# User Guide

Author : Juyeon Lee , Moeun Son

Date : 19.02.2019

Address of Github : <https://github.com/Juyeon-Lee/Epita_Java_project.git>

## Main features

The scope that we covered by this application is to :

* Database ‘H2’ for store and manage data
* Be able to operate Create/Update/Delete/Read of all Questions which are stored in a database
* Use console application to play the program
* Be able to search questions based on topics
* Be able to assemble automatically a quiz (a quiz is a set of questions) that gathers all the questions covering a given list of topics.
* Export this quiz under a plain text format if user want to export
* Run the evaluation and provide the automatic mark in the end of this execution[[1]](#footnote-1)

## Compilation & Installation Notice

Before start using our application, the quiz manager works on the eclipse(console) and the database uses H2.

1. Go to the downloaded h2 folder and follow the instructions

[ h2 - bin - h2.bat(click) - connet the h2 console

jdbc url : jdbc:h2:~/test

user name : sa ]

2. Copy queries in 'sturcture\_question.sql' at sql folder in our project file and paste them into the h2 console, then execute.

3. Copy queries in 'sampledata\_question.sql' at sql folder in our project file and paste them into the h2 console, then execute.

4. Finally, The initial data is saved after execution.

5. Terminate the h2 console.

6. Right click of our project file, and go run as-run configuration-arguments-VM arguments, then enter "-Dconf.location=app.properties"

6. Run the Launcher at eclipse. (press ‘ctrl+F11’)

## Order of Function

When you run the program, there will be two parts; admin questions/student(solve a quiz).

The ‘admin questions’ part will be able to operate CRUD on Open Questions and MCQ Questions, and the ‘student(solve a quiz)’ part is for quizzes which questions are collected by nothing(so all questions), topics, and difficulty levels in them.

Let’s talk about the ‘admin questions’ part. First, a manager can create/update/delete both open question and mcq question as many as he or she wants. Before update or delete, the manager can read a list of questions. It will include all information of questions. This function is same with ‘show all’. It will print everything in console. Plus, a manager can search questions by a topic.

Next is the ‘student(solve a quiz)’ part. First, a student can choose to one's quiz by nothing, topic, or difficulty and solve the quiz. Second, student can choose to extract the quiz to a file after every quiz is finished. Finally, the student scores are printed and the quiz ends.

## Scenario of Data Management

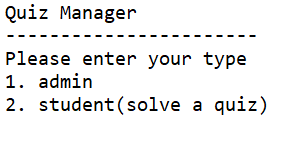


Figure 1

1. [Figure 1] is start page.
2. If I enter ‘1’, the [Figure 2] is admin questions page. You should enter number from 1 to 5.

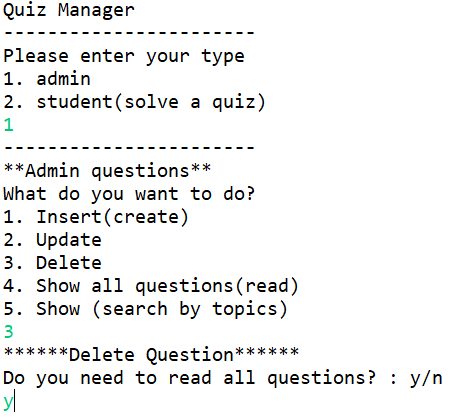


Figure 2

1. Insert mode page [Figure 3]

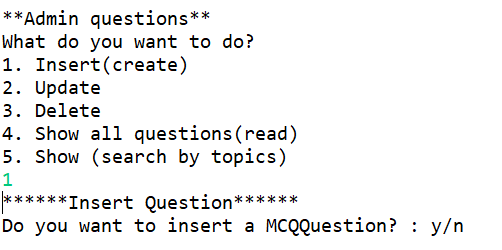


Figure 3

1. [Figure 4] You can insert a open question or a mcq question. Many questions are more than one line, so you should enter ‘end’ when you want to finish entering the question.

You can enter several topics, and you have to enter ‘end’ when you want to finish entering it.

Because I answered ‘y’ to the question, I would enter a mcq question and enter the choices and answer. Otherwise, I would not see a screen to enter the choices and answer.

After the insertion is successful, a message and logs would appear, like “insert success” and “a question is created.” Then the program would ask me, “Do you want to insert again?”. If you answer “y”, you can insert again and again.

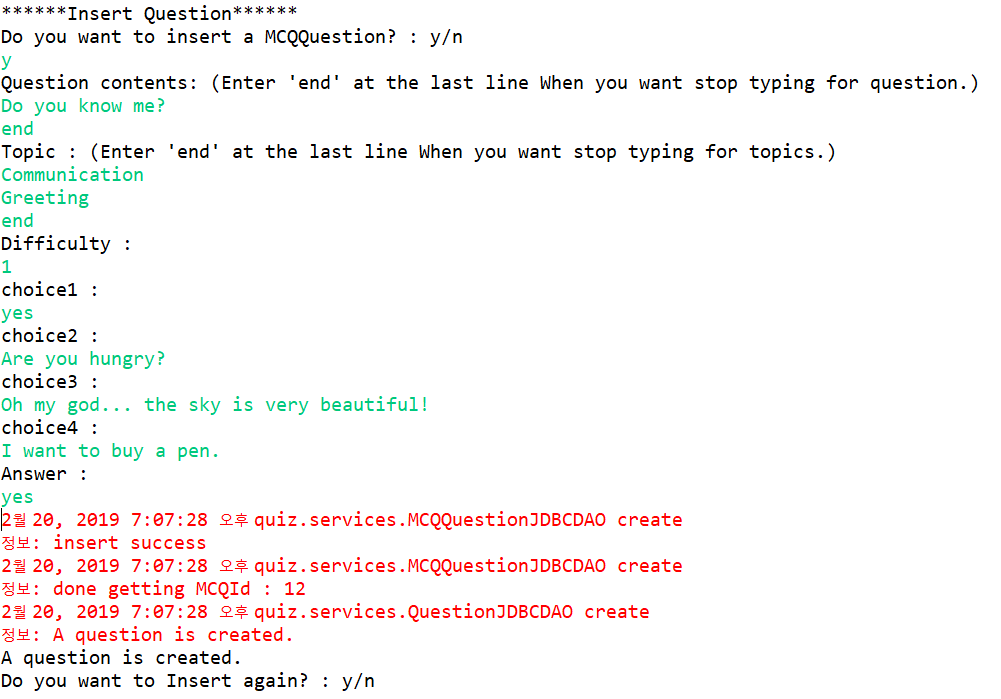


Figure 4

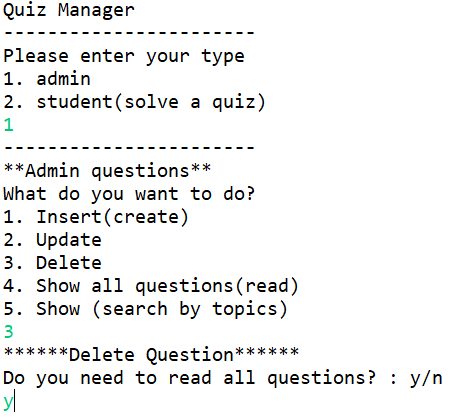
1. Update and delete is almost similar. About question “What do you want to do?”, if you answer “2”, you’d go to update page, or if you answer “3”, you’d go to delete page.

Figure 5

At the [Figure 5], there is a delete page.

Before update or delete, you can see a list of questions if you want. You should enter ‘y’ or ‘n’.

1. You can read all questions, because you enter ‘y’. After all questions, you can delete by following the form. [Figure 6]

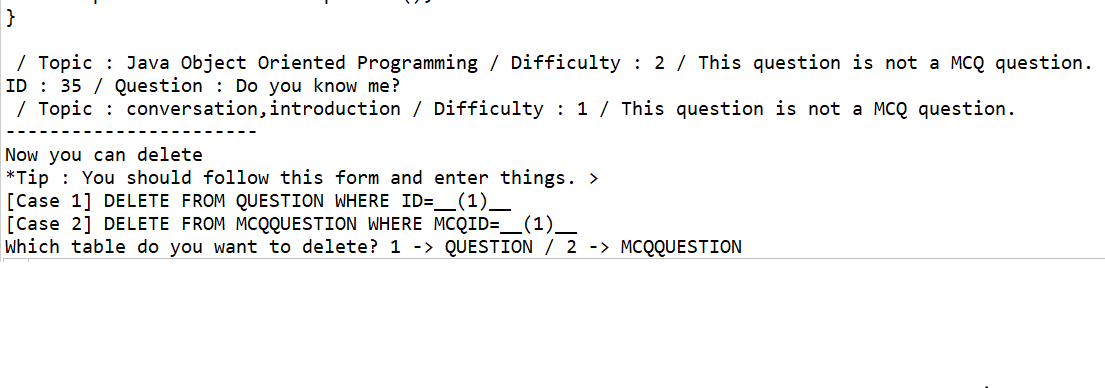


Figure 6

You should select table by answering as ‘1’ or ‘2’.

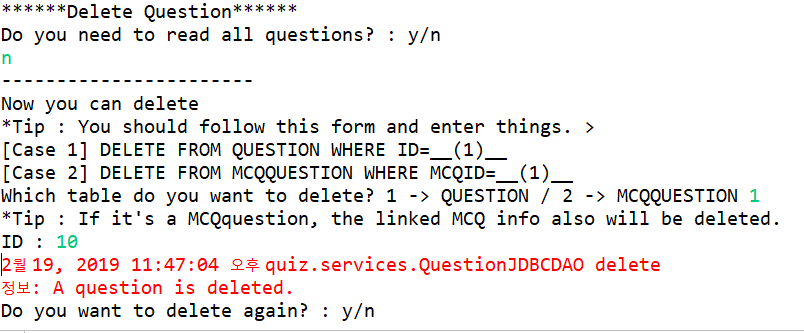
1. After select a table which the data you want to delete is in it, you can delete a row by entering a id. If it’s a MCQquestion, the linked MCQ info also will be deleted. Once it’s successful, there will be a sentence “A question is deleted.” Then if you want to delete another again, you must enter ‘y’. [Figure 7]

Figure 7

1. This picture is for solving a quiz. At the start page, if you answer ‘2’, you can solve a quiz. Before start of the quiz, you should enter your name and id. Then select a type of quiz. At the picture, there was a quiz by difficulty1. [Figure 8]

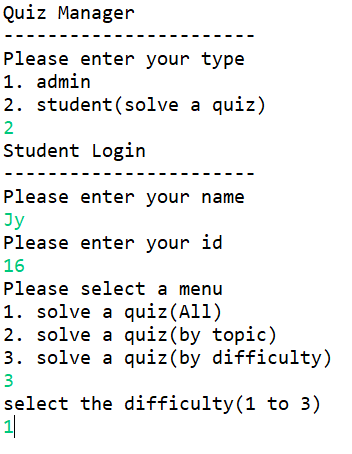


Figure 8

1. After finishing the quiz, you can extract the quiz to a file, and see your score of the quiz. [Figure 9]

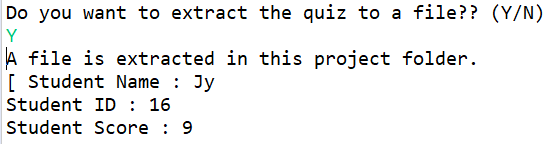


Figure 9

## Schemas

We have two tables in database, which are ‘Question’ and ‘MCQQuestion’. At table Question, the primary key is ID and it’s integer. Also, there are QUESTION and TOPIC of varchar type, and there are MCQ and DIFFICULTY of integer type.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Question** | | | | |
| INT | VARCHAR(1024) | INT | VARCHAR(60) | INT |
| ID | QUESTION | MCQ | TOPIC | DIFFICULTY |

At table MCQQuestion, the primary key is MCQID and it’s integer. Also, there are CHOICE1, CHOICE2, CHOICE3, CHOICE4, and ANSWER of varchar type.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **MCQQuestion** | | | | | |
| INT | VARCHAR(1024) | VARCHAR(1024) | VARCHAR(1024) | VARCHAR(1024) | VARCHAR(1024) |
| MCQID | CHOICE1 | CHOICE2 | CHOICE3 | CHOICE4 | Answer |

## Sample Data

We had set up sample data in database, like under two table. There are about 10 sample data for each table.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Question | | | | |
| INT | VARCHAR(1024) | INT | VARCHAR(60) | INT |
| **ID** | **QUESTION** | **MCQ** | **TOPIC** | **DIFFICULTY** |
| 1 | **What will happen when you compile and run the following code?**  public class Test{    public static void main(String[] args){    double d1 = Double.NaN;  double d2 = d1;    if(d1 == d2)  System.out.println("Equal");  else  System.out.println("Not Equal");  }    } | 1 | Java Language Basics | 1 |
| 2 | **What will happen when you compile and run the following code?**  public class Test{    public static void main(String[] args){  int i = 10;  int j = 25;  System.out.println(i + ' ' + j);  }  } | 2 | Java Language Basics | 1 |
| 3 | **What will happen when you compile and run the following code?**  public class Test {    public static void main(String[] args){    int i = 10;  int j = 12;  i+=++j;  System.out.println(i);  }  } | 3 | Java Language Basics | 1 |
| 4 | **What will happen when you compile and run the following code?**  public class Test {    public static void main(String[] args){  int i = 010;  System.out.println(i);  }  } | 4 | Java Language Basics | 1 |
| 5 | **What will happen when you compile and run the following code?**  public class Test{    public static void main(String[] args) {  int i = 0;  int j = i++ + ++i;  System.out.println( j );  }  } | 5 | Java Operators | 1 |
| 6 | **What will happen when you compile and run the following code?**  public class Test{    public static void main(String[] args) {  int i = 0;  int j = ++i + i++;  System.out.println( j );  }  } | 6 | Java Operators | 1 |
| 7 | **What will happen when you compile and run the following code?**  public class Test{    public static void main(String[] args) {  int i = 100;  int j = 10;  System.out.println( i%j );  }  } | 7 | Java Operators | 1 |
| 8 | **What will happen when you compile and run the following code?**  public class Test{    public static void main(String[] args) {  int i = 0;  int j = 1;  if(!i && j)  System.out.println("1");  else  System.out.println("2");  }  } | null | Java Operators | 1 |
| 9 | **What will happen when you compile and run the following code?**  class One{  int i = 1;  public int getInt(){  return i;  }  }  class Two extends One{  int i = 2;  public int getInt(){  return i;  }  }  public class Test{  public static void main(String[] args) {  One one = new One();  Two two = (Two)one;  System.out.println( two.getInt() );  }  } | 8 | Java Object Oriented Programming | 3 |
| 10 | **What will happen when you compile and run the following code?**  class One{  public static void print(){  System.out.println("1");  }  }  class Two extends One{  public static void print(){  System.out.println("2");  }  }  public class Test{  public static void main(String args[]){  One one = new Two();  one.print();  }  } | 9 | Java Object Oriented Programming | 3 |
| 11 | **What will happen when you compile and run the following code?**  class One{  public static void print(int i){  System.out.println("Parent");  }  }  class Two extends One{  public static void print(byte b){  System.out.println("Child");  }  }  public class Test{    public static void main(String args[]){  One one = new Two();  one.print(10);  }  } | 10 | Java Object Oriented Programming | 3 |
| 12 | **Will this code compile successfully?**  class One{  public void process(){  System.out.println("Parent");  }  }  public abstract class Test extends One{  public abstract void process();  } | null | Java Object Oriented Programming | 2 |
| 13 | **What’s your university name?** | null | conversation, introduction | 1 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| MCQQuestion | | | | | |
| INT | VARCHAR(1024) | VARCHAR(1024) | VARCHAR(1024) | VARCHAR(1024) | VARCHAR(1024) |
| **MCQID** | **CHOICE1** | **CHOICE2** | **CHOICE3** | **CHOICE4** | **Answer** |
| 1 | Equal | Not Equal | The code will not compile | The code will compile but will throw run time error | Not Equal |
| 2 | 10 25 | 35 | 67 | The code will not compile | 67 |
| 3 | 13 | 23 | 22 | 12 | 23 |
| 4 | Compilation error | Runtime error | 8 | 10 | 8 |
| 5 | 1 | 3 | 2 | 4 | 2 |
| 6 | 3 | 2 | 4 | 1 | 2 |
| 7 | 10 | 0 | Compilation error | None of the above | 0 |
| 8 | 1 | 2 | Compilation error | Runtime error | Runtime error |
| 9 | 1 | 2 | Compilation error | Runtime error | 1 |
| 10 | Parent | Child | Compilation error | Runtime error | Parent |

1. https://thomas-broussard.fr/work/java/courses/project/fundamental.xhtml [↑](#footnote-ref-1)