COURSE: ADVANCED JAVA PROGRAMMING

PROJECT: JAVA QUIZ MANAGER

TECHNICAL SPECIFICATIONS

Venkat Kumari Natarajan

Table of contents

INTRODUCTION:

The technical specification document explains the features of the project. It also contains the objectives, limitations, requirements, project risks, application flow for the project. The main objective of the project is to create online quiz manager that lists the questions and evaluates the answers given by the student.

The frontend of the project has been done using HTML, JSP and Angular Js. The Backend has been done using SPRING framework, especially JPA using REST and Java 8. Tomcat v8.5 is the server used in the project.

This quiz manager can be used in schools, colleges and competitive exams to evaluate large number of students and the score of the student can be stored in a database. This is helpful in creating the statistics of the score among the students.

OBJECTIVES:

The objective is to create online quiz manager. The quiz manager can be used by two identities.

-Admin: To add, delete and update questions and answers.

-Student: To mark the answers and get the final score.

The admin can access the score of the student in the database.

LIMITATIONS:

- Manage the online quiz
- Allow the students to view only his/her score after the completion of the quiz
- The type of question is only multiple choice.

SOFTWARE REQUIREMENTS:

Below are the technologies used for the creation of quiz manager.

Backend:

- Java version 9
- Spring
- JSP
- TOMCAT

Frontend:

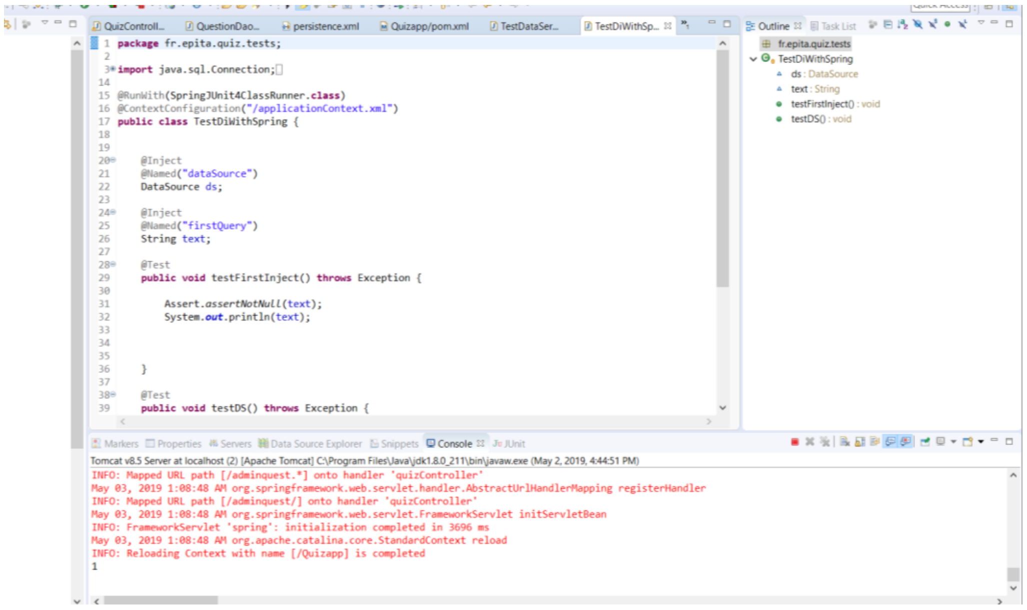
- HTML 5
- CSS
- JSP
- Angular

Database:

- Object DB

Testing:

JUnit is a Regression Testing Framework used by developers to implement unit testing in Java and accelerate programming speed and increase the quality of code. JUnit Framework can be easily integrated with either of the following −Eclipse, Ant, Maven.



```java
package fr.epita.quiz.tests;

import java.sql.Connection;

@RunWith(SpringJUnit4ClassRunner.class)
@ContextConfiguration("/applicationContext.xml")
public class TestDiWithSpring {


    @Inject
    @Named("dataSource")
    DataSource ds;

    @Inject
    @Named("firstQuery")
    String text;

    @Test
    public void testFirstInject() throws Exception {

        Assert.assertNotNull(text);
        System.out.println(text);



    }

    @Test
    public void testDS() throws Exception {
```

```
Tomcat v8.5 Server at localhost (2) [Apache Tomcat] C:\Program Files\Java\jdk1.8.0_211\bin\javaw.exe (May 2, 2019, 4:44:51 PM)
INFO: Mapped URL path [/adminquest.*] onto handler 'quizController'
May 03, 2019 1:08:48 AM org.springframework.web.servlet.handler.AbstractUrlHandlerMapping registerHandler
INFO: Mapped URL path [/adminquest/] onto handler 'quizController'
May 03, 2019 1:08:48 AM org.springframework.web.servlet.FrameworkServlet initServletBean
INFO: FrameworkServlet 'spring': initialization completed in 3696 ms
May 03, 2019 1:08:48 AM org.apache.catalina.core.StandardContext reload
INFO: Reloading Context with name [/Quizapp] is completed
1
```

APPLICATION MODULE:

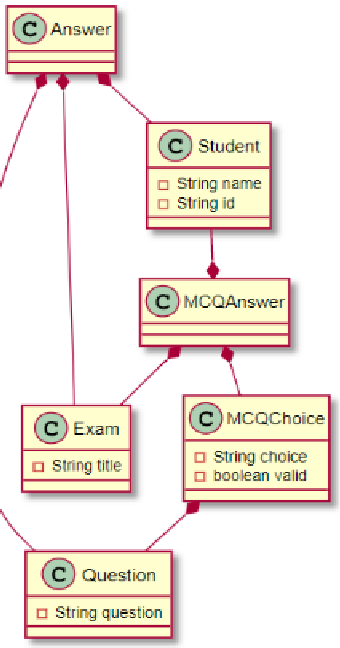
There are two sub modules in this phase.

- Admin module
- User module

User module: The student can use the application to answer the quiz. Upon completion, the candidate will receive his result.

Admin module: The admin/professor can use the application to add, deleted and update questions. They can choose the type of questions and view the score of the students.

APPLICATION FLOW:



CONCEPTION:

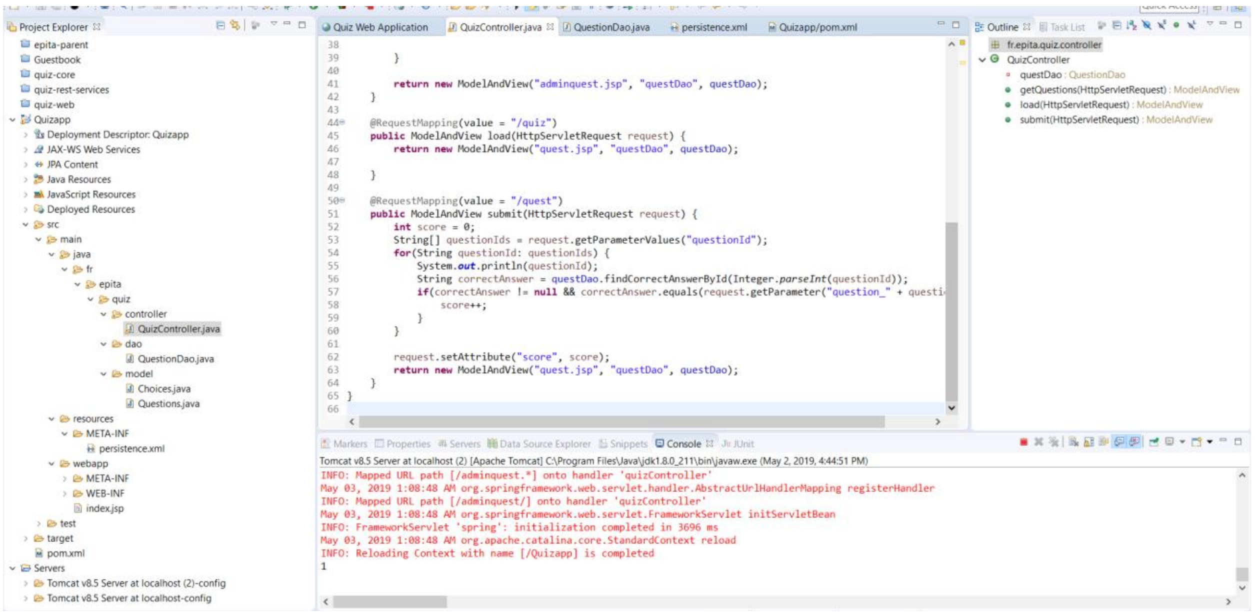
There are three MVC modules involved in developing the backend of the quiz manager.

They are:

1) Data model
2) Controller
3) DAO

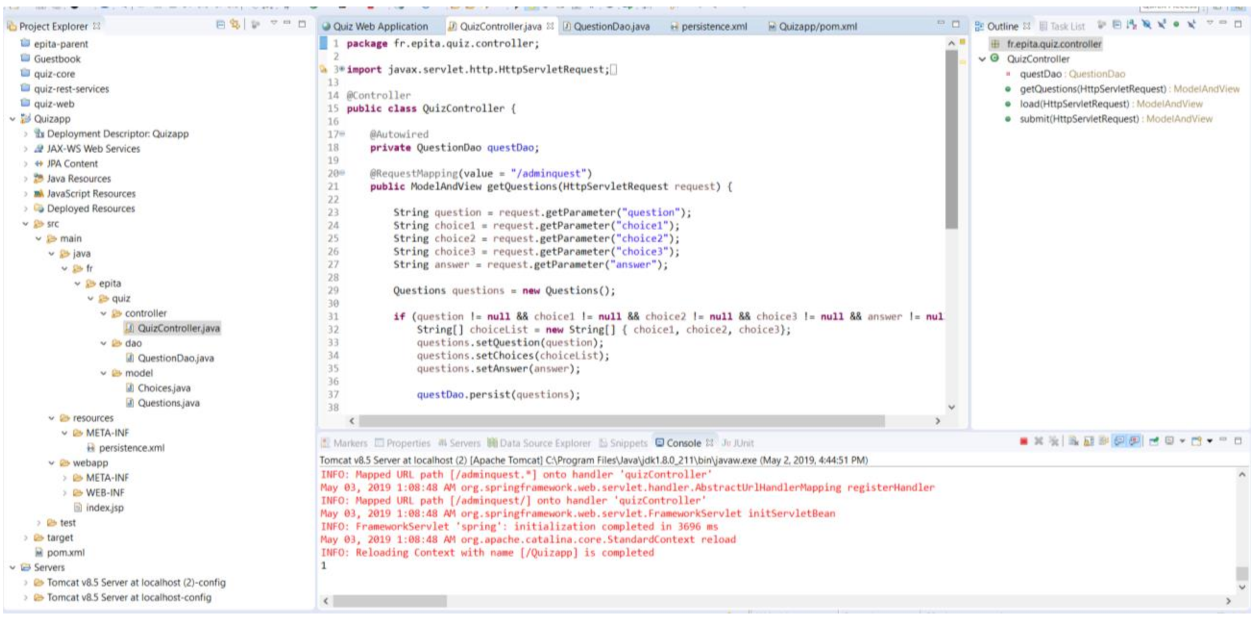
QuizContoller.java:

The backend functionalities are embedded into this class.



```java
        }

        return new ModelAndView("adminquest.jsp", "questDao", questDao);
    }

    @RequestMapping(value = "/quiz")
    public ModelAndView load(HttpServletRequest request) {
        return new ModelAndView("quest.jsp", "questDao", questDao);

    }

    @RequestMapping(value = "/quest")
    public ModelAndView submit(HttpServletRequest request) {
        int score = 0;
        String[] questionIds = request.getParameterValues("questionId");
        for(String questionId: questionIds) {
            System.out.println(questionId);
            String correctAnswer = questDao.findCorrectAnswerById(Integer.parseInt(questionId));
            if(correctAnswer != null && correctAnswer.equals(request.getParameter("question_" + questi
                score++;
            }
        }

        request.setAttribute("score", score);
        return new ModelAndView("quest.jsp", "questDao", questDao);
    }
}
```

```
Tomcat v8.5 Server at localhost (2) [Apache Tomcat] C:\Program Files\Java\jdk1.8.0_211\bin\javaw.exe (May 2, 2019, 4:44:51 PM)
INFO: Mapped URL path [/adminquest.*] onto handler 'quizController'
May 03, 2019 1:08:48 AM org.springframework.web.servlet.handler.AbstractUrlHandlerMapping registerHandler
INFO: Mapped URL path [/adminquest/] onto handler 'quizController'
May 03, 2019 1:08:48 AM org.springframework.web.servlet.FrameworkServlet initServletBean
INFO: FrameworkServlet 'spring': initialization completed in 3696 ms
May 03, 2019 1:08:48 AM org.apache.catalina.core.StandardContext reload
INFO: Reloading Context with name [/Quizapp] is completed
1
```

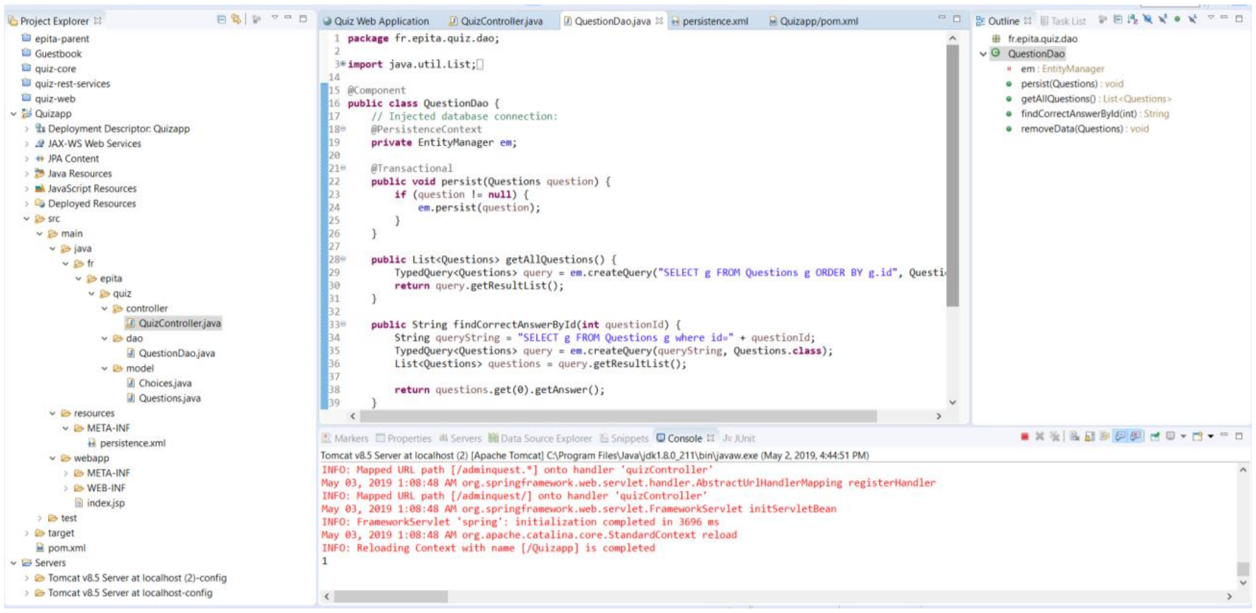


```java
package fr.epita.quiz.controller;

import javax.servlet.http.HttpServletRequest;

@Controller
public class QuizController {

    @Autowired
    private QuestionDao questDao;

    @RequestMapping(value = "/adminquest")
    public ModelAndView getQuestions(HttpServletRequest request) {

        String question = request.getParameter("question");
        String choice1 = request.getParameter("choice1");
        String choice2 = request.getParameter("choice2");
        String choice3 = request.getParameter("choice3");
        String answer = request.getParameter("answer");

        Questions questions = new Questions();

        if (question != null && choice1 != null && choice2 != null && choice3 != null && answer != nul
            String[] choiceList = new String[] { choice1, choice2, choice3};
            questions.setQuestion(question);
            questions.setChoices(choiceList);
            questions.setAnswer(answer);

            questDao.persist(questions);

```

```
Tomcat v8.5 Server at localhost (2) [Apache Tomcat] C:\Program Files\Java\jdk1.8.0_211\bin\javaw.exe (May 2, 2019, 4:44:51 PM)
INFO: Mapped URL path [/adminquest.*] onto handler 'quizController'
May 03, 2019 1:08:48 AM org.springframework.web.servlet.handler.AbstractUrlHandlerMapping registerHandler
INFO: Mapped URL path [/adminquest/] onto handler 'quizController'
May 03, 2019 1:08:48 AM org.springframework.web.servlet.FrameworkServlet initServletBean
INFO: FrameworkServlet 'spring': initialization completed in 3696 ms
May 03, 2019 1:08:48 AM org.apache.catalina.core.StandardContext reload
INFO: Reloading Context with name [/Quizapp] is completed
1
```

QuestionDAO.java:

The database functionalities are embedded into this class.



```java
package fr.epita.quiz.dao;

import java.util.List;

@Component
public class QuestionDao {
    // Injected database connection:
    @PersistenceContext
    private EntityManager em;

    @Transactional
    public void persist(Questions question) {
        if (question != null) {
            em.persist(question);
        }
    }

    public List<Questions> getAllQuestions() {
        TypedQuery<Questions> query = em.createQuery("SELECT g FROM Questions g ORDER BY g.id", Questi
        return query.getResultList();
    }

    public String findCorrectAnswerById(int questionId) {
        String queryString = "SELECT g FROM Questions g where id=" + questionId;
        TypedQuery<Questions> query = em.createQuery(queryString, Questions.class);
        List<Questions> questions = query.getResultList();

        return questions.get(0).getAnswer();
    }
```

```
Tomcat v8.5 Server at localhost (2) [Apache Tomcat] C:\Program Files\Java\jdk1.8.0_211\bin\javaw.exe (May 2, 2019, 4:44:51 PM)
INFO: Mapped URL path [/adminquest.*] onto handler 'quizController'
May 03, 2019 1:08:48 AM org.springframework.web.servlet.handler.AbstractUrlHandlerMapping registerHandler
INFO: Mapped URL path [/adminquest/] onto handler 'quizController'
May 03, 2019 1:08:48 AM org.springframework.web.servlet.FrameworkServlet initServletBean
INFO: FrameworkServlet 'spring': initialization completed in 3696 ms
May 03, 2019 1:08:48 AM org.apache.catalina.core.StandardContext reload
INFO: Reloading Context with name [/Quizapp] is completed
1
```

Bibliography:

https://github.com/thomasbroussard

https://thomas-broussard.fr/work/java/courses/project/advanced.xhtml