

---

# Quiz System Report

## Independent Study

Qi Chen - April 20, 2016

---

Introduction	2
The design of this system	2
Back-end	2
Servlet	8
Front-end	9
Relationship among back-end, servlet and front-end	9
Some techniques of this system	10
data flow	10
how to cover one line of the code	11
Further Plan	12

---

## Introduction

The Quiz System is used for students to test their java/python ability which combine the QuizJet and QuizPet system.

The improvement of this kind of system compared with old QuizJet and QuizPet system are as followed.

1. combined this two old system into one system, making it easy to maintain.
2. restructured the code in both system and remove the java code in the JSP page as much as possible, making the code more readable and to revise
3. better the interface of the system. for example, the keywords in the test code which is provided for the users is highlighted, making it more readable and comfortable.

The new system is designed and implemented with MVC structure, lowering the coupling, and the following context will discuss how this system works.

## The design of this system

### Back-end

There are three packages: bean, dao, Util, which are used to get the data from database and transfer them into the context which is needed in the front-end.

#### 1. For bean package

This package is designed according to the context in the database and used to encapsulate the data module.

##### 1.1 Jclasses.java

---

```
package bean;
public class Jclasses {
    private int classId;
    private String ClassName;
    private String ClassTester;
    private int QuizID;
    public int getClassId()
    public void setClassId(int classId)
    public String getClassName()
    public void setClassName(String className)
    public String getClassTester()
    public void setClassTester(String classTester)
    public int getQuizID()
    public void setQuizID(int quizID)
}
```

In this class, the information of this class given by quizID and be passed to the front-end completely.

### 1.2. Jquestion.java

```
package bean;
public class Jquestion {
    private int questionID;
    private String title;
    public int getQuestionID()
    public void setQuestionID(int questionID)
    public String getTitle()
    public void setTitle(String title)
}
```

In this class, the information of all quiz's id and title can be passed to the front-end completely, which will be used to the navigation menu.

### 1.3. Jquiz.java

```
package bean;
import java.sql.Blob;
public class Jquiz {
    private int quizeID;
    private String Title;
    private Blob code;
    private String rdfID;
    private int AnsType;
    private int minVar;
    private int maxVar;
    private int privacy;
    private int quesType;
    public int getQuizeID()
    public void setQuizeID(int quizeID)
    public String getTitle()
    public void setTitle(String title)
    public Blob getCode()
    public void setCode(Blob code)
    public int getMinVar()
    public void setMinVar(int minVar)
    public int getMaxVar()
    public void setMaxVar(int maxVar)
    public int getPrivacy()
    public void setPrivacy(int privacy)
    public int getQuesType()
    public void setQuesType(int quesType)
    public String getRdfID()
    public void setRdfID(String rdfID)
    public int getAnsType()
    public void setAnsType(int ansType)
}
```

This class is mainly used to pass the detailed information of Quiz, including its id, code and so on from the database to the front-end.

## 2 For dao package

This package is used to get the information we need from the database and pass them to the servlet with the format which is defined in the bean package

### 2.1 JclassesDAO.java

```

package dao;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import Util.DBConnect;
import bean.Jclasses;
public class JclassesDAO {
    private DBConnect dbc;
    private ResultSet rs;
    public List<Jclasses> getAllClasses(int quizId)
}

```

There is only one function in this class, which is used to get all classes in the quiz. For example, in the java test system(old QuizJet system), some quiz need more than one classes, like BankAccount.class or CashRegister.class, whose id should be found with this JclassesDAO.class.

## 2.2 JquestionDAO.java

```

package dao;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import Util.DBConnect;
import bean.Jquestion;
public class JquestionDAO {
    private DBConnect dbc;
    private ResultSet rs;
    public List<Jquestion> getListQuestion(String language) throws
SQLException
}

```

This class is to list all the quiz questions given the the language the system tests. For example, if this system is used to test java, the getListQuestion function will return all the java questions.if this system is used to test python, the getListQuestion function will return all the python questions.

## 2.3 JquizDAO.java

```

package dao;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import Util.DBConnect;
import bean.Jquiz;
public class JquizDAO {
    private DBConnect dbc;
    private ResultSet rs;
    public List<Jquiz> getListQuizbyQuestionID(int id ) throws
SQLException{}
    public Jquiz getJquiz(String rdfID) throws SQLException{}
    public Jquiz getJquizByQuizId(int quizID) throws SQLException{}
}

```

This class is used to get the complete information of quiz.

- getListQuizbyQuestionID() function can get all quiz information given a question id
- getJquiz() function can get the quiz information given a rdfID
- getJquizByQuizId() function can get the quiz information given a quizID

### 3. For Util package

This package is used to encapsulate the functions which are frequently used in the dao package

#### 3.1 DBConnect.java

This class is to connect database and get data from it.

#### 3.2 Tool.java

```

package Util;
import java.util.ArrayList;
import java.util.List;
public class Tool {

    public static List<Integer> getAllSemicolon(String code)
    public static int calcualteNumberofSpace(String string)
}

```

getAllSemicolon() function is used to return all the indexes of new line give a code, which is used to blanket a line for testers to fill in.

---

calcualteNumberofSpace() function is used to calculate how many spaces before a given string, which is good to make the code needing filling in centered.

### 3.3 Coding.java

```
package Util;
import java.io.File;
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.io.InputStreamReader;
import java.sql.SQLException;
import java.util.ArrayList;
import java.util.List;
import java.util.Random;
import org.apache.commons.io.IOUtils;
import bean.Jquiz;
public class Coding {
    public static String getCode(Jquiz jquiz)
    public static String code2Exec(String code)
    public static String code2fontend(String codeStr)
    public static String[] spilitCode(String code)
    public static String getCmd2compile(boolean window, String
compilerHomeDir, String curActionDir,String testerSurfix, String
language, boolean multiClassFlag)
    public static String getCmd2execute(boolean window, String
compilerHomeDir, String curActionDir,String testerSurfix, String
language, boolean multiClassFlag)
    public static String runCode(String compile, String exec, boolean
multiClassFlag, String curActionDir,String language, String testerSurfix,
Integer queType, int size, ArrayList<String> fileNames,String
extraClassDir) throws IOException, InterruptedException
}
```

This class is mainly used to deal with something relevant to coding.

- getCode(Jquiz jquiz) function is used to get the code part give a quiz
- getCode(Jquiz jquiz) function is used to change the format of code to make it suitable to run, for example, changing the "&lt;" to "<".
- getCode(Jquiz jquiz) function is used to change the format of the code to make it suitable to display on front-end interface. For example, changing the "<" to "&lt;"
- getCmd2compile function is used to return the command string how to compile the code
- getCmd2execute function is used to return the command string how to execute the code
- runCode function is used to run the code and return the answer.

---

### 3.4 FileOperation.java

```
package Util;
import java.io.File;
import java.io.FileWriter;
import java.io.IOException;
import java.sql.SQLException;
import java.text.DateFormat;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Random;
import javax.servlet.ServletOutputStream;
import org.apache.tomcat.util.http.fileupload.FileUtils;
public class FileOperation {
    public static String replace = "___";
    public static char replaceChar = replace.charAt(0);
    public static String createActionFilePath(String appDir, String
compilerHomeDir, String usr, boolean window)
    public static String getExtraClassDir4Multiclasses(String appDir,
boolean window)
    public static String createTempFile(String curActionDir, boolean
multiClassFlag, int queType, String codePart,String language,
ServletOutputStream out)
    public static void deleteTempFiles(String tempFileName, String
curActionDir, String surfix)
    public static String stringTransfer(String string)
    public static void log(String str)
}
```

This class is used to do something relevant to operate file.

- createActionFilePath function is used to return a file path string to create a temporary java file
- getExtraClassDir4Multiclasses is used to return the file paths strings to create other java files for extra classes
- createTempFile is used to create a temporary java file
- deleteTempFiles is used to delete the temporary java file which is created by previous function
- stringTransfer function is used to make the answer which the code runs more nice looking
- log function is used to report some errors or records

## Servlet



This middle layer(servlets) is used to get the data from the back-end and then pass them to the front-end.

it includes three files: MenuNavigation.java, displayQuiz.java and EvaluationAnswer.java

## Front-end

The following is the interface of Quiz System.



The screenshot displays the QuizJET Java Evaluation Tool interface. The top navigation bar includes 'Home' and 'Help' links, and the 'QUIZJET javaEvaluationtool' logo. A left sidebar lists various categories: Variables, Classes, Primitive\_Data\_Types, Constants, Arithmetic\_Operations, Strings, Decisions, Boolean\_Expressions, Switch, Loops\_while, Loops\_do\_while, and Loops\_for. The main content area shows a question: 'Question is : Fill the blank to make the final value of result: 36.5'. Below the question is a code editor with the following Java code:

```
public class Tester {
    public static void main(String[] args) {
        CashRegister register = new CashRegister();

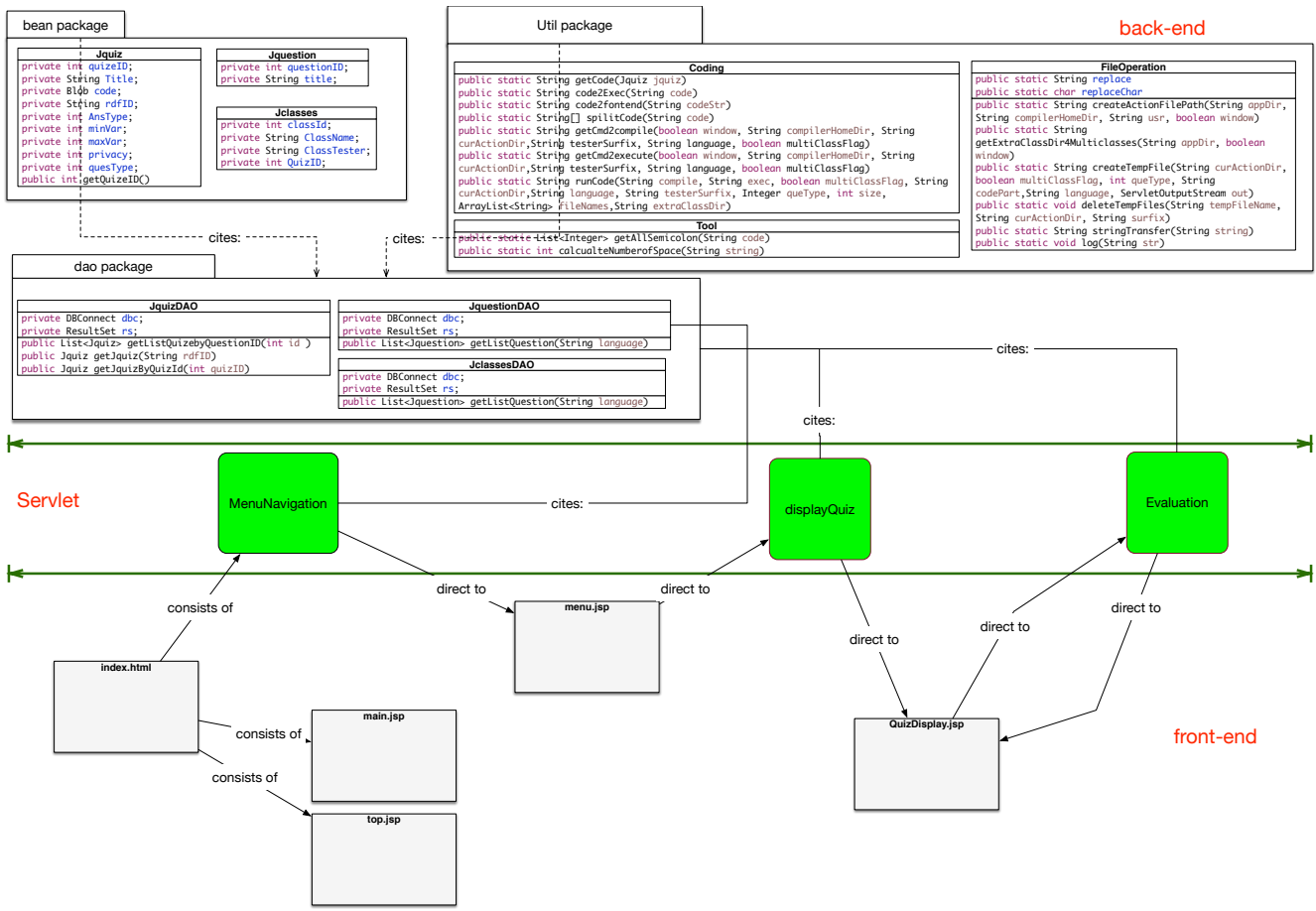
        register.enterPayment(50.5);
        double result = register.giveChange();
    }
}
```

A 'Submit' button is located at the bottom right of the code editor.

The front-end consists of 5 files: index.html, main.html, menu.jsp and QuizDisplay.jsp

## Relationship among back-end, servlet and front-end

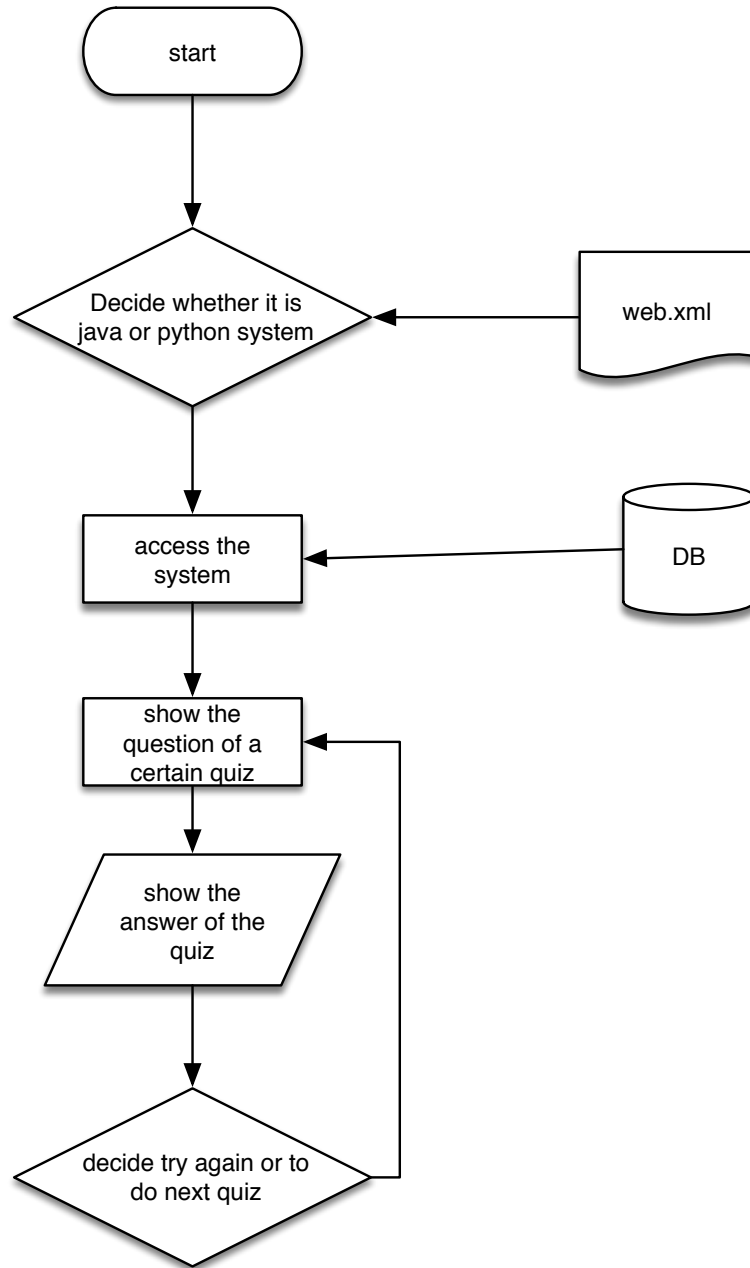
The relationship can be seen from the following graph.



## Some techniques of this system

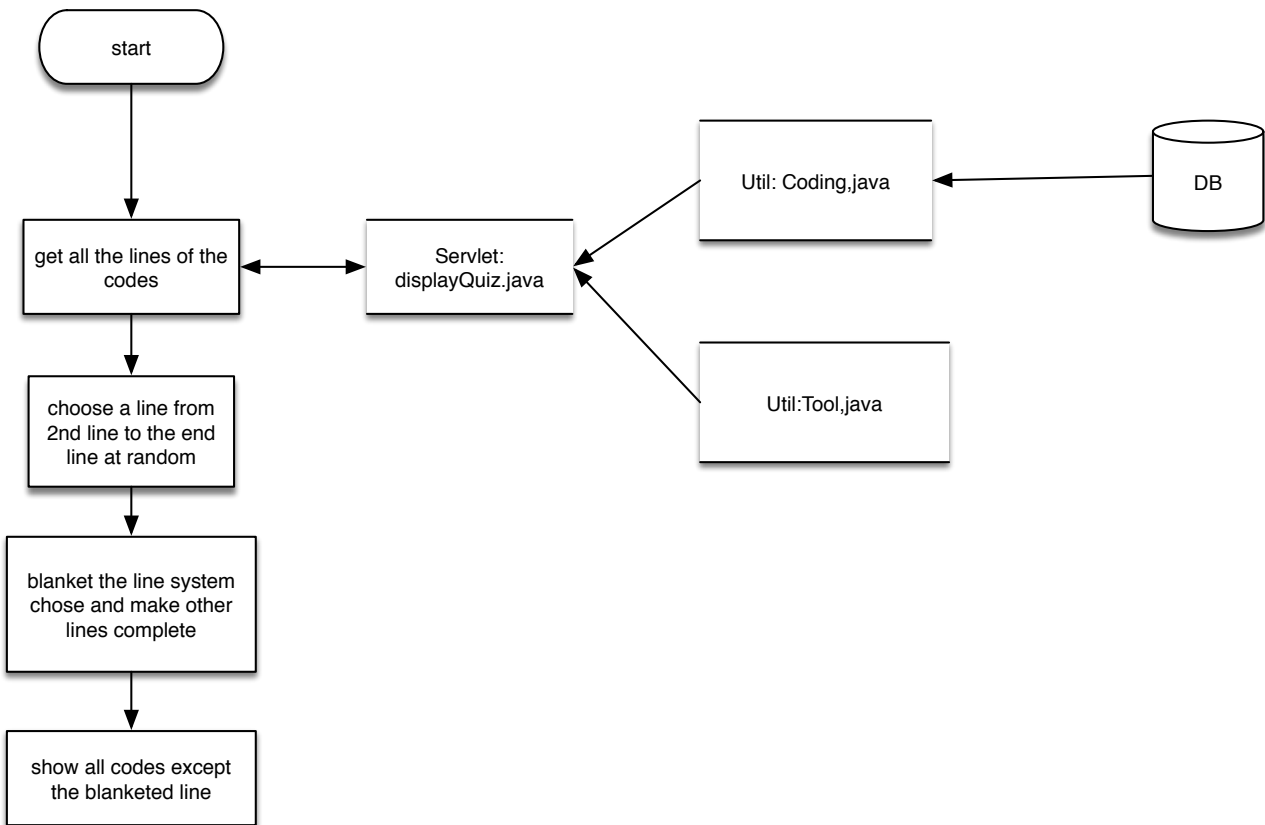
### data flow

The data flow of this system is as followed.



### how to cover one line of the code

When showing the code, the procedure of how to decide which part is needed to be filled in is as followed.



## Further Plan

The system can work in general, but there is something that needs revising.

1. This system reads code from database, process it and write it on the interface, but the formats of code in the database is not always same.

For example, the normal view of the quiz question is as followed.

jClass1 03BankAccount.java

**Question is :**

Fill the blank to make the final value of **result**:

**447.0**

```
public class Tester {
    public static void main(String[] args) {
        BankAccount myBankAccount = new BankAccount(447.0);
        the place where you need to fill in _____
    }
}
```

But some quiz's interface is different.

jInterfaces1 09Car.java 09Computer.java 09Mechanism.java

**Question is :**

Fill the blank to make the output

**4.0**

**0**

**Honda**

**0**

```
public class Tester {
    public static void main(String[] args) {
        the place where you need to fill in _____
        Mechanism mech2 = new Car("Honda", 4);

        Computer comp = (Computer) mech1;

        System.out.println(comp.getProcessorSpeed());
        System.out.println(comp.reportProblems());

        System.out.println(((Car) mech2).getBrand());
        System.out.println(mech2.reportProblems());
    }
}
```

The difference of above two pictures is that, the texts in every line are not alined. The reason why it occurs is that some codes in the database starts with tabulator key instead of space. It can be solved by changing some contents whose attribute is "Code" in the ent\_jquiz table in

the database.

QuizID	AuthorID	GroupID	Title	Description	Code	MinVar	MaxVar	AnsType	Privacy	rdfID	Q
1	6	2	jWhile1	While Loop	BLOB	4	10	3	1	jwhile1	1
2	6	2	jWhile2	While Loop2	BLOB	1	5	3	1	jwhile2	1
6	6	2	jInteger1	basic integer	BLOB	1	8	3	1	jinteger1	1
8	6	2	jString1	basic String literals	BLOB	3	9	7	0	jpri_string1	2
10	6	2	jWhile3		BLOB	1	6	3	1	jwhile3	1
11	6	2	jWhile4		BLOB	2	6	3	1	jwhile4	1
58	6	2	jDouble1		BLOB	0	9	8	1	jdoube1	1
59	6	2	jClass1	print getter	BLOB	100	500	8	1	jclass1	1
60	6	2	jBankAccount		BLOB	100	300	8	1	jbankaccount	1
61	6	2	jBa_ques		BLOB	1	9	8	1	jba_ques	1
62	6	2	jDowhile1	dowhile1	BLOB	2	8	3	1	jdowhile1	1
63	6	2	jDowhile2	dowhile2	BLOB	1	5	3	1	jdowhile2	1
64	6	2	jDowhile3	dowhile3	BLOB	10	15	3	1	jdowhile3	1

The changes we need do is to delete tabulator key and add some spaces at the front of each line in codes.

These changes should work on the following contexts whose attribute is "Title":

- Class (Bank Account This)
- jConstants
- Division
- String(equals)
- String(replace)
- Earthquake
- jSwitch1
- jInterfaces1
- jInterfaces2
- jInterfaces3
- jInterfaces4
- jInterfaces5

2. Add one more line at the end of "Code" in "Code" attribute in the ent\_jquiz table.

In our system, it add out print line of code at the end of main method in java system, and some codes do not has enough space to be added this out print code. just as shown below.

```
1 public class Tester {
2
3     public static void main(String[] args) {
4
5         Double d1 = _Param.5;
6         Double d2 = new Double(5._Param);
7         double d3 = _Param._Param;
8
9         Integer i = (int) (d1 + d2 + d3);
10
11         int result = i;
12     }
13 }
14
15
```

Thus what we should do is to add one space line under "int result=i;" code.

```

1 public class Tester {
2
3     public static void main(String[] args) {
4
5         Double d1 = _Param.5;
6         Double d2 = new Double(5._Param);
7         double d3 = _Param._Param;
8
9         Integer i = (int) (d1 + d2 + d3);
10
11         int result = i;
12     }
13 }
14
15
16

```

The same changes should be done on the tuples whose attributes is title:

- jWarpper\_Class3
- j2D\_Arrays1
- j2D\_Arrays2(the max/min value of the param should also be fixed)

3. prevent user's malicious code

I do not have a good solution how to prevent user's malicious code when they are coding. For example, we should try to stop users write a dead loop when they are testing.

jClass1    03BankAccount.java

**Question is :**

Fill the blank to make the final value of **result**:

**447.0**

```

public class Tester {
    public static void main(String[] args) {
        BankAccount myBankAccount = new BankAccount(447.0);
        while(true){}
    }
}

```