User Guide:

A JAVA Project Report

(Quiz Manager Web Application)

Submitted by:

Shiyamaladevi PACHAIAPPAN

(patchaiappanshiyamaladevi@gmail.com)

Introduction

This report will give overall information and technical specifications, that how the quiz manager is established, efforts and implementations involved.

Quiz manager application in Java using Eclipse and H2 database can be quite a good experience which requires understanding of libraries, and all the aspects of Java. By implementing all the procedures and methods we have used almost 8 classes to establish great output for the end user, and we have used brainstorming to solve the complete code and efforts are made to get an output for a code by gathering a list of procedures and UML diagrams.

Involved many features and concepts in this context to develop the code and made code run on any computer architecture by making Java Architectural neutral.

Overview

This report includes all the basic operations and steps which are involved in the java quiz manager web application project development.

Motivation:

This project has been greatly influenced us to learn the more concepts of **JAVA**.

What we learned is:

Java is a general-purpose, concurrent, class-based, object-oriented computer programming language that is specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another.

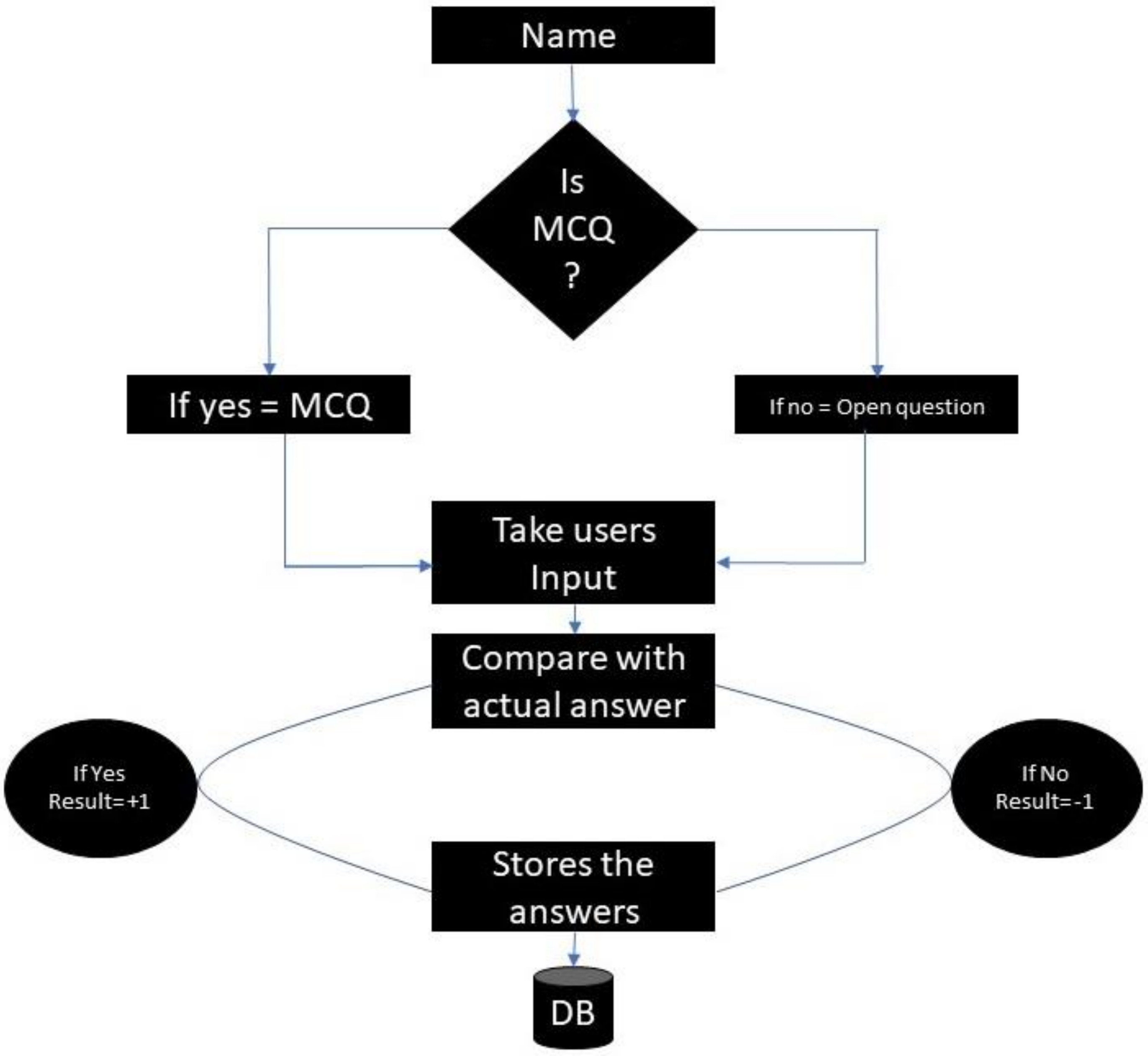
Objective

The main goal of this project is to develop quiz manager web application using Java which can automatically connect to a database and store information in the database.

The main process of the Quiz manager is:

1. Input Name
2. Display the ono-by-one Questions
3. User submits the quiz
4. User is able to see the score in web page
5. Answers will be stored into the data base.

Flow Diagram



DAO, Operations and Methods

Authenticate: User authentication is done and takes the Name as the Username then it connects the database and execute the Questions.

Display : This process is used to display all the questions one-by-one from the database.

Check : Console will check whether the question is MCQ or Open Question.

Storage : Console will take the users input and answers will be stored in the database.

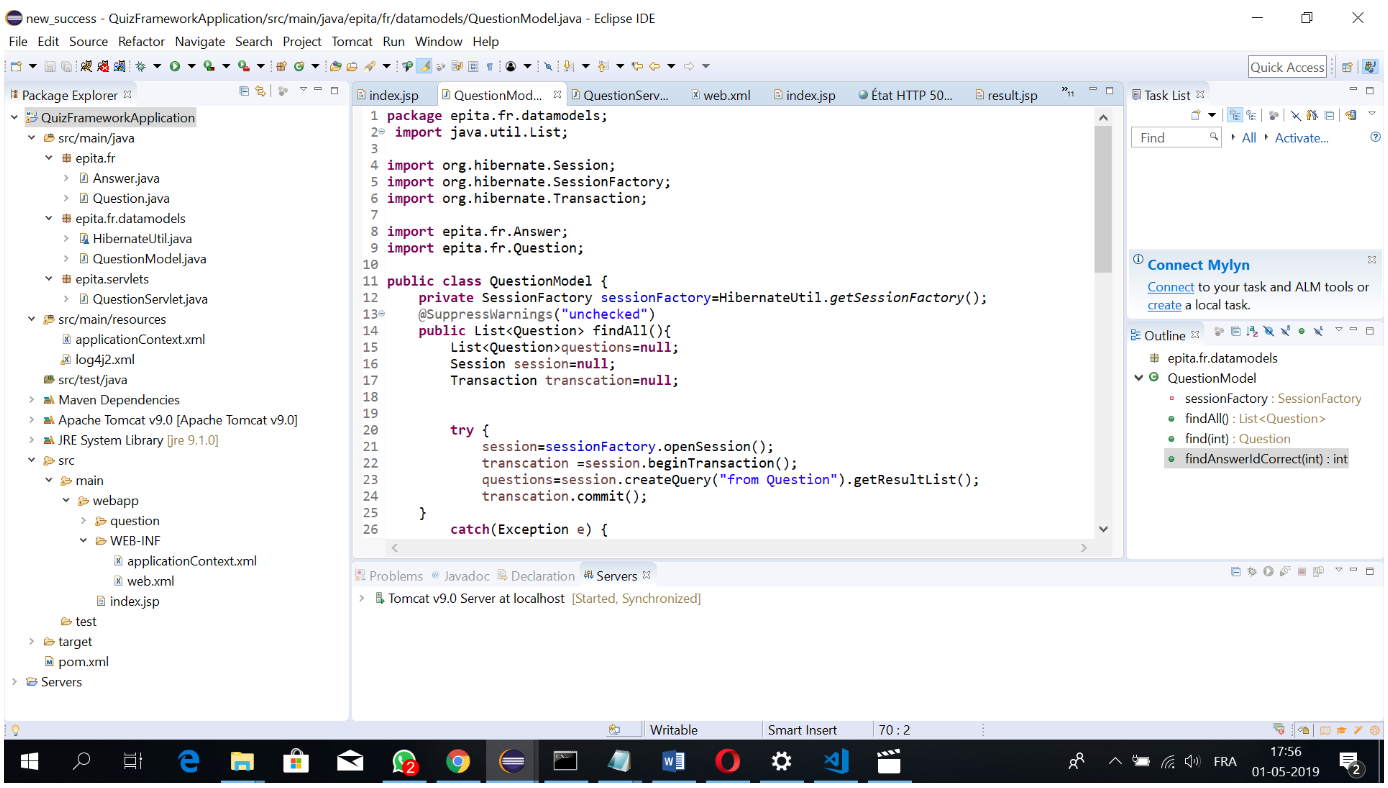
TECHNICAL SPECIFICATIONS:

TESTING:

To satisfy the requirements and check whether every build services in the application are executed as expected, one of the earliest, testing efforts performed on the code is Unit Testing. To quickly test any new code or changes to existing code without the overhead and additional time involved in tasks such as server configuration, services setup and application deployment, we integrated the popular testing framework JUnit Framework to execute the unit test on every small code module. Our objectives are in writing and executing the tests

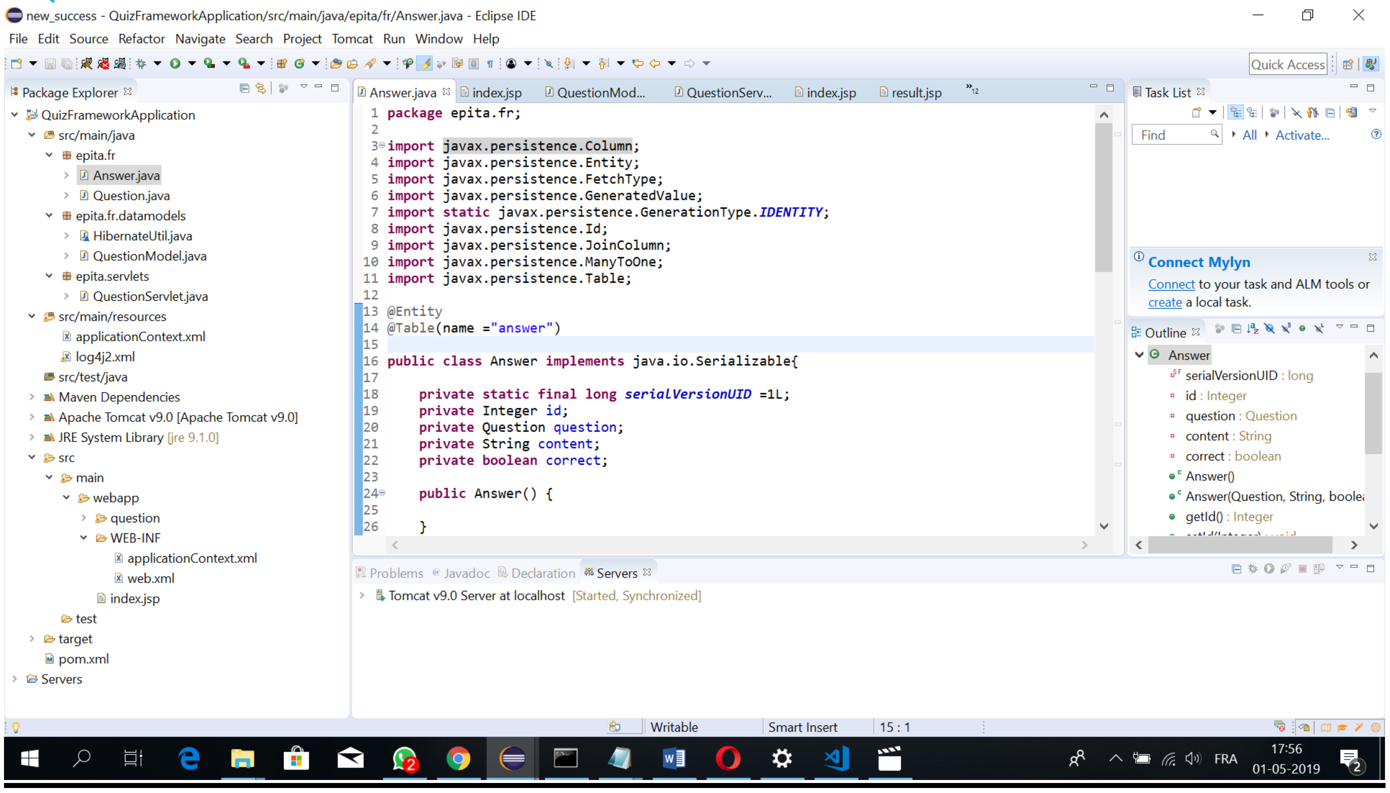
- We want to code and run the tests without leaving the IDE (Eclipse).
- There should be no special deployment of the code required - We should be able to exploit other code analysis tools such as Metrics and find bugs right from within the IDE so we can find any bugs right away and fix those issues.

SCREEN SHORT:



```java
package epita.fr.datamodels;
 import java.util.List;

import org.hibernate.Session;
import org.hibernate.SessionFactory;
import org.hibernate.Transaction;

import epita.fr.Answer;
import epita.fr.Question;

public class QuestionModel {
    private SessionFactory sessionFactory=HibernateUtil.getSessionFactory();
    @SuppressWarnings("unchecked")
    public List<Question> findAll(){
        List<Question>questions=null;
        Session session=null;
        Transaction transcation=null;


        try {
            session=sessionFactory.openSession();
            transcation =session.beginTransaction();
            questions=session.createQuery("from Question").getResultList();
            transcation.commit();
    }
        catch(Exception e) {
```

```
Tomcat v9.0 Server at localhost  [Started, Synchronized]
```



new_success - QuizFrameworkApplication/src/main/java/epita/fr/Answer.java - Eclipse IDE

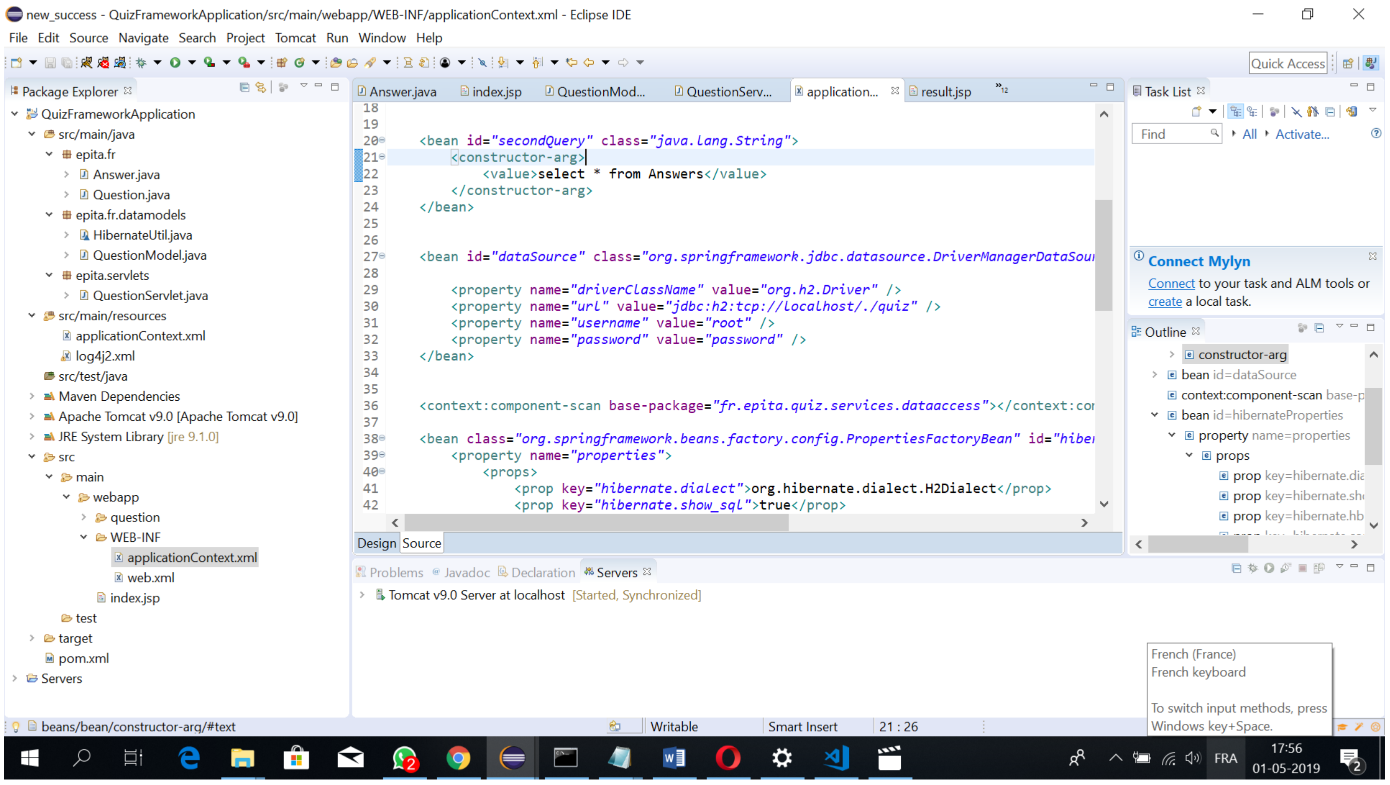
```
QuizFrameworkApplication
  src/main/java
    epita.fr
      Answer.java
      Question.java
    epita.fr.datamodels
      HibernateUtil.java
      QuestionModel.java
    epita.servlets
      QuestionServlet.java
  src/main/resources
    applicationContext.xml
    log4j2.xml
  src/test/java
  Maven Dependencies
  Apache Tomcat v9.0 [Apache Tomcat v9.0]
  JRE System Library [jre 9.1.0]
  src
    main
      webapp
        question
        WEB-INF
          applicationContext.xml
          web.xml
        index.jsp
    test
  target
  pom.xml
Servers
```

```java
package epita.fr;

import javax.persistence.Column;
import javax.persistence.Entity;
import javax.persistence.FetchType;
import javax.persistence.GeneratedValue;
import static javax.persistence.GenerationType.IDENTITY;
import javax.persistence.Id;
import javax.persistence.JoinColumn;
import javax.persistence.ManyToOne;
import javax.persistence.Table;

@Entity
@Table(name ="answer")

public class Answer implements java.io.Serializable{

    private static final long serialVersionUID =1L;
    private Integer id;
    private Question question;
    private String content;
    private boolean correct;

    public Answer() {

    }
```

```
Answer
  serialVersionUID : long
  id : Integer
  question : Question
  content : String
  correct : boolean
  Answer()
  Answer(Question, String, boolea
  getId() : Integer
```

Tomcat v9.0 Server at localhost [Started, Synchronized]

Writable   Smart Insert   15 : 1



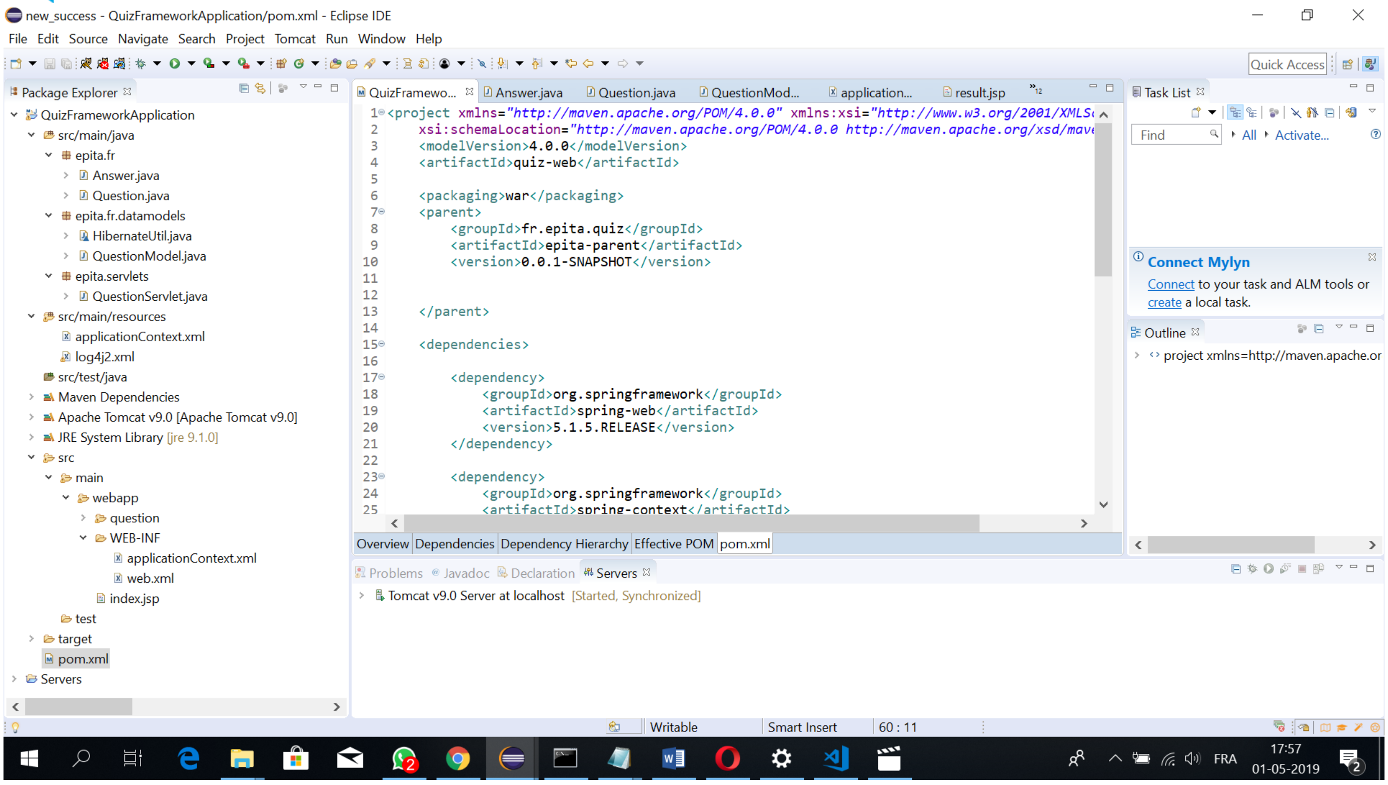
new_success - QuizFrameworkApplication/src/main/webapp/WEB-INF/applicationContext.xml - Eclipse IDE

```xml
    <bean id="secondQuery" class="java.lang.String">
        <constructor-arg>
            <value>select * from Answers</value>
        </constructor-arg>
    </bean>


    <bean id="dataSource" class="org.springframework.jdbc.datasource.DriverManagerDataSou

        <property name="driverClassName" value="org.h2.Driver" />
        <property name="url" value="jdbc:h2:tcp://localhost/./quiz" />
        <property name="username" value="root" />
        <property name="password" value="password" />
    </bean>


    <context:component-scan base-package="fr.epita.quiz.services.dataaccess"></context:cor

    <bean class="org.springframework.beans.factory.config.PropertiesFactoryBean" id="hibe
        <property name="properties">
            <props>
                <prop key="hibernate.dialect">org.hibernate.dialect.H2Dialect</prop>
                <prop key="hibernate.show_sql">true</prop>
```

```
constructor-arg
bean id=dataSource
context:component-scan base-p
bean id=hibernateProperties
  property name=properties
    props
      prop key=hibernate.dia
      prop key=hibernate.sh
      prop key=hibernate.hb
```

Tomcat v9.0 Server at localhost [Started, Synchronized]

beans/bean/constructor-arg/#text   Writable   Smart Insert   21 : 26


new_success - QuizFrameworkApplication/pom.xml - Eclipse IDE

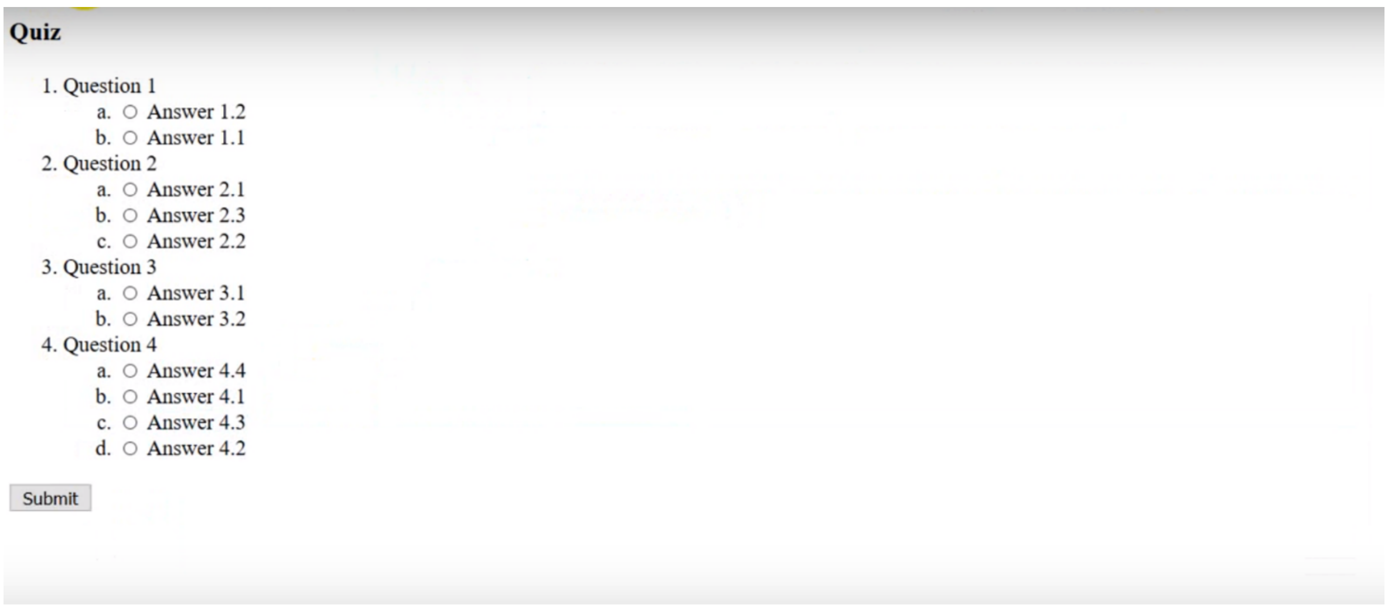
File Edit Source Navigate Search Project Tomcat Run Window Help
Quick Access
Package Explorer
QuizFramewo... Answer.java Question.java QuestionMod... application... result.jsp
Task List
Find All Activate...
Connect Mylyn
Connect to your task and ALM tools or create a local task.
Outline
project xmlns=http://maven.apache.or
Overview Dependencies Dependency Hierarchy Effective POM pom.xml
Problems Javadoc Declaration Servers
Writable Smart Insert 60 : 11
FRA 17:57 01-05-2019

```
QuizFrameworkApplication
  src/main/java
    epita.fr
      Answer.java
      Question.java
    epita.fr.datamodels
      HibernateUtil.java
      QuestionModel.java
    epita.servlets
      QuestionServlet.java
  src/main/resources
    applicationContext.xml
    log4j2.xml
  src/test/java
  Maven Dependencies
  Apache Tomcat v9.0 [Apache Tomcat v9.0]
  JRE System Library [jre 9.1.0]
  src
    main
      webapp
        question
        WEB-INF
          applicationContext.xml
          web.xml
        index.jsp
    test
  target
  pom.xml
Servers
```

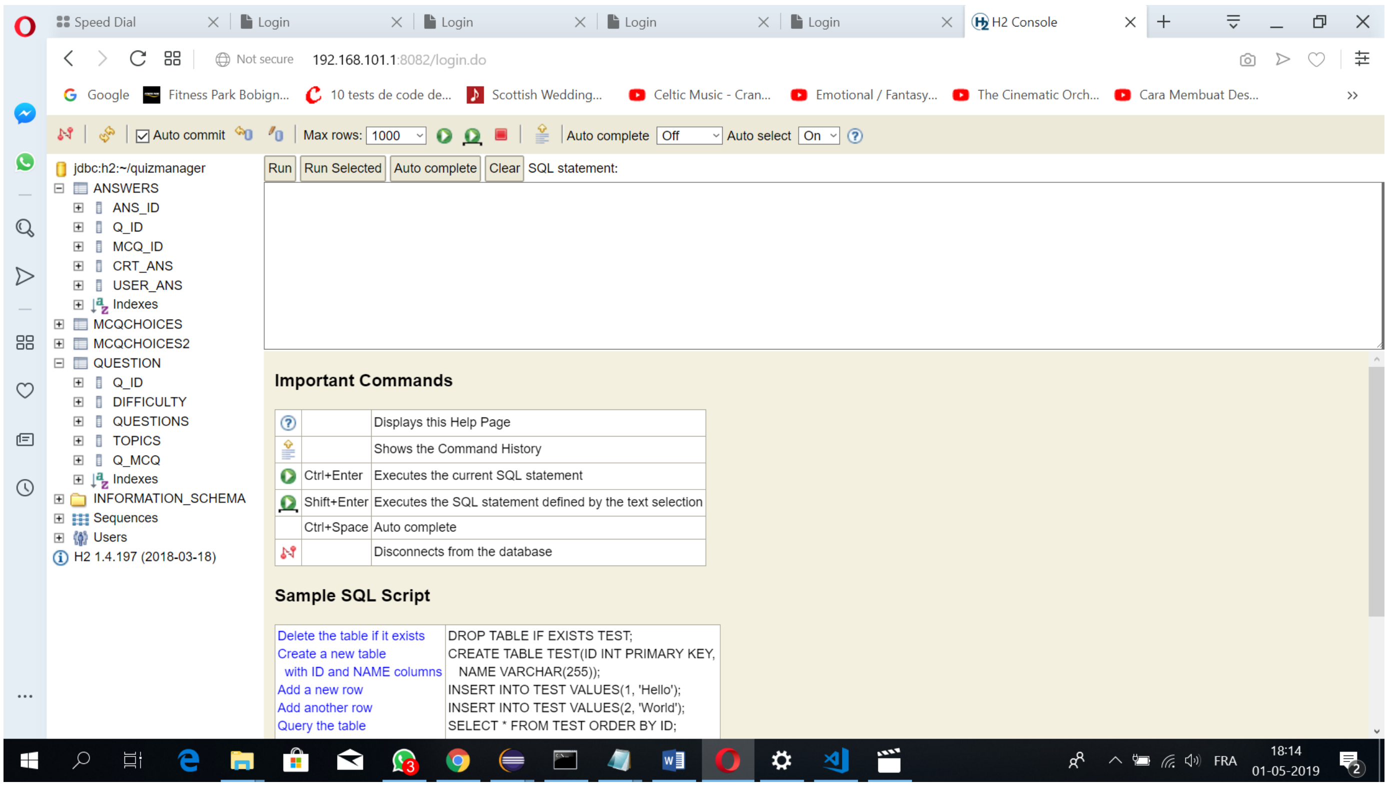
```
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSc
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/mave
    <modelVersion>4.0.0</modelVersion>
    <artifactId>quiz-web</artifactId>

    <packaging>war</packaging>
    <parent>
        <groupId>fr.epita.quiz</groupId>
        <artifactId>epita-parent</artifactId>
        <version>0.0.1-SNAPSHOT</version>


    </parent>

    <dependencies>

        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-web</artifactId>
            <version>5.1.5.RELEASE</version>
        </dependency>

        <dependency>
            <groupId>org.springframework</groupId>
            <artifactId>spring-context</artifactId>
```

```
Tomcat v9.0 Server at localhost  [Started, Synchronized]
```

OUTPUT:


Quiz

1. Question 1
   a. ○ Answer 1.2
   b. ○ Answer 1.1
2. Question 2
   a. ○ Answer 2.1
   b. ○ Answer 2.3
   c. ○ Answer 2.2
3. Question 3
   a. ○ Answer 3.1
   b. ○ Answer 3.2
4. Question 4
   a. ○ Answer 4.4
   b. ○ Answer 4.1
   c. ○ Answer 4.3
   d. ○ Answer 4.2



Auto commit Max rows: 1000 Auto complete Off Auto select On

SQL statement:

```
jdbc:h2:~/quizmanager
  ANSWERS
    ANS_ID
    Q_ID
    MCQ_ID
    CRT_ANS
    USER_ANS
    Indexes
  MCQCHOICES
  MCQCHOICES2
  QUESTION
    Q_ID
    DIFFICULTY
    QUESTIONS
    TOPICS
    Q_MCQ
    Indexes
  INFORMATION_SCHEMA
  Sequences
  Users
  H2 1.4.197 (2018-03-18)
```

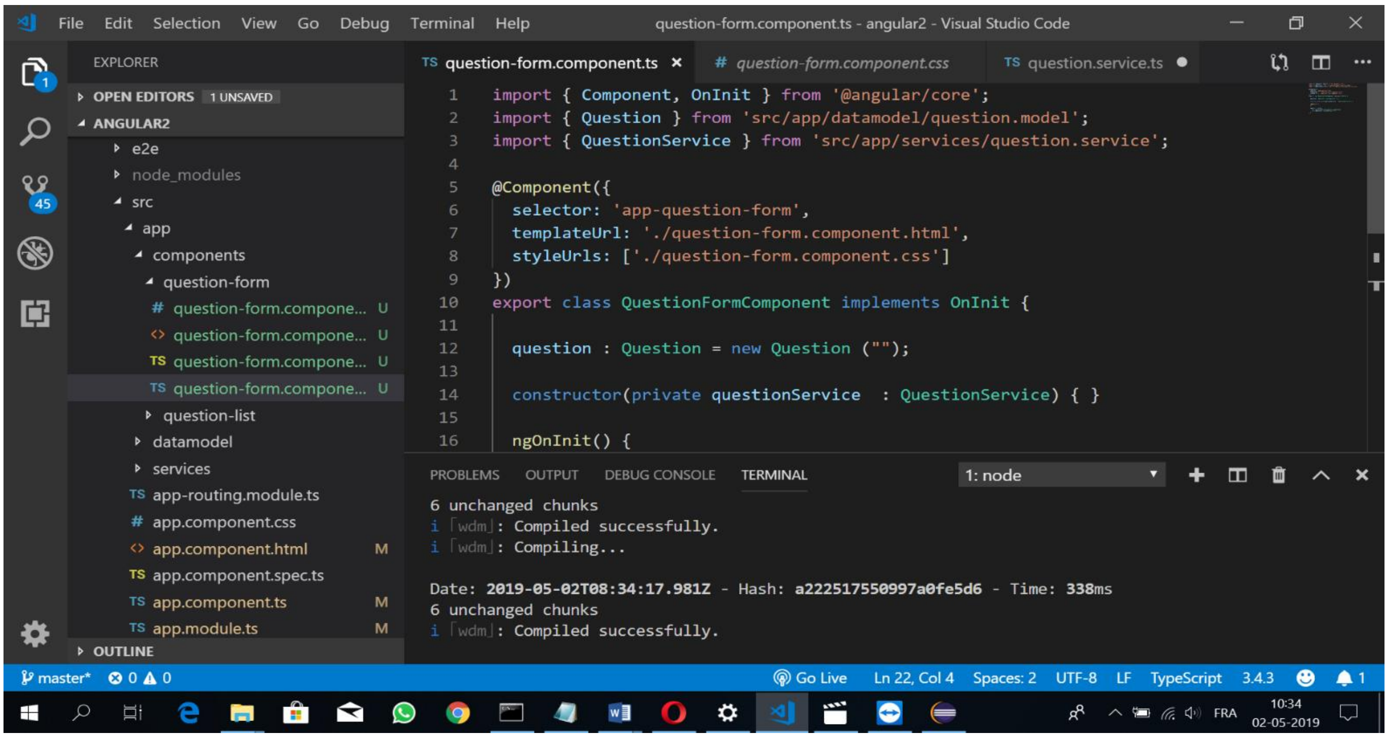
Important Commands

?		Displays this Help Page
		Shows the Command History
	Ctrl+Enter	Executes the current SQL statement
	Shift+Enter	Executes the SQL statement defined by the text selection
	Ctrl+Space	Auto complete
		Disconnects from the database

Sample SQL Script

Delete the table if it exists	DROP TABLE IF EXISTS TEST;
Create a new table with ID and NAME columns	CREATE TABLE TEST(ID INT PRIMARY KEY, NAME VARCHAR(255));
Add a new row	INSERT INTO TEST VALUES(1, 'Hello');
Add another row	INSERT INTO TEST VALUES(2, 'World');
Query the table	SELECT * FROM TEST ORDER BY ID;

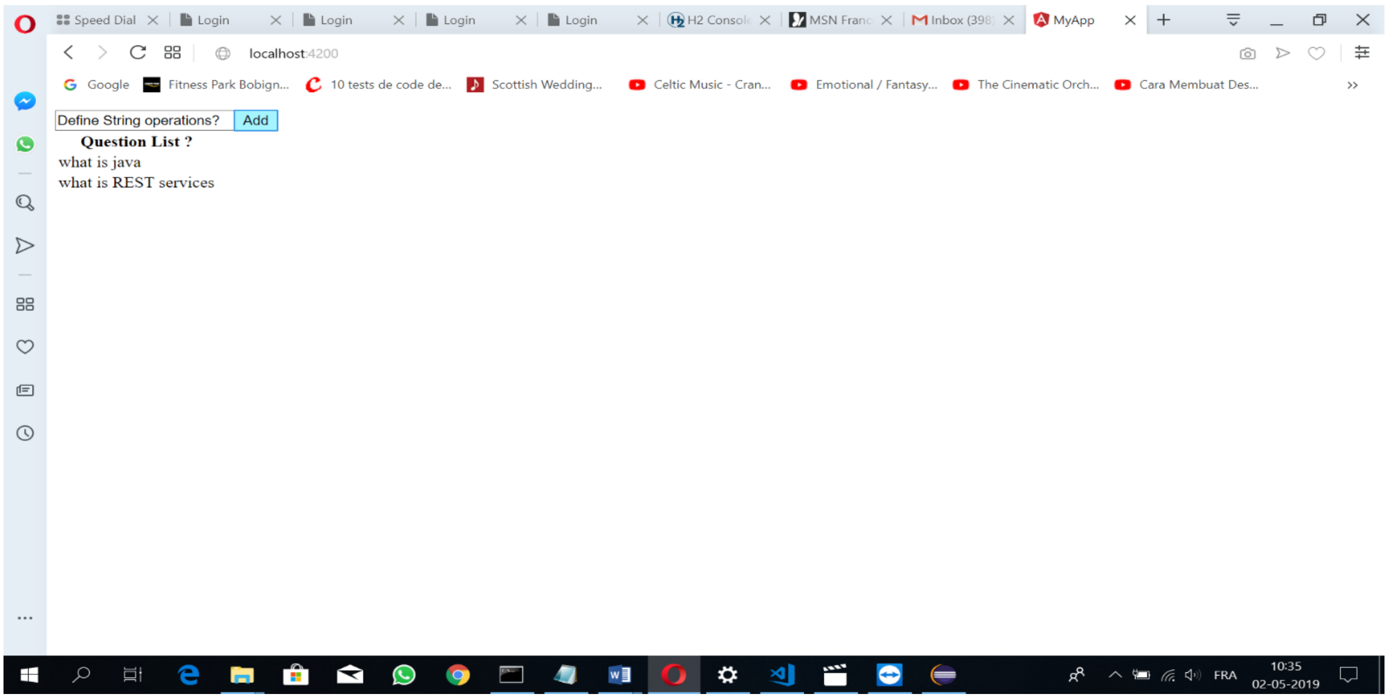
ANGULAR :



```
OPEN EDITORS  1 UNSAVED
ANGULAR2
  e2e
  node_modules
  src
    app
      components
        question-form
          question-form.compone...  U
          question-form.compone...  U
          question-form.compone...  U
          question-form.compone...  U
        question-list
      datamodel
      services
      app-routing.module.ts
      app.component.css
      app.component.html        M
      app.component.spec.ts
      app.component.ts          M
      app.module.ts             M
OUTLINE
```

```typescript
import { Component, OnInit } from '@angular/core';
import { Question } from 'src/app/datamodel/question.model';
import { QuestionService } from 'src/app/services/question.service';

@Component({
  selector: 'app-question-form',
  templateUrl: './question-form.component.html',
  styleUrls: ['./question-form.component.css']
})
export class QuestionFormComponent implements OnInit {

  question : Question = new Question ("");

  constructor(private questionService  : QuestionService) { }

  ngOnInit() {
```

```
6 unchanged chunks
i ｢wdm｣: Compiled successfully.
i ｢wdm｣: Compiling...

Date: 2019-05-02T08:34:17.981Z - Hash: a222517550997a0fe5d6 - Time: 338ms
6 unchanged chunks
i ｢wdm｣: Compiled successfully.
```


Define String operations? Add

Question List ?

what is java

what is REST services

Configuration

Username	: Admin(Case sensitive)
Password	: Admin(Case Sensitive)
Database	: H2
Drivers	: H2 Embedded Driver

Development Environment

PLATFORM USED	:	Windows 7
LANGUAGE USED	:	Core Java
		(Spring framework , Hibernate ,JSP)
IDE	:	Eclipse
DATABASE	:	H2
SERVER	:	Tomcat v9.0

Reference

http://thomas-broussard.fr/