BTech451
Final Presentation

Dian LIN

Supervised By Dr. Wannes van der Mark, Dr. Tariq Khan

Code Runner Extension

Code Runner is a free and open-source question-type in Moodle that allows students to answer programing assignments.

Introduction: Push up barriers(code implementations) into Code Runner against cheating.

Motivation: The Functionality of cheating detection is not covered yet by Code Runner. Which may cause:

1. Being unfair to hard-working students
2. A loss of faith in Code Runner
3. Students try to cheat in Code Runner

Recall

- Studies have been done in first semester:

1. Two research:

    -Similarity checking **without comments*** (not feasible, normally high similarity occurs)

    -Programming variation(Hard to be detected by Code Runner)

2. Anti-cheat idea testing (Java): Each student being assigned unique option(question)

    -Database schema validation

    -Prototype build-up

Reason of testing: Code Runner is a plugin of Moodle, implementing the idea without testing would possibly destroy Moodle as well as the database.

*** What Code Runner works currently does not require students to add comments**

Research Process

(Similarity Checking without Comments)

Length of Program	Similarity percentage
3 lines (shortest)	100%
3 - 67 lines	68% - 77%
67 lines(longest)	56.6%
Median Length: 30 lines	Median similarity:76.7%(relatively high)

Analysis: High similarity would occur due to short coding length, same question and example provided in lectures.

Conclusion: Similarity checking is not a feasible anti-cheating approach because of above reasons.

Research Process
(Programming variation)

- Question types being selected:

1. recursive question: summing numbers.
2. pre-define question: binary tree.
3. sorting question: bubble-sort, insertion-sort and merge-sort.

- **Analysis**:

1. Solutions of recursive and sorting questions are both easily found online, but pre-define question are not possible since Code Runner define Class for participants.
2. Programming languages translations is an unavoidable personal skill which is not able to be detected by Code Runner.

Conclusion: pre-defined questions become a preferred question creation way but can't prevent copy-paste cheating.

Second semester studies

- **One research**: Question recycling. Aims to find out how many or what percentage of old questions in Code Runner have been reused.

- **Two implementations**:
  1. Unique questions(options) being assigned to different students.(prototype from Semester 1)
  2. Similarity checking **with comments*:** checking between submissions

- **Requirement:** PHP programming language

***When comments are required to be added in Code Runner**

Research process

(Question Recycling)

Target: one of Stage one Computer Science courses.

	Number of Questions	Percentage	Number of Test Cases	Percentage
Reused	35	85%	34	83%
New Defined	6	15%	7	17%
Total	41	100%	41	100%

Analysis: most questions have been recycled without changing test cases. A high probability of recycling indicates a low anti-cheat performance.

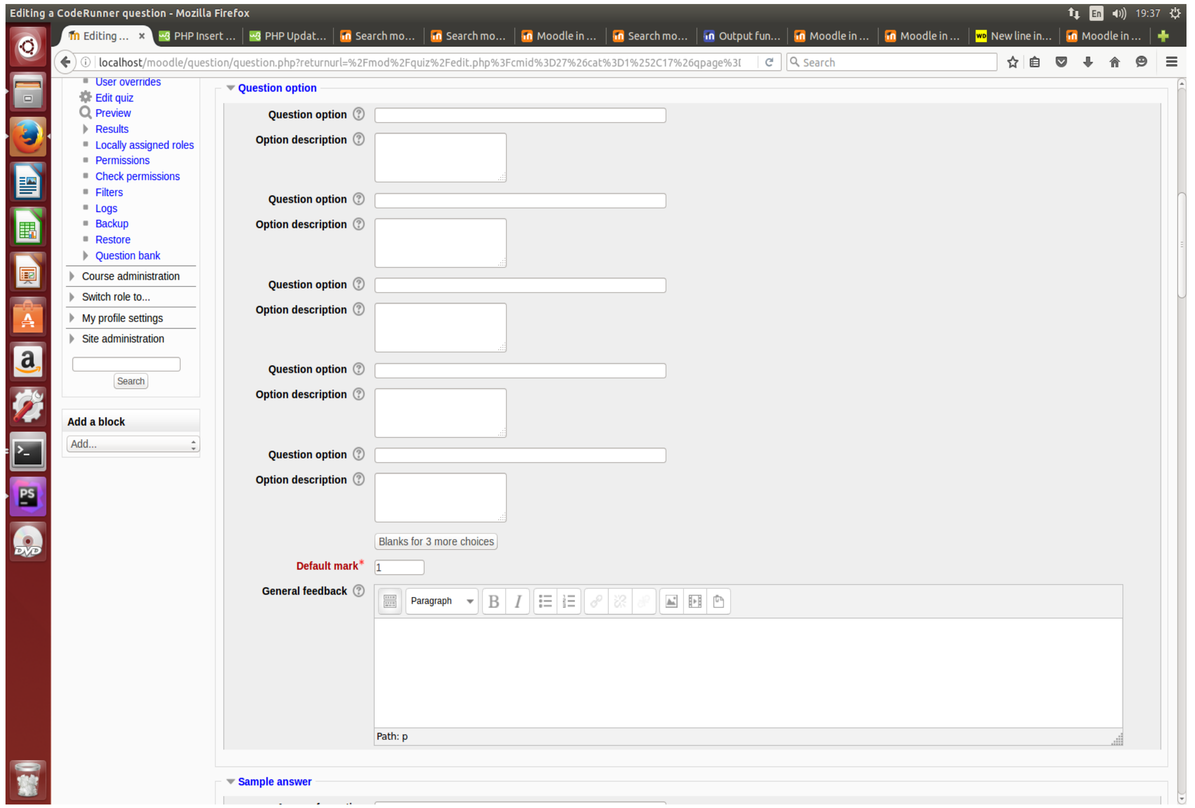
Conclusion: Question recycling provides an invisible way for cheaters to get an unfair pass. It can be solved by restricting students' access after they have completed the course.

Code Runner Implementation 1 (Personalized assessment)

- **Idea**: Instead of assigning different questions, being assigned options.
- **Description**: Creating several options under one question description, then assigning random options to random students.
- **Why option:** Even difficulty guarantees a fair system.
- **Different from Question Bank**: Question bank basically defines different questions, which implies that different difficulties may leader to unfair assignments.

Personalized assessment
(Continued)

Most of user interface and functionalities should be inherited. For UI, one more field called "Question options" will be added, and corresponding text area will also be added in the rest parts in question creation page to specify test cases and sample answers for each options.



Editing a CodeRunner question - Mozilla Firefox

localhost/moodle/question/question.php?returnurl=%2Fmod%2Fquiz%2Fedit.php%3Fcmid%3D27%26cat%3D1%252C17%26qpage%3I

Question option

Question option
Option description
Question option
Option description
Question option
Option description
Question option
Option description
Question option
Option description

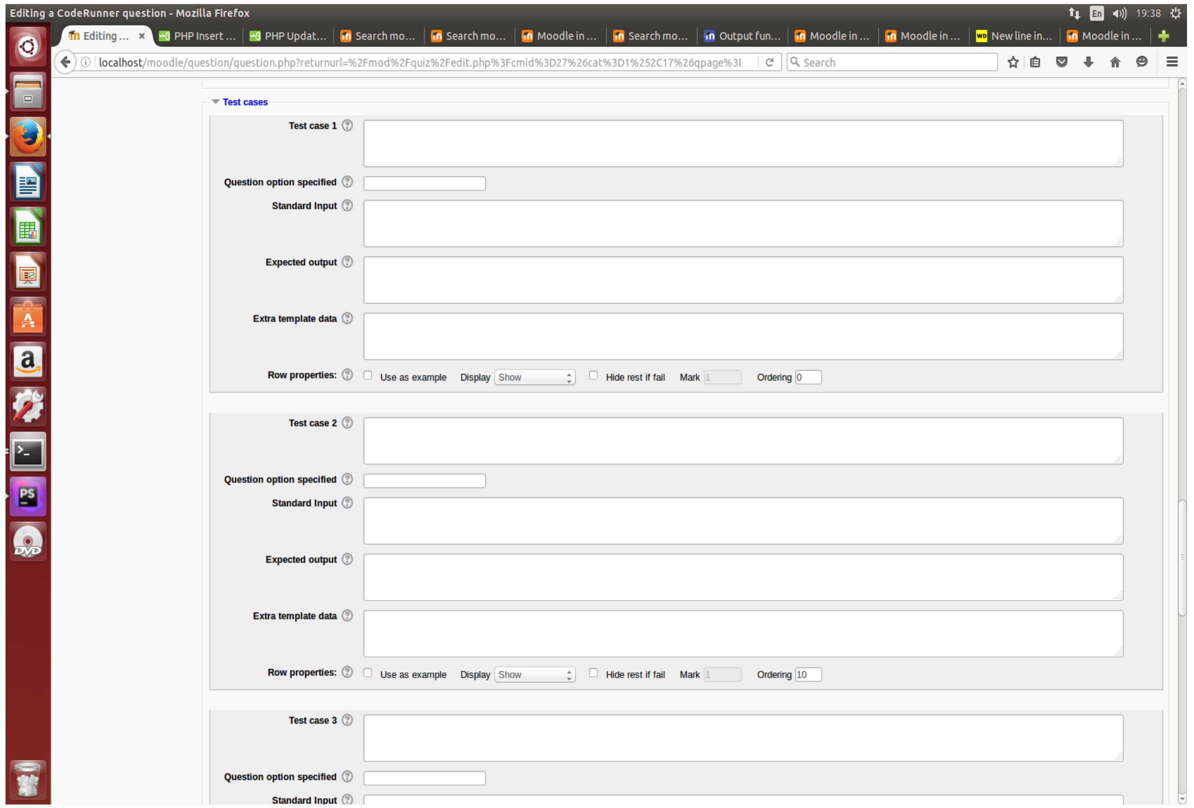
Blanks for 3 more choices

Default mark* 1

General feedback

Path: p

Sample answer



Editing a CodeRunner question - Mozilla Firefox

localhost/moodle/question/question.php?returnurl=%2Fmod%2Fquiz%2Fedit.php%3Fcmid%3D27%26cat%3D1%252C17%26qpage%3I

Test cases

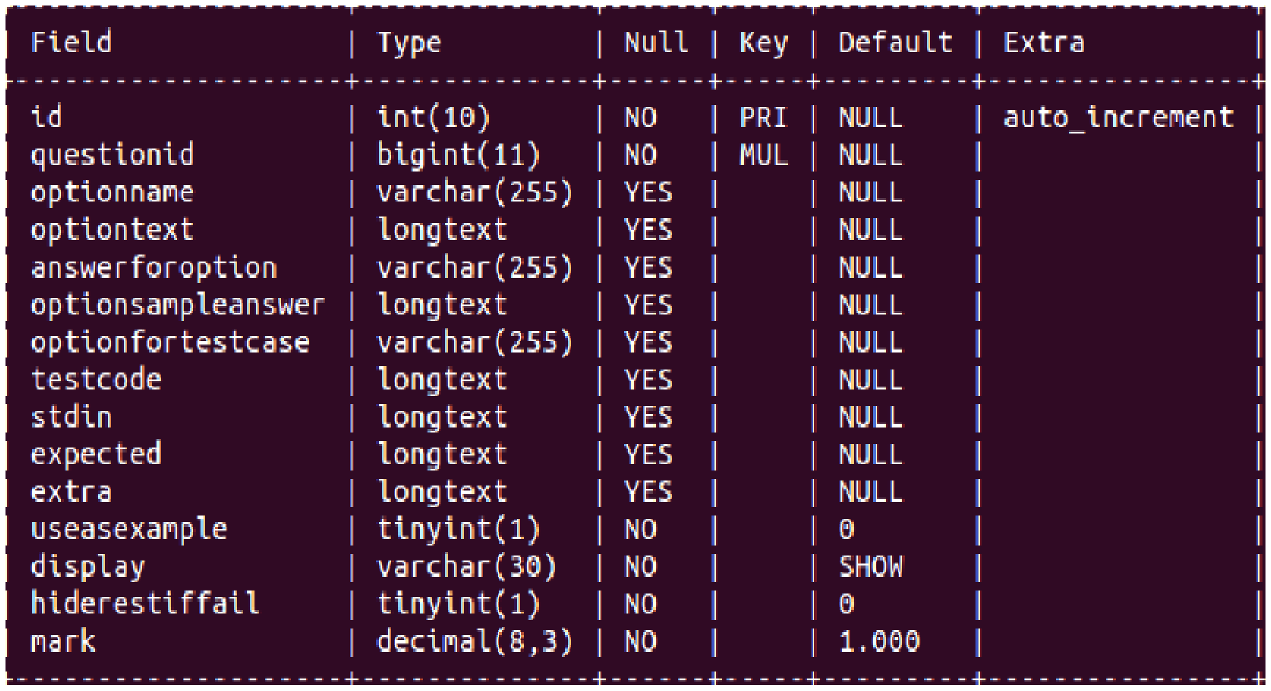
Test case 1
Question option specified
Standard Input
Expected output
Extra template data
Row properties: Use as example Display Show Hide rest if fail Mark 1 Ordering 0

Test case 2
Question option specified
Standard Input
Expected output
Extra template data
Row properties: Use as example Display Show Hide rest if fail Mark 1 Ordering 10

Test case 3
Question option specified
Standard Input

Personalized assessment
(Continued)

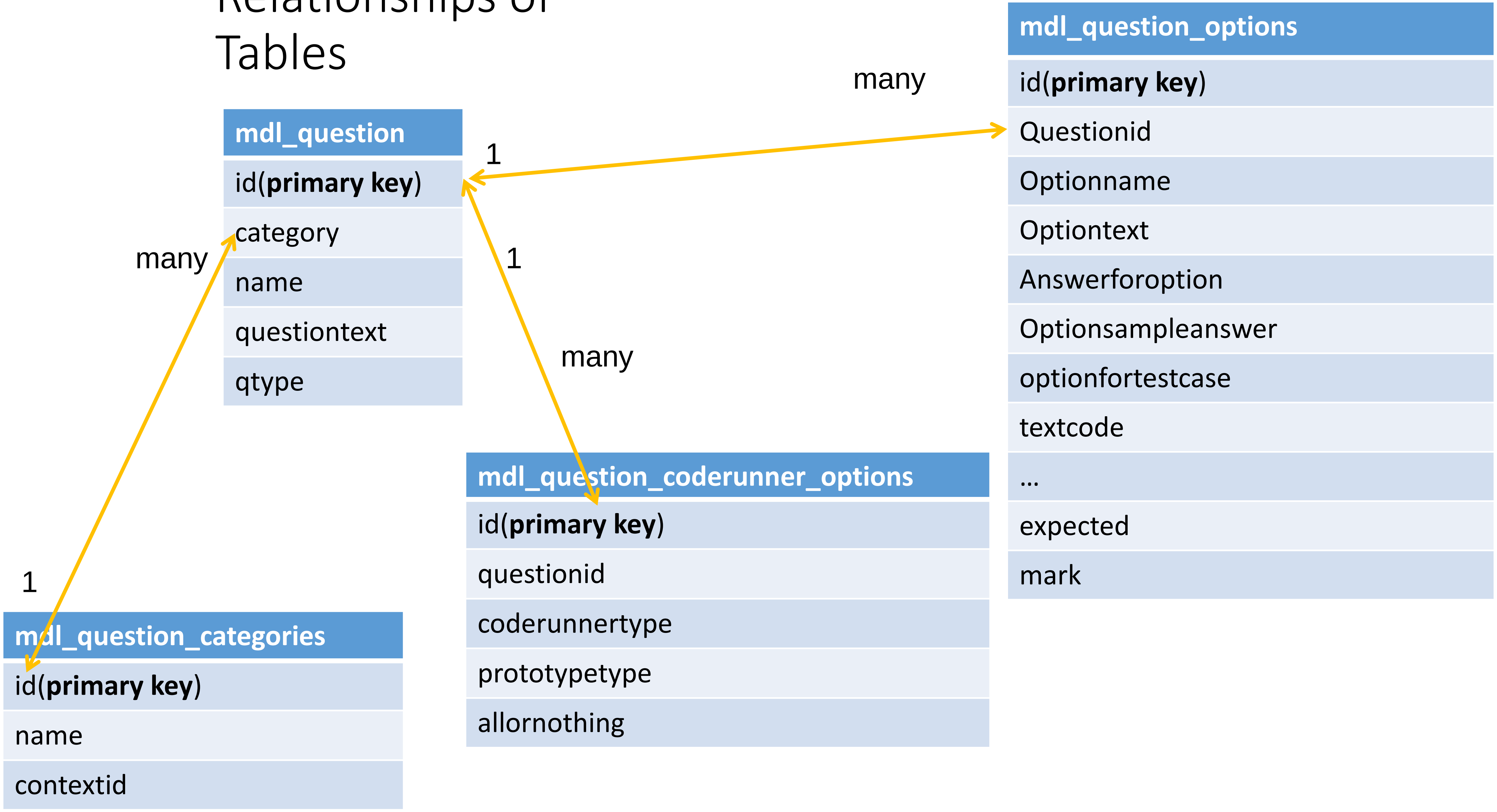
- In order to make the UI functional, user input should be inserted into Moodle database for further use when students attempt options.
- New database schema needs to be defined.
- All details of test cases will store in new database table.
- All functions used to relate to “testcases” table will have to redirected to new database table,
- Mdl_question_options
- New relationships are built up.



```
+-------------------+---------------+------+-----+---------+----------------+
| Field             | Type          | Null | Key | Default | Extra          |
+-------------------+---------------+------+-----+---------+----------------+
| id                | int(10)       | NO   | PRI | NULL    | auto_increment |
| questionid        | bigint(11)    | NO   | MUL | NULL    |                |
| optionname        | varchar(255)  | YES  |     | NULL    |                |
| optiontext        | longtext      | YES  |     | NULL    |                |
| answerforoption   | varchar(255)  | YES  |     | NULL    |                |
| optionsampleanswer| longtext      | YES  |     | NULL    |                |
| optionfortestcase | varchar(255)  | YES  |     | NULL    |                |
| testcode          | longtext      | YES  |     | NULL    |                |
| stdin             | longtext      | YES  |     | NULL    |                |
| expected          | longtext      | YES  |     | NULL    |                |
| extra             | longtext      | YES  |     | NULL    |                |
| useasexample      | tinyint(1)    | NO   |     | 0       |                |
| display           | varchar(30)   | NO   |     | SHOW    |                |
| hiderestiffail    | tinyint(1)    | NO   |     | 0       |                |
| mark              | decimal(8,3)  | NO   |     | 1.000   |                |
+-------------------+---------------+------+-----+---------+----------------+
```

Mdl_question_options

Relationships of Tables



mdl_question
id(**primary key**)
category
name
questiontext
qtype

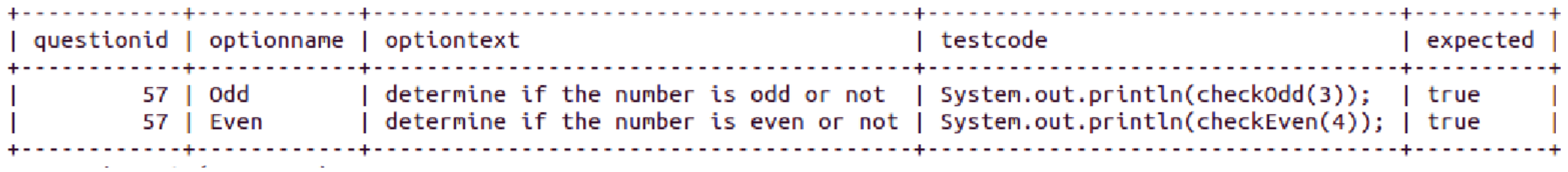
mdl_question_categories
id(**primary key**)
name
contextid

mdl_question_coderunner_options
id(**primary key**)
questionid
coderunnertype
prototypetype
allornothing

mdl_question_options
id(**primary key**)
Questionid
Optionname
Optiontext
Answerforoption
Optionsampleanswer
optionfortestcase
textcode
…
expected
mark

Personalized assessment
(Continued)

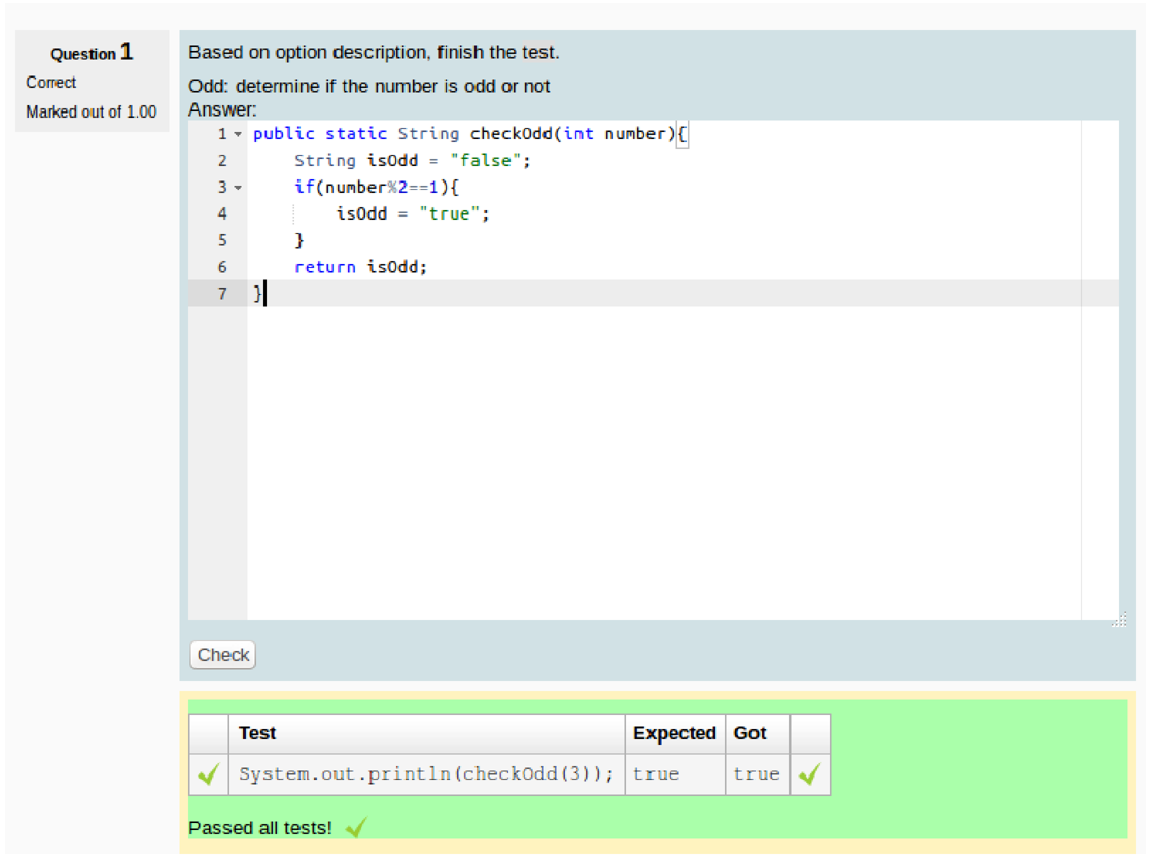
- After two options, Odd and Even, were saved from UI by new anti-cheat system, all details about each options should be inserted into new database table with current question ID. Shown below:



```
+------------+------------+--------------------------------------+------------------------------------+----------+
| questionid | optionname | optiontext                           | testcode                           | expected |
+------------+------------+--------------------------------------+------------------------------------+----------+
|         57 | Odd        | determine if the number is odd or not | System.out.println(checkOdd(3));  | true     |
|         57 | Even       | determine if the number is even or not | System.out.println(checkEven(4)); | true     |
+------------+------------+--------------------------------------+------------------------------------+----------+
```

Personalized assessment
(Continued)

- As new anti-cheat system maintains the functionality from Sandbox which is used to run the students answers based on test cases. After student view of question has been changed to option view, Sandbox works in the same way as before.



Question 1
Correct
Marked out of 1.00

Based on option description, finish the test.

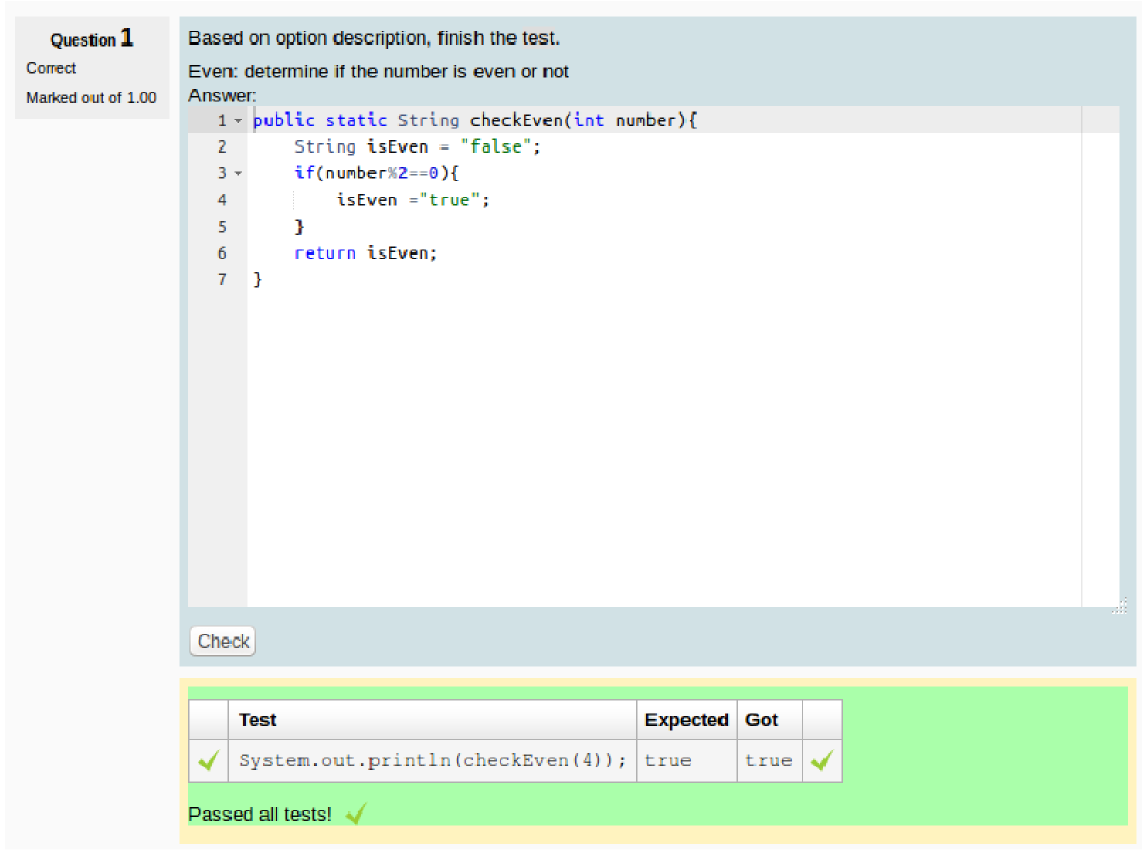
Odd: determine if the number is odd or not
Answer:

```
public static String checkOdd(int number){
    String isOdd = "false";
    if(number%2==1){
        isOdd = "true";
    }
    return isOdd;
}
```

Check

	Test	Expected	Got	
✓	System.out.println(checkOdd(3));	true	true	✓

Passed all tests! ✓



Question 1
Correct
Marked out of 1.00

Based on option description, finish the test.

Even: determine if the number is even or not
Answer:

```
public static String checkEven(int number){
    String isEven = "false";
    if(number%2==0){
        isEven ="true";
    }
    return isEven;
}
```

Check

	Test	Expected	Got	
✓	System.out.println(checkEven(4));	true	true	✓

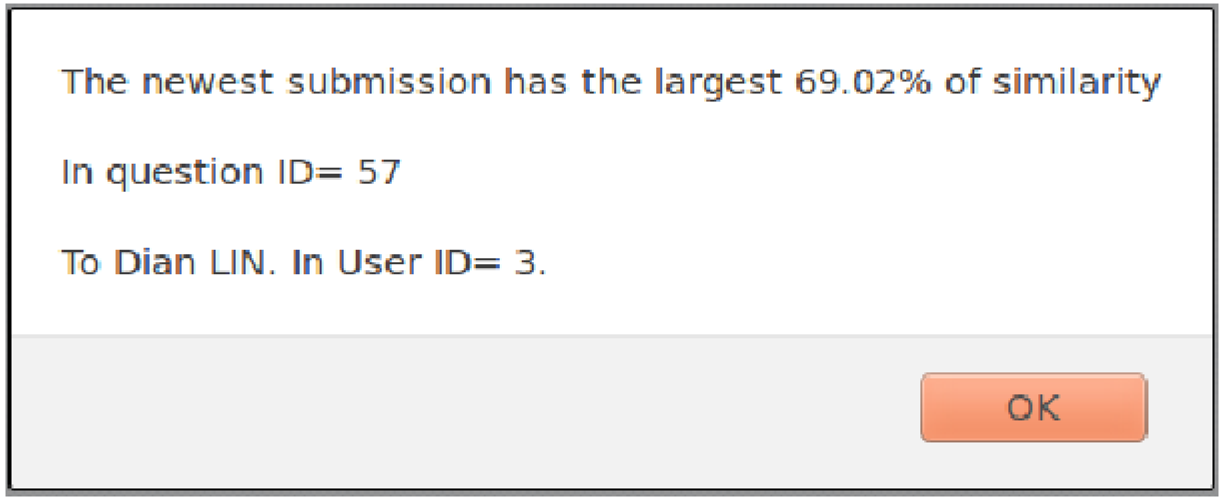
Passed all tests! ✓

Code Runner Implementation 2 (Similarity Checking with Comments)

- **Idea**: similarity checking with comments required
- **Description**: Code Runner requires students to add comments for their program.
- **Advantage**:
  1. Comments are personal thoughts so that they are unlike to be same
  2. Adding more distinction between submissions.
  3. Help students to have a better learning outcomes. (deeper understanding on codes)
- Expected outputs: (only highest similarity will be tracked)
  1. Comparison between the newest submission and submitted ones.
  2. Summary similarity table of every submission after the assignment is due.

Similarity Checking with Comments
(Continued)

- Tracking the details of the student who has the highest similarity percentage with current submission. May be used in further analysis for lecturers or tutors. One example below:



The newest submission has the largest 69.02% of similarity

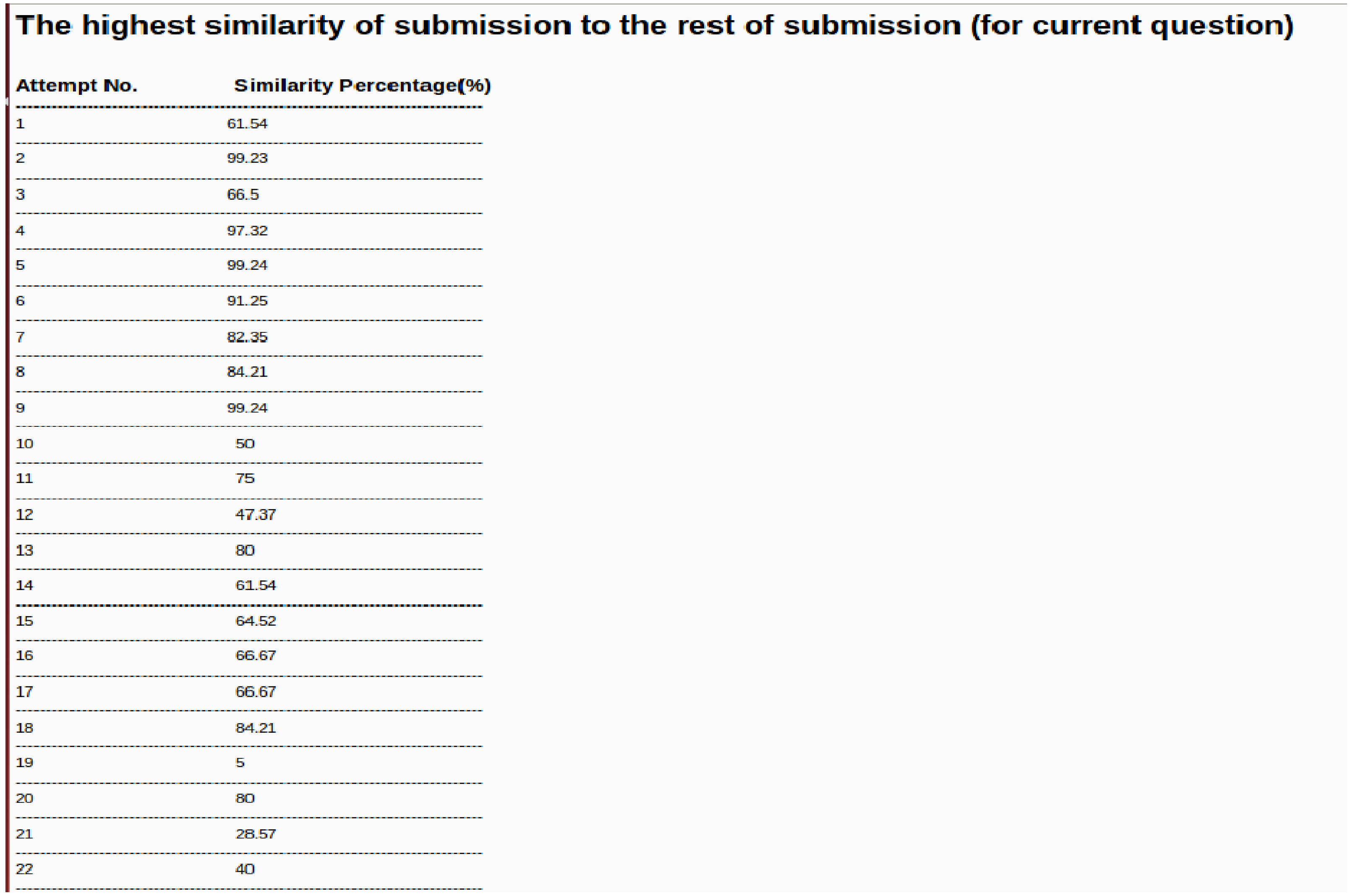
In question ID= 57

To Dian LIN. In User ID= 3.

Similarity Checking with Comments
(Continued)

- After the assignment is inactive, summary similarity table will be generated. Only highest similarity percentage would be recorded.

- Where program with comments still have high similarity percentage should be noticed.



The highest similarity of submission to the rest of submission (for current question)

Attempt No.	Similarity Percentage(%)
1	61.54
2	99.23
3	66.5
4	97.32
5	99.24
6	91.25
7	82.35
8	84.21
9	99.24
10	50
11	75
12	47.37
13	80
14	61.54
15	64.52
16	66.67
17	66.67
18	84.21
19	5
20	80
21	28.57
22	40

Future work

- Significant downsides:
  1. Nonsense comments could possibly be added, similarity will be reduced while it is not allowed.
  2. Heavy manual work-loads for options creating.
- Solutions:
  1. Requires Artificial Intelligent knowledge. AI plugin, setting training set within the plugin, where training set contains sample comments, should be large enough including different comment styles and texts. Matching students' comments with training set.
  2. Introduce the idea from Problet* where one general case created, distinct options will be auto-generated and used for long time.
- Research required.

***** Refer to "Automated Generation of Self-Explanation Questions in Worked Examples in a Model-Based Tutor", Amruth N. Kumar.

Conclusion

- There is no guarantee to say that no one is able to cheat in Code Runner at the end of the Project.
- New anti-cheat system will effectively reduce the probability of cheating by letting students feel difficult to get unfair pass.
- The system helps students to have a better understanding on course learning outcomes
- Code Runner becomes relatively fair.
- Future works required to make the system better.

Thank you!

¿Questions?