EPITA

QUIZ MANAGEMENT SYSTEM

ADVANCED JAVA PROJECT

Team Members:

Omkar Yadav

Mounika Kasaraneni

Table of contents:

1. Subject Description

The project Quiz Manager implements the basic features of Quiz management system. This system is implemented based on Questions and selecting options (Multiple choice). Mainly this deals with quiz assessments.

The admin creates the questions, updates and displays the results of the quiz and the student can select the options for each question and submits them at the end. The user will get the result of the quiz at the end.

2.Subject Analysis

2.1 Major Features

Following are the major features of Quiz management system:

- **Login-** this feature allows the students to access the quiz if they are authorized.
- **Create questions and Options**- this feature allows the admin to create the questions and answers(options) for the quiz.
- **Update Questions**- this feature allows the admin to update the questions.
- **Select Option**- this feature allows the student to select the answer/option
- **Save result**- this feature allows the quiz management system to save the result.
- **Take a Quiz**- this feature allows the student to take a quiz.

2.2 Application feasibility

- This system allows a simple way to operate a quiz mechanism by organizing the questions and multiple choice answers to each question. A simple GUI interface is used to conduct this quiz and to store the result at the end of the quiz.
- The development tools used to develop this application are very easily accessible such as Eclipse development kit, Derby Database and Tomcat version 9.0 Server.
- The security is also provided by restricting the unauthorized people to access the quiz by providing the login.

2.3 Data Description

Login details are stored in the database such as E-mail and password of the student. Here Derby Database is used to store these values. Tomcat web server is used to test the applications which are developed using servlets and JSP's.

```
1 INSERT INTO LOGIN (email, password) VALUES ('root@quizify.com','root');
2 INSERT INTO LOGIN (email, password) VALUES ('skillerx@quizify.com','root');
```

2.4 Expected Results

- **Login**: The student is allowed to access the quiz management system once the login credentials gets authenticated. The system will check the values that are entered are true or false. If the credentials are true then the system allows the student to access the quiz.

If the entered credentials are false then the system will not allow the student to take the quiz and it gets redirected to the login again. So, if the student don't know the login credentials cannot appear for the quiz.

- **Quiz Management System:** After the student gets authenticated, a screen will be displayed with a message saying take a quiz.

  Following are the steps for taking quiz:

- Questions will be displayed with multiple choice.
- The student should select one option.
- A Next button will be there to go to next question.
- After the last question a submit button will be there to submit the quiz.
- A result will be displayed at the end of the quiz

  Here mainly two features are implemented quiz editing and quiz execution.

2.5 Algorithm Study

- API Oriented web application
- UI based application

2.6 Scope of Application (limits, evolutions)

Quiz management is the core functionality based on the requirement of the application, student management system is not considered in the scope of the application and it has been assumed as a part of student management system.

- **Limitation:**

The login details are stored in the database using the queries. The system is based on API oriented web application which is very easy for a new student to take the quiz.

Lack of organizing sample evaluation of the quiz. Only MCQ's are composed not any other like open questions

Former questions are reused for every test.

- **Evaluation:**

The system is a service oriented architecture (API based). The quiz management system is effectively developed within the scope of the application proposed. The GUI is developed by using different fuctionalities like quiz editing and quiz execution by storing the result of all the questions. This is possible by using servelts, springs and database.

3.Conception

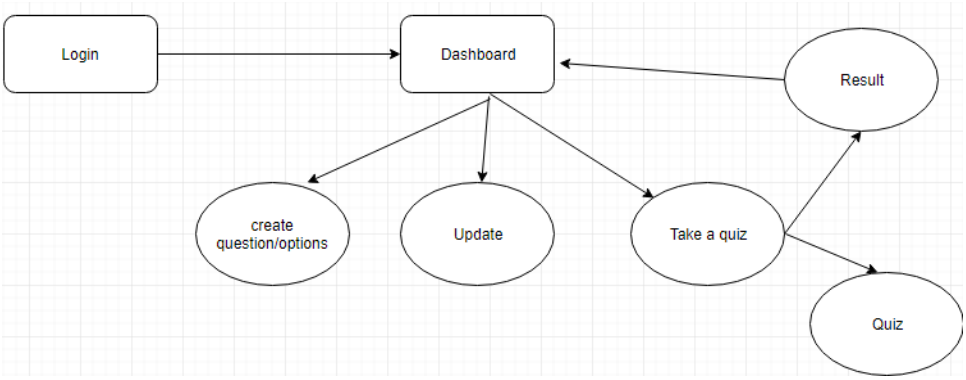
3.1 Chosen Algorithm

- An API oriented GUI based application is used for development of the project.

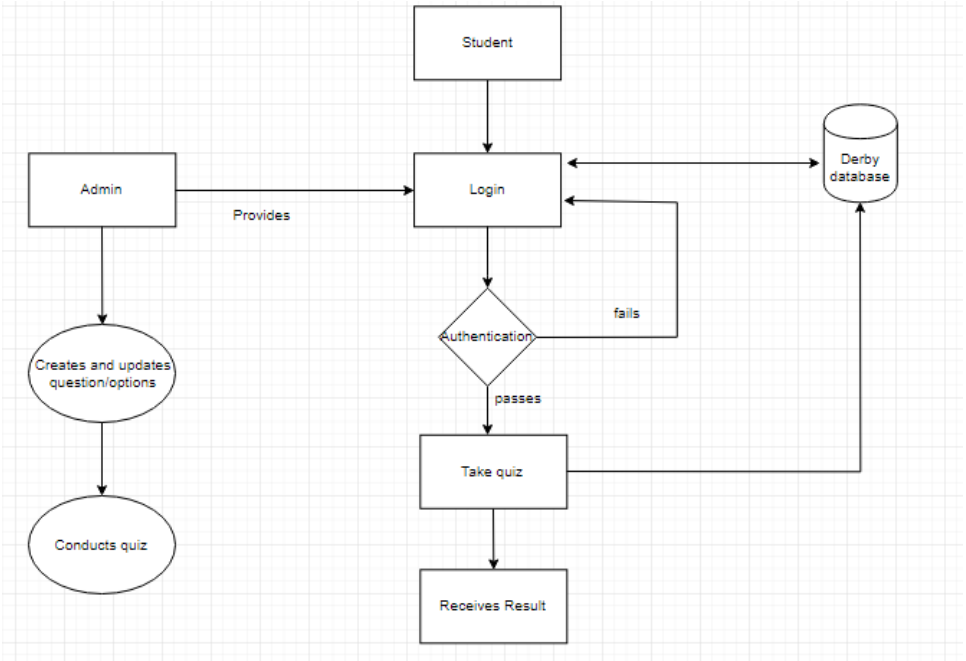
3.2 Data Structures

Data structure used in this project is LIST, which is a collection of elements. List is used in the project to store the questions for the quiz. String is also a widely used data structure in this project.

3.3 Global application Flow



3.4 Global Schema and major features Schema



4.GUI Operations Description

Following are the GUI operations implemented:

4.1 Login

- This operation is responsible to save the Login details of the student by taking details of email and password. If it is authenticated it will redirect to take the quiz and if the details are incorrect then it will again redirect to login.
- dopost() method is used to receive post requests by the servlet for the parameters login and password.
- The boolean value will be true if the parameters are true and is authenticated. It will be false if the parameters are incorrect and will not be saved in the Login database.

4.2 Create question/options

- This operation allows the admin to update, delete, search and create the questions for the quiz. The admin can create the Mcq options for the questions.
- Here for delete data and mcq's we are using a method question.gettype. If it is not null it will delete the data and mcq's.

4.3 Select Option

- This operation allows the student to select the option from given mcq.

4.4 Update Question

- This operation updates the questions in the quiz. List data structure is used to update the questions and collect the questions.
- If question type does not return null then the questionDAO will update the question when the student is taking the question.

4.5 Take a Quiz

- This operation allows the students to take the quiz once they are authenticated in the Login. The login page will be redirected to the quiz page.
- Session is created for each student when they do login. This session will open until they finish the quiz and by selecting the submit at the end to get the result of the quiz.

4.6 Save the Result

- This operation allows to save all the options selected by the student in the quiz.
- Here Transaction.commit() method is used to save the changes after selecting the options for the questions. These are saved in the database to display the result at the end of the quiz.

5. Configuration instructions

5.1 Development tools

- Eclipse Oxygen eclipse Java EE IDE for Web Developers.

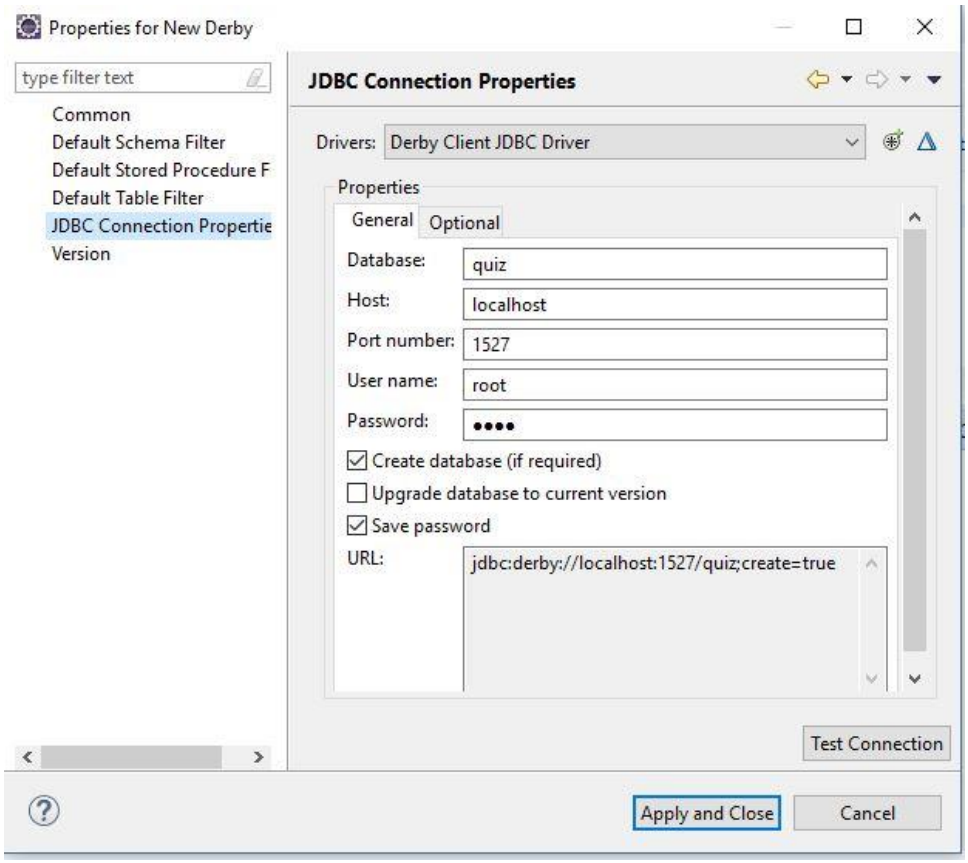
- Derby Database.
- Java JDK 8
- Tomcat version 9.0Server

5.2 Schema and Database Configuration

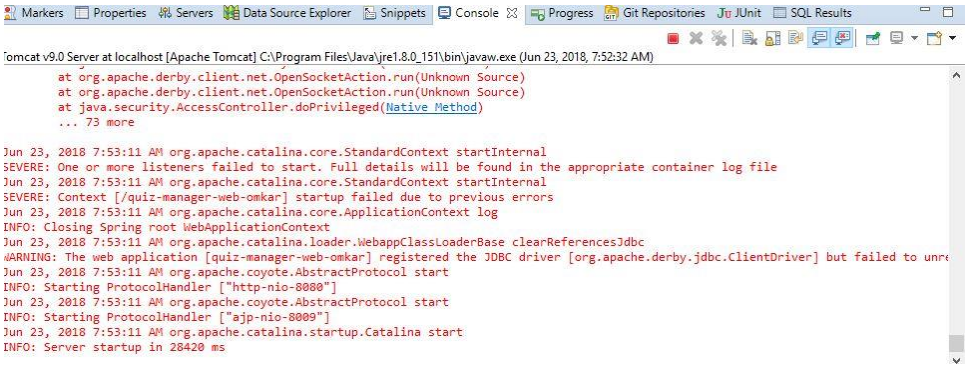
For the development, we have used a database table “LOGIN” for saving the email and password values.

An “data.sql” file is maintained under “SQL” folder with all the required queries to create the table.

- Below screenshot shows the details of database configuration properties



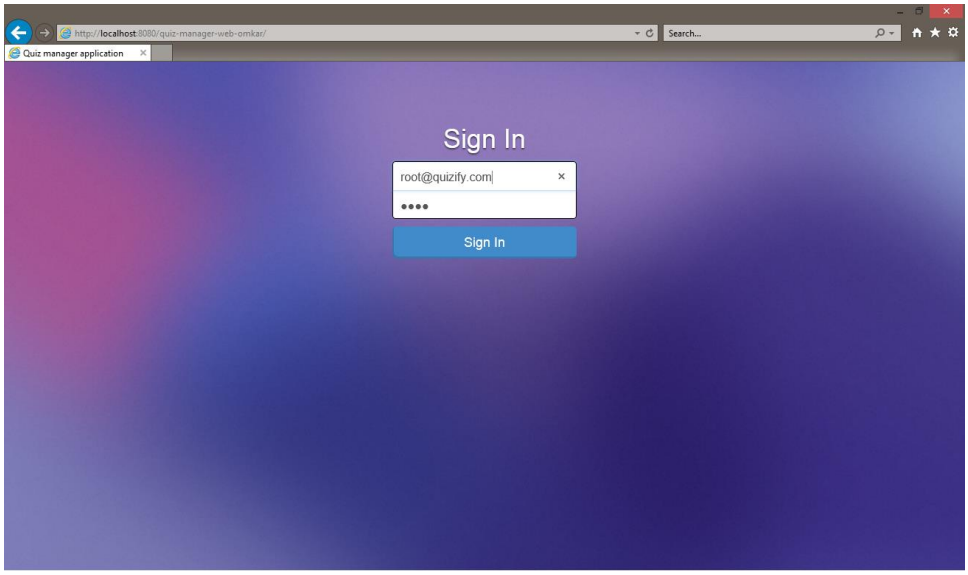
- Below screenshot shows the Tomcat version 9.0 server connection



6. Commented Screenshots

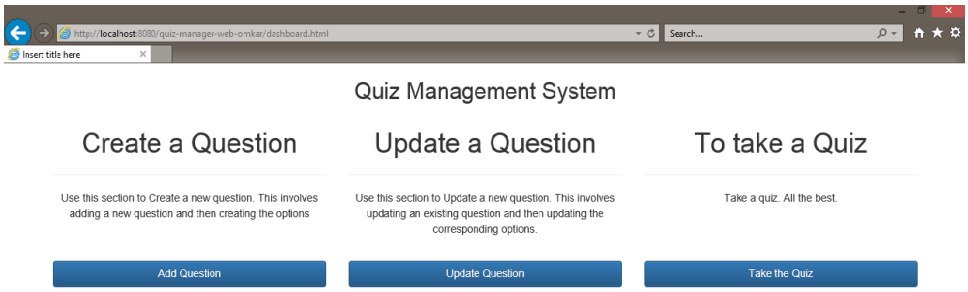
- **Sign In:**

The student should enter the email id and password to take the quiz.



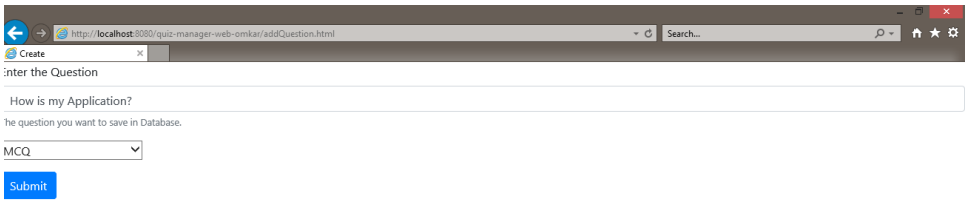
- **Quiz Management System:**

After login the main part of quiz execution and editing takes place. In the below screenshot we can see three options as create a question, Update a question and take a quiz.



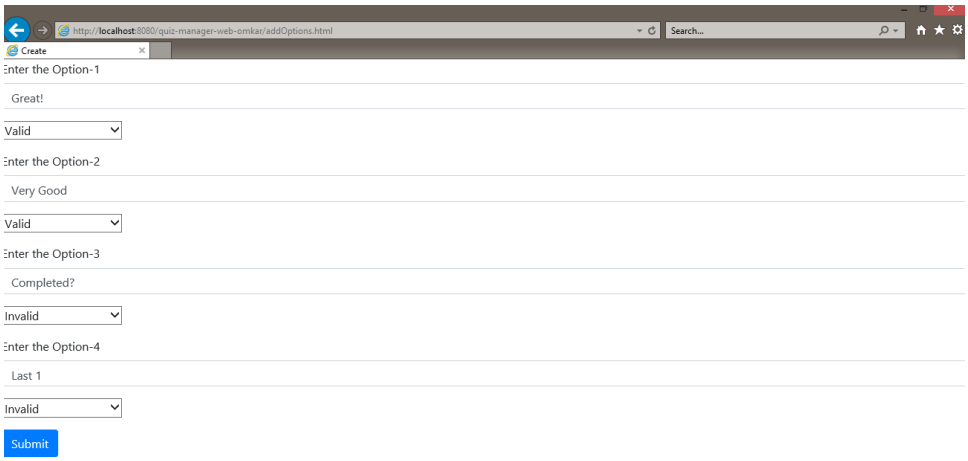
- **Create a question:**

The quiz management system can create a question and save te question by using te submit button.



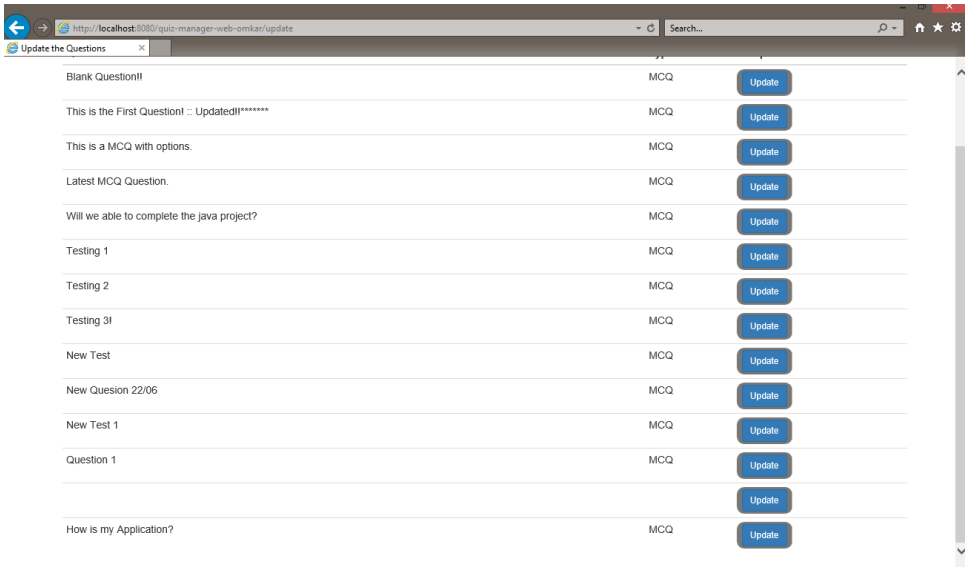
- **Create option:**

MCQ's(Options) are created for the questions.

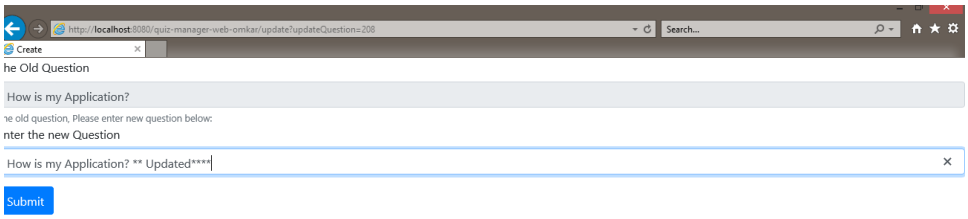


- **Update questions:**

Here questions can be updated.

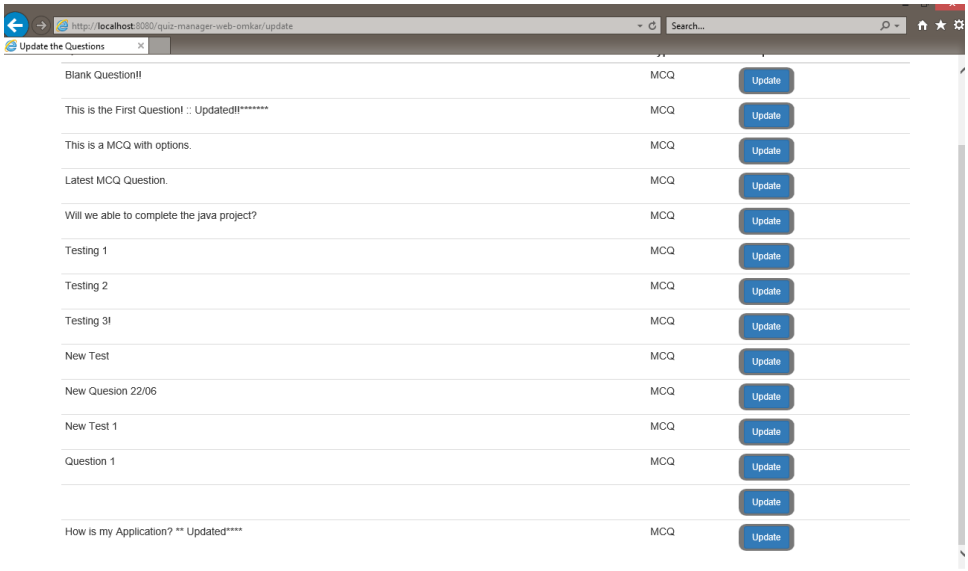


- **Updating old question with new question:**



-

Looks like the below screenshot after updating the question.



7.Bibliography

- Project Scope, Design patterns, OO concepts: http://thomas-broussard.fr/
- Some references from https://stackoverflow.com/