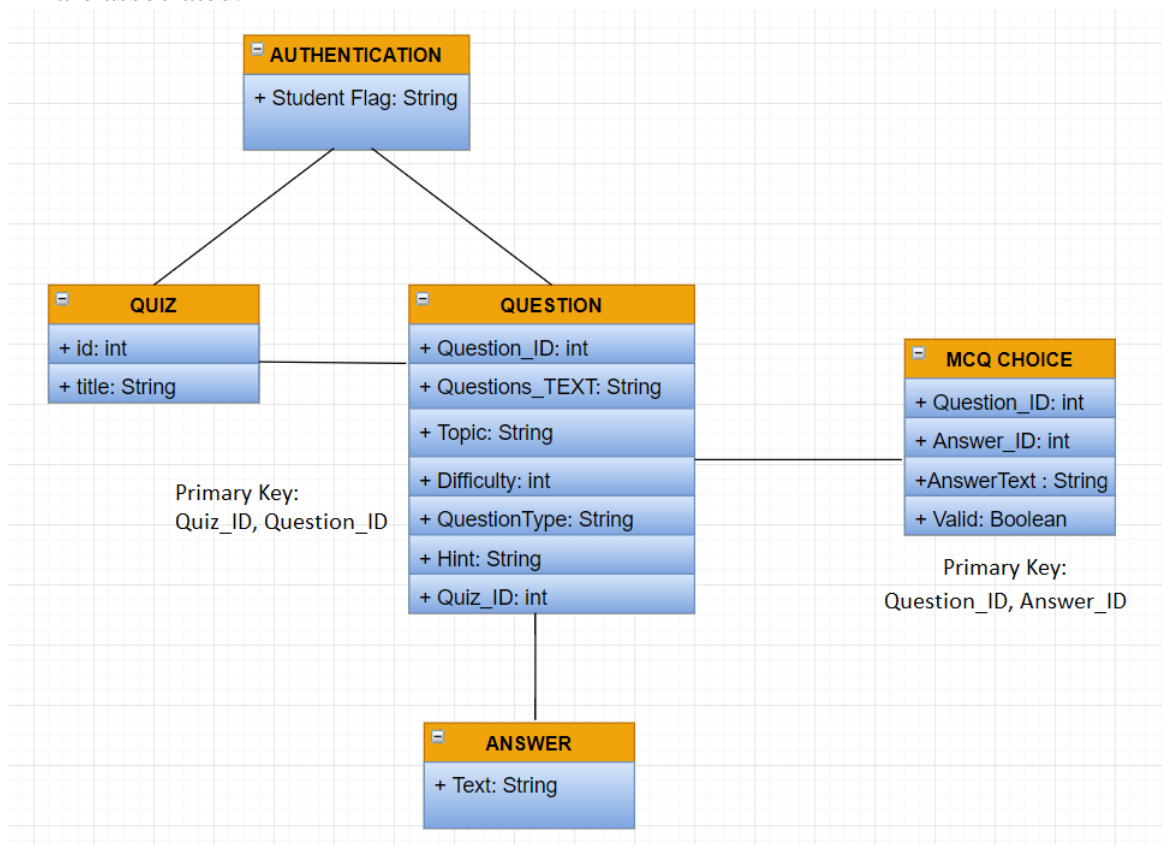
- 1 – Generating Quiz Based on Topic
- 2 – CRUD Functionality

Student Login:

- 1 – Based on the Topic Selected the Quiz Questions will be displayed.
- 2 – Once the Quiz is completed Result will be displayed along with Percentage.

## 2. SCHEMA

Schema is the skeleton structure that represents the logical view of the entire database. It defines how the data is organized and how the relations among them are associated.



### 3. Operations Description

#### 3.1 Authentication

User authentication is the verification of an active human-to-machine transfer of credentials required for confirmation of a user's authenticity.

```
SELECT * FROM USERS WHERE USER_ID = ? and PASSWORD = ? and  
STUD_FLAG = ?"
```

The User ID, Password and StudentFlag will be checked with the database and if they are valid the Authentication will be successful and Connection will be established.

#### 3.2 Login

Login is the procedure used to get access to an operating system or application. Login requires the user to have his Login ID and Password.

Here we have 2 Kinds of login:

- 1 – Teacher Login
- 2 – Student Login

The functionality of the application depends on the Type of Login.

#### 3.3 Create/Update/Delete the Quiz and Question

Create Quiz:

```
qz.createQuiz(quiz);
```

Method is used to Create a new Quiz depending on the Topic

Create Question:

```
qz.createQuestion(question);
```

Method is used to Create a new Question under a Specific Topic.

While Creating a Question the Difficulty level of the Question and Type of Question has to be Specified.

Update Question:

```
quizJdbcDAO.updateQuestion(question,newQuestionText);
```

Method is used to Update an existing Question Under a Specific Topic.

Delete Question:

```
quizJdbcDAO.deleteQuestion(question);
```

Method is used to Delete an existing Question Under a Specific Topic.

**MCQ Choice Creation:**

```
quizJdbcDAO.insertMCQChoice(mcqChoice);
```

Method is used to create choices for MCQ Type Questions.

While Creating MCQ Choices the count of correct choices and the correct choice has to be Specified.

**Search Based on Question Topic:**

```
quizJdbcDAO.searchQuestion(question, searchFlag);
```

Method is used to Search the Questions available based on the Topic provided by the user.

## 4. Configuration Instructions

### 4.1 Development Tools

We have used Eclipse JEE for developing the Quiz Manager Application

**Eclipse** is an integrated development environment (IDE) used in computer programming, and in 2014 was the most widely used Java IDE in one website's poll. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins.

### 4.2 Database Configuration

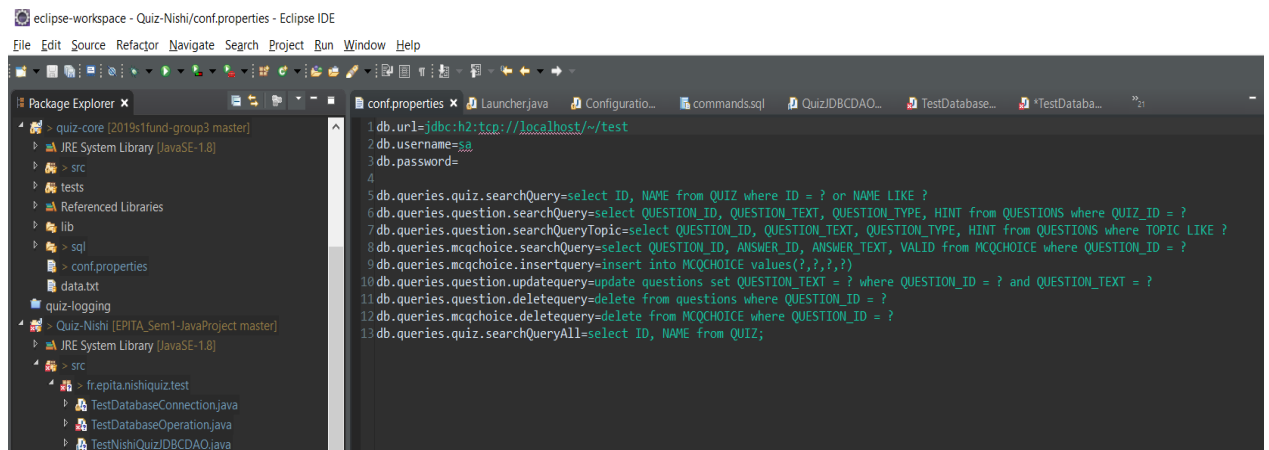
We have used H2 Database Engine as the database for our Quiz Manager Application.

**H2** is a relational database management system written in Java. It can be embedded in Java applications or run in client-server mode.

**Configuration Properties:**

The Below given Configuration Properties have to be used to connect to the Database. The details have to be maintained in the Configuration file. 0

1. URL = jdbc:h2:tcp://localhost/~ /test
2. USERNAME = sa
3. PASSWORD = By default the password is Blank



The Saved Settings for H2 Login has to be **Generic H2(Server)**

English ▼ Preferences Tools Help

### Login

Saved Settings: Generic H2 (Server) ▼

Setting Name: Generic H2 (Server) Save Remove

---

Driver Class: org.h2.Driver

JDBC URL: jdbc:h2:tcp://localhost/~test

User Name: sa

Password:

Connect Test Connection

## 5. SQL Queries

```
CREATE TABLE USERS ( USER_ID VARCHAR(255) PRIMARY KEY,  
PASSWORD VARCHAR(255), STUD_FLAG VARCHAR(255));
```

```
insert into USERS values ('stud','stud','S');
```

```
insert into USERS values ('admin','admin','A');
```

The Users Table holds Admin and Student details with the User ID as the Primary Key, Password and the Stud\_Flag.

```
CREATE TABLE QUIZ (ID bigint auto_increment PRIMARY KEY, NAME  
VARCHAR(255));
```

The Quiz Table holds all the Quiz Names created by the user along with their ID as the Primary Key.

```
CREATE TABLE QUESTIONS (QUIZ_ID INT, QUESTION_ID bigint  
auto_increment, QUESTION_TEXT VARCHAR(255), TOPIC  
VARCHAR(255), DIFFICULTY INT, QUESTION_TYPE VARCHAR(1), HINT  
VARCHAR(50), PRIMARY KEY(QUIZ_ID,QUESTION_ID));
```

The Question Table holds all the questions created by the user with the Quiz ID, Question ID, Question text, Topic of the question, Difficulty of the question, Question Type(MCQ or Open), Hint if the Question is Open Type where Quiz ID and Question ID are Primary Keys.

```
CREATE TABLE MCQChoice(QUESTION_ID INT, ANSWER_ID INT ,  
ANSWER_TEXT VARCHAR(255), VALID BOOLEAN, PRIMARY KEY  
(QUESTION_ID , ANSWER_ID));
```

The MCQChoice Table holds all the MCQ choices for the MCQ Type questions created with the Question ID, Answer ID, Answer Text and Validity of the answer where Question ID and Answer ID are Primary Keys.

quiz.searchQuery

select ID, NAME from QUIZ where ID = 70 or NAME LIKE Java

db.queries.question.searchQuery

select QUESTION\_ID, QUESTION\_TEXT, QUESTION\_TYPE, HINT from  
QUESTIONS where QUIZ\_ID = 70

db.queries.mcqchoice.searchQuery

select QUESTION\_ID, ANSWER\_ID, ANSWER\_TEXT, VALID from  
MCQCHOICE where QUESTION\_ID = 320

db.queries.mcqchoice.insertquery

insert into MCQCHOICE values(public, private, protected, none)

db.queries.question.updatequery

update questions set QUESTION\_TEXT = Programming where QUESTION\_ID  
= 320 and QUESTION\_TEXT = Programming

db.queries.question.deletequery

delete from questions where QUESTION\_ID = 320

db.queries.mcqchoice.deletequery

delete from MCQCHOICE where QUESTION\_ID = 320

db.queries.quiz.searchQueryAll

select ID, NAME from QUIZ;



## 6. References

- Project Requirement, Mind maps and Java Concepts - <https://thomas-broussard.fr/>
- Schema construction - <https://www.draw.io/>
- Reference Website - <https://www.javacodeexamples.com/online-java-test-quiz>
- Programming - <https://stackoverflow.com/>
- Database - <https://www.w3schools.com/sql/>