Development of Homework Evaluation Assistant System for PHP Programming Teaching

Buye Lou

School of Information, Capital University of Economics and Business, Beijing 100070, China

loubuye@163.com

Keywords: PHP Teaching; Homework Evaluation; Current User; Homework Set; Web Page Stub

Abstract. This paper introduces the development of the homework evaluation assistant system for PHP programming teaching. The role of the system is to facilitate teachers to evaluate the homework (i.e. some web pages or websites) done by students. The system consists of supporting subsystem, homework stub subsystem and homework set. With the support subsystem, students can upload finished homework to the homework set and the teacher can switch current user. With the homework stub system, students can access their own homework stored in the homework set with the same URLs through the browser, the teacher can access all the students' homework with the same URLs through the browser.

Introduction

In PHP programming teaching, it is necessary to arrange some homework (i.e. some web pages or websites) for students to practice, and to evaluate the students' homework is an essential part [1]. For PHP programming homework, the general evaluation process is as follows:

(1) Collect homework submitted by students through E-mail and other means of communication;

(2) Copy or deploy a student's homework to a web server;

(3) Access and browse the homework (web pages or websites) through the browser, and then evaluate the student's homework;

(4) Remove or unload the student's homework from the web server.

This process needs to be applied repeatedly to each student's homework. Although the process and operation of evaluating each student's homework are almost the same (assuming that the homework has specified the name or URL of each web page), the workload of evaluating is quite heavy for a teacher due to the large number of students. In addition, there are the following problems in the actual work:

(1) It is not convenient to re-evaluate homework. Second or more evaluation of questionable homework is unavoidable, in which case it may be necessary to copy or deploy the homework to the web server multiple times.

(2) It is not convenient to evaluate homework in stages. Some large homework requires students to complete in stages, and it also requires the teacher to evaluate homework in stages. This will multiply the workload of teachers' evaluation.

(3) Too many uncertainties affect the correctness of the evaluation. These uncertainties include: the small difference between the operating environment when students do homework and the operating environment when teachers evaluate homework; disoperation occurs in the process of students submitting assignments, teachers collecting and deploying homework, etc.

This paper introduces the development of the homework evaluation assistant system. The system can overcome the above problems and assist teachers to evaluate students' homework.

Structure and Function of the System

System Objectives. The objectives of this homework evaluation assist system include:

(1) Students can upload the finished homework (complete or partial) to the web server at any time according to the teaching schedule, and can access their homework (web pages or websites) and confirm the running results of the homework through the browser.

(2) The URLs of each student visiting his homework is the same, that is, the URLs specified in homework.

(3) The teacher can access any student's homework at the same URLs: first select the student, then access through the browser, and the system will automatically connect to the selected student's homework.

Structure and Function. This homework evaluation assistant system is composed of support subsystem, homework stub subsystem and homework set, as shown in Fig.1.

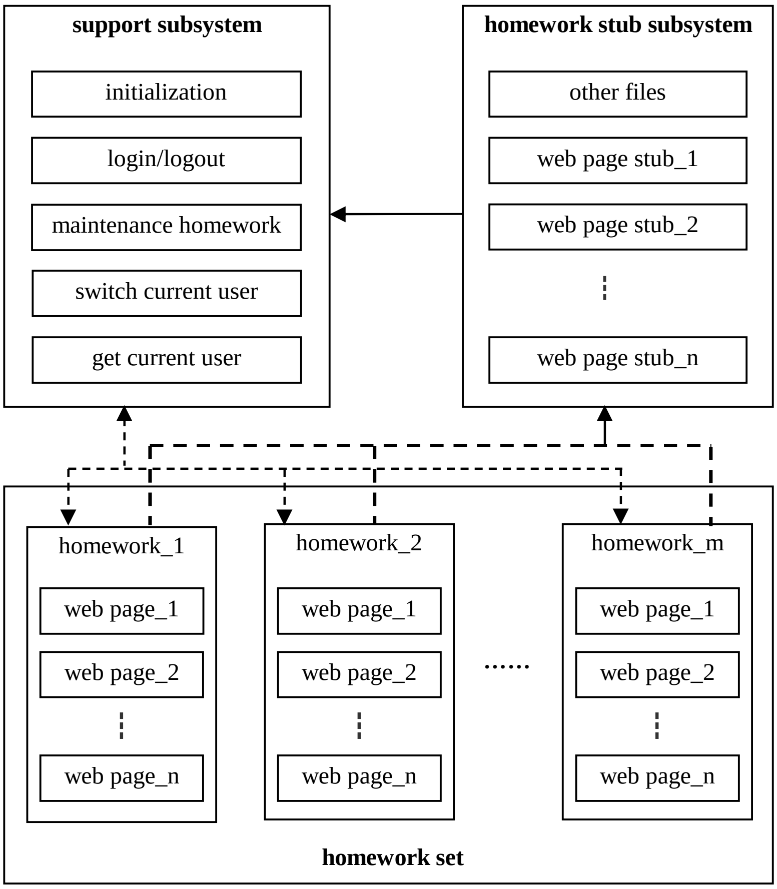

Fig.1 Structure and function of the system

The main function of the support subsystem is to maintain the homework set and set the current user. With the support subsystem, students can upload their homework to the homework set. The current user is associated with the session [2], and there can be only one current user at any time during a session.

The homework stub subsystem is based on the homework reference answers. It rewrites all the web pages that students are required to write into web page stubs.

Regardless of which student's homework is accessed, it is done by accessing the homework stub subsystem. The URL of each web page stub in the homework stub subsystem is the URL of the corresponding web page specified by the homework, so visiting any student's homework uses the same URLs. Each web page stub automatically connects to the corresponding web page file in the current user's homework, which is transparent to the user.

The homework set consists of a number of independent student homework. Each homework assignment contains web pages that are generally not accessible directly from the client browser.

Support subsystem, homework stub subsystem and homework set are related as follows:

(1) The homework set is maintained by the support subsystem, and the homework stub subsystem accesses the homework set.

(2) The current user is set in the support subsystem, and the homework stub subsystem needs to get the current user's information to connect to the corresponding homework.

The support subsystem includes function modules such as initialization, login and logout, maintenance homework, switching current user, and getting current user.

(1) Initialization. This function is restricted to teachers only, including: initializing student identity information, including username, password, etc.; prepare homework space for each student.

(2) Login and logout. System users include students and teachers. At the beginning, the login user is the current user.

(3) Maintenance homework. Maintain the current user's homework files in the Homework Set, including uploading, downloading, or deleting the homework files.

(4) Switch current user. This function is restricted to teachers only, and the login teacher can select any registered user as the current user. For login students, the current user can only be themselves.

(5) Get current user. This module returns the username of the current user for the specified session. The current user is associated with the session, that is, there can only be one current user at any time in a session.

Design and Implementation of the System

The design and implementation of main functions are introduced here.

The user identity information of the support subsystem is stored in the database table *user* with the following relational schema [3]:

user (username, password, realname, email, isTeacher)

The identity information of the teachers can be entered manually in advance.

The homework set is stored in the file system and corresponds to a folder called the homework set root folder. The folder's address needs to be added to the include_path item in the initialization file php.int so that each web page stub in the homework stub subsystem can include and process corresponding web page file in the student's homework [4].

The input to the initialization module of the supporting subsystem is an Excel sheet containing student identity information. This module reads students' information and adds corresponding records in the *user* table, and generates corresponding subfolders in the root folder of homework set. Each student's homework corresponds to a subfolder whose name is the student's username and which stores the web page file of the student's homework.

Make sure that the support subsystem is relatively independent from the homework stub subsystem, that is, the support subsystem should not interfere with the running of the homework, and, in turn, the running of the homework should not affect the state of the support subsystem. The state of the support subsystem is primarily the current user associated with the session. Because the function of homework and its implementation technology cannot be restricted, so only the implementation technology of support subsystem can be constrained. Here, instead of using session variables to store relevant information of the current user, relevant information is stored in a database table *inuse*, with the following relational mode:

inuse (sessionid, user, teacher)

Where *sessionid* is the primary key. When a user logs in successfully during a new session, a record is added to the table, which is the current user. Where the *sessionid* column holds the session id of the current session and the *user* column holds the username of the current user. If the login user is a student, the *teacher* column is NULL. If the login user is a teacher, the *teacher* column also saves the teacher's username. When the login user is a teacher, he can change the current user by calling the switch current user module. That is, when *teacher* column is non NULL, the value of the *user* column can be changed.

When the user logs out, the corresponding record will be deleted from the *inuse* table. If there is no normal logout, the corresponding record should be deleted when the user logs in successfully the next session.

The input of get current user module is the session ID of the current session. The module returns the username of the current user by retrieving the *inuse* table.

The implementation of any web page stub is similar: first invoke the get current user module in the support subsystem to get username of the current user, and then set the address of the corresponding web page file in the current user's homework, and finally contain and process the code of the web page file, as shown in Fig.2.

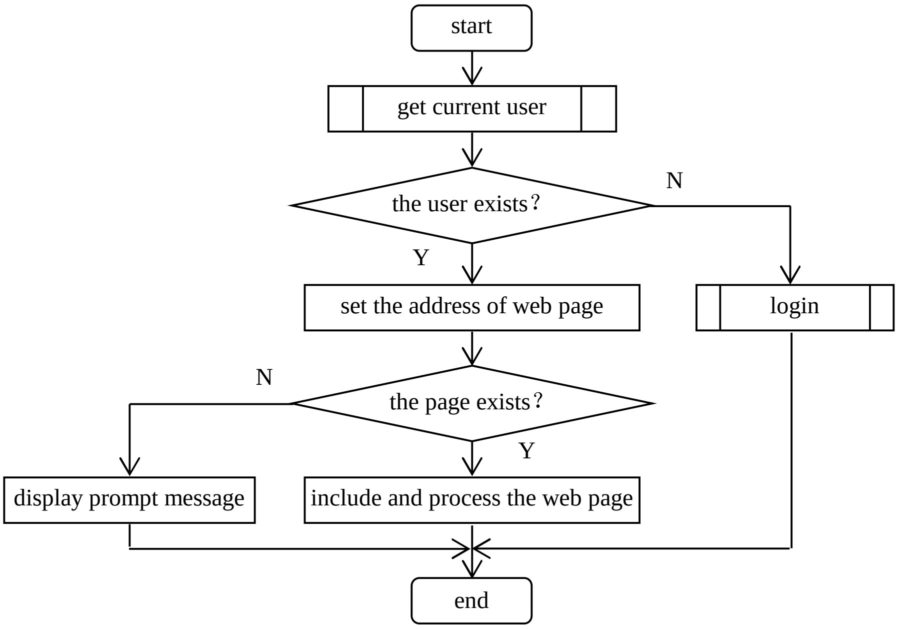

Fig.2 web page stub flow chart

A user should first run the support subsystem and set the current user, and then run the homework stub subsystem in the same session to access and check the current user's corresponding homework.

Conclusions

This system can overcome many problems existing in the general evaluation method, reduce the workload, and improve the reliability and convenience.

(1) For students, they can upload the finished homework (complete or part) to the web server at any time, and confirm its running effect. If they are not satisfied, they can modify it and upload it again.

(2) For teachers, there is no need to collect and deploy homework, and any student's homework can be accessed and evaluated directly and at any time.

References

[1] Lou Buye, Zhang Jun. PHP Web Application Development (Tsinghua university press, Beijing, 2017).

[2] Vijay Joshi. PHP jQuery Cookbook (Packt Publishing, Birmingham, 2010).

[3] Daniel Bartholomew. Getting Started with MariaDB Second Edition (Packt Publishing, Birmingham, 2015).

[4] W.J.Gilmore. Beginning PHP and MySQL: From Novice to Professional, Fourth Edition (Apress, Berkeley, 2010).