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

[a]465239880 @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.

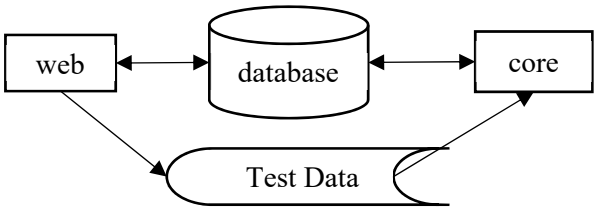

Fig. 1. The architecture of the open source online judge system

The programming problem of the HUSTOJ system is judged at the core end. The core end polls the queue of pending questions from the database. When the length of the queue is greater than 0, the answer code will be extracted for compilation. If the compilation error occurs, an error is returned. After the compilation is successful, the system will use the test data to verify the correctness of the compiled program, and only when all the test data has passed will it return success. When the answer is completely correct, the system will also check the code for plagiarism.

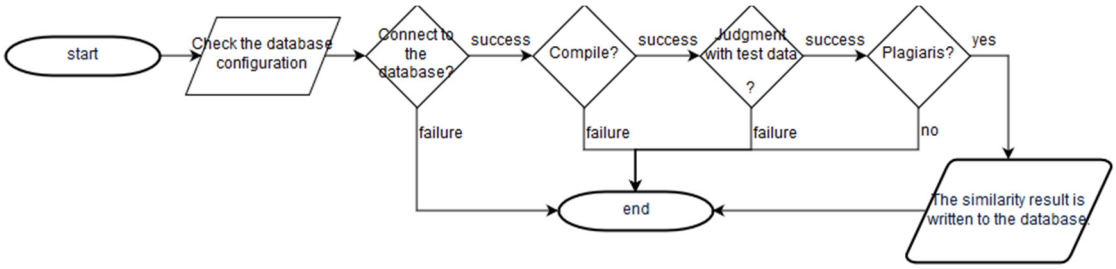

Fig. 2. The core end judgment flow chart

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.

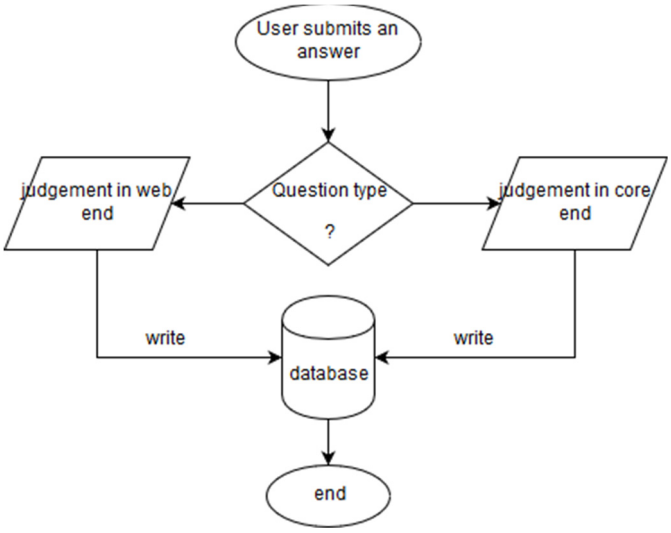

Fig. 3. The judgement flow chart of the code fill-in-the-blank.

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 1. Table problem_fill

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

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

Table 2. Table solution_fill

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

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:

```
$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);
}
$source=$problem_tempsource;
```

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:

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

```
            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.

Problem ID: User: Language: All Result: All Search

RunID	User	Problem	Result	Memory	Time	Language	Code Length	Submit Time	Judger
1061	admin	1000	Accepted	1120	0	[C]	193 B	2019-03-04 20:43:33	admin
1060	admin	1000	Compiling	0	0	[C]	193 B	2019-03-04 20:39:56	LOCAL
1059	admin	1008	Wrong Answer	2020	0	[C++/code-fill]	6 B	2019-02-27 22:31:21	admin
1058	admin	1008	Accepted	2020	0	[C++/code-fill]	6 B	2019-02-27 22:31:10	admin
1057	admin	1008	Compile Error	0	0	[C/code-fill]	6 B	2019-02-27 22:30:18	admin
1056	admin	1003	Accepted	0	0	[result-fill]	11 B	2019-02-27 16:45:24	LOCAL
1055	admin	1003	Wrong Answer	0	0	[result-fill]	19 B	2019-02-27 16:43:41	LOCAL

Fig. 4. The result of the test

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

[1]. Wei Xiaofeng, Jiang Jiafu, Zhong Hong, Han Qi. Reflections on the development of programming education in primary and secondary schools in China [j]. China Education Informationization, 2018 (24): 1-4+9.

[2]. Key Points of Educational Informationization and Cybersecurity in 2019. http: // www. moe. gov. cn/srcsite/A16/s3342/201903/t20190312_373147.html.

[3]. Sun Yuxia, Xia Duanfeng, Yang Wei, Tong Qiang, Xu Xinshan. Discussion on Programming Practice Teaching Based on Online Evaluation System [j]. Computer Programming Skills and Maintenance, 2016(22): 29-30.

[4]. Online judge system in Baidu Encyclopedia. https://baike.baidu.com/item/OJ/8129019.

[5]. Zhang Haobin. Design and implementation of open source online evaluation system based on open cloud platform [j]. Computer Science, 2012, 39(s3): 339-343+348.

[6]. The open source code of HUSTOJ. https://github.com/zhblue/hustoj.

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