Design and Implementation of Fill-in-the-blank Questions based on Open Source Online Judge System

Zhiwei Jiang a, Xiaohua Xu b, *

School of Computer and Information Engineering, Tianjin Agricultural University, Tianjin, 300384, China

a465239880 @qq.com, b, * xuxiaohua@tjau.edu.cn

Abstract. This report discusses the shortcomings of the existing online judge system's problem type, and describes the implementation and applicability of the fill-in-the-blank questions based on the open source online judge system. It is based on the open source online judge system named HUSTOJ, to realize the online judgment function of code-fill-in-the-blank and result-fill-in-the-blank questions by analyzing its system architecture, judgment process and redesigning its database.

Keywords: Online judge, fill-in-the-blank questions, HUSTOJ, programming practice.

1. Introduction

With the advent of the artificial intelligence era, programming skills have become a research hotspot of education at home and abroad [1]. Recently, in China, the Ministry of Education's newly released "Key Points of Educational Informationization and Cybersecurity in 2019" pointed out that education informationization should become the supporter and leader of education modernization, and it needs to promote the implementation of programming education in primary and secondary schools [2]. However, through years of teaching practice experience, programming education can not effectively reflect the students' programming level through the examination of the face-to-face test. For this reason, the online judge system has been developed to assist teachers in programming teaching [3].

Online judge system originated from ACM International Collegiate Programming Contest (ACM-ICPC). The user can submit the source code of various programming languages (such as C, C++) online, compile and execute it, and verify the correctness of the program source code through pre-designed test data [4]. Today, quite a few universities in China have developed appropriate online judge systems, such as POJ, HUSTOJ, ZOJ, XUJCOJ, Lanqiao Cup practice system and more.

The existing online judge system has been perfected to meet the needs of programming teaching in domestic universities. Nevertheless, if these systems are deployed to primary and secondary schools and high school, there will be two obvious shortcomings:

- The single type of question is easy for students to lose interest in doing the problem.
- Programming problems are so difficult that they are not suitable for beginners to practice.

For the purpose of reducing the difficulty of learning programming and enhancing the interest of beginners in programming learning, this report will add code-fill-in-the-blank and result-fill-in-the-blank questions based on the open source system called HUSTOJ. HUSTOJ [5-6] is the online judge system of Huazhong University of Science and Technology (HUST). It was open source under the GPL protocol at the end of 2008. The system architecture consists of two parts: web and core. This report mainly focuses on the modification and extension of the web part.

2. HUSTOJ System

The HUSTOJ system is deployed with using the LNMP (Linux + Nginx + MySQL + PHP) architecture. The structure of the system consists of a front end and a core judge end, and the two ends share a database to realize data communication between them. The answer submitted by the user will be stored in the database through the web side, and afterward the core will read the data from the database to judge the problem and write the result to the database. This is beneficial to achieve centralized control of data and to ensure data consistency and maintainability.
3. Design of Fill-in-the-blank Questions

The result-fill-in-the-blank questions are dissimilar from the code-fill-in-the-blank questions. The answer to the result-fill-in-the-blank question is a string, which requires the user to submit a string that cannot contain any other characters including spaces and newlines. It does not require being compiled and it can be directly judged at the web end. It can effectively save system resources and improve the efficiency of judgment. The answer to the code-fill-in-the-blank question is a code. And the system fills it into the missing piece of pre-prepared compilable code and writes it to the database. Therefore, the code-fill-in-the-blank problem needs to be compiled, and the judgment is the same as the programming problem. Both of them are read by the core to evaluate the database.
4. Design of the Database

There are more than 20 tables in the database of HUSTOJ system. The tables used at the core end are the solution, the problem, the compileinfo, the sim, the users, the custominput, the runtimeinfo, and the source_code. The solution table holds the answers submitted by the user, and the problem table holds the details of the problem, and the source_code table holds the codes submitted by the user.

Since the table for recording the fill-in-the-blank information cannot affect the original database, in order to effectively manage the fill-in-the-blank problem, the problem_fill and solution_fill tables are added to the original database.

The problem_fill table records the mark of the fill-in-the-blank question, the correct answer to the result-fill-in-the-blank question, the missing code for the code-fill-in-the-blank question, and the missing mark.

<table>
<thead>
<tr>
<th>#</th>
<th>Name</th>
<th>Type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>problem_id</td>
<td>int(11)</td>
<td>Question id, primary key</td>
</tr>
<tr>
<td>2</td>
<td>problem_flag</td>
<td>char(1)</td>
<td>Question mark (0/1)</td>
</tr>
<tr>
<td>3</td>
<td>problem_tempcode</td>
<td>text</td>
<td>Missing code for problem description</td>
</tr>
<tr>
<td>4</td>
<td>language</td>
<td>int unsigned</td>
<td>Code language</td>
</tr>
<tr>
<td>5</td>
<td>problem_tempsource</td>
<td>text</td>
<td>Missing code for compiling</td>
</tr>
<tr>
<td>6</td>
<td>fillmd5</td>
<td>varchar(32)</td>
<td>Md5 for missing mark</td>
</tr>
<tr>
<td>7</td>
<td>problem_answer</td>
<td>varchar(256)</td>
<td>Answer to the result-fill-in-the-blank</td>
</tr>
</tbody>
</table>

The solution fill table records the answers submitted by the user.

<table>
<thead>
<tr>
<th>#</th>
<th>Column name</th>
<th>type of data</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>solution_id</td>
<td>int(11)</td>
<td>Submit id, primary key</td>
</tr>
<tr>
<td>2</td>
<td>problem_flag</td>
<td>char(1)</td>
<td>Question mark (0/1)</td>
</tr>
<tr>
<td>3</td>
<td>solution_answer</td>
<td>varchar(256)</td>
<td>the answers submitted by the user</td>
</tr>
</tbody>
</table>

5. The Code Design of the Judgment

After filling the answer of the missing code, the code-fill-in-the-blank passes it to the core end for judgment. When there is more than one code filled in, the system uses md5 to distinguish each answer of the fill-in-the-blank question. The key code is as follows:

```php
$problem_tempsource = $row_fill['problem_tempsource'];
for($i=0;$i<substr_count($row_fill['problem_tempsource'],$row_fill['fillmd5']);$i++){
    $pos=strpos($problem_tempsource, $row_fill['fillmd5']);
    $problem_tempsource = substr_replace($problem_tempsource, $source[$i], $pos, 32);
}
```

The judgment of the result-fill-in-the-blank is implemented directly on the web side. Compared with the answer submitted by the user and the correct answer, the same is true, otherwise it is wrong. The key code is as follows:

```php
if(count($source) == count($problem_answer)){
    for($i=0;$i<count($source);$i++){
```

68
if(strcmp($source[$i], $problem_answer[$i]))
    $err_answer=$err_answer." The answer is wrong, please check the answer. \n";
else $err_answer=" Wrong answer!";

6. System Testing

The system's judgment is based on the black box test method, and the report will also use the black box test method to test the system. From the test results in Figure 4, it is known that the system after the extended fill-in-the-blank question can achieve the desired effect.

![Fig. 4. The result of the test](image)

7. Summary

This paper deeply studied the working principle of the open source online judge system, and added the function of setting the fill-in-the-blank question. The questioner could not only set the programming problem, but also set the result-fill-in-the-blank and the code-fill-in-the-blank question. The system source code after the extended fill-in-the-blank question was stored in the GitHub code repository [7]. In the following research, we could continue to expand other types of questions such as multiple-choice questions and judgment questions. And you could also add other practical features such as school-level exams and contests.

Acknowledgements

Fund Project: College Students' Innovation and Entrepreneurship Training Program, Tianjin, China (Grant No: 201710061079).

References


[7]. The new system source code in GitHub. https://github.com/NopeALie/OnlineJudge.