

Automated Programming Assignment Grading System

S. K. T. D. Subashith

Department of Computer Science and Engineering,
University of Moratuwa, Sri Lanka
tdiluksha.13@gmail.com

Abstract—Today there are many e-learning systems that fulfill the thirst for education. Those e-learning systems have many courses in different fields. There are so many e-classes that teach how to program in many different programming languages. But in most of those e-classes instructors have to spend a lot of time to evaluate assignments submitted by students. If the grading process can be done automatically, it will save lots of time of instructors. There are many systems that use the functionality of evaluating programs automatically. They provide support for hosting online programming competitions, but most of them do not provide platforms for e-classes. Therefore Automated Programming Assignment Grading System (APAGS) is the solution which supports both e-class environment and programming competition environment. The APAGS is a web system which has developed using Symfony framework according to the model, control and view architecture to provide both e-class environment and competition hosting environment. Its main goal is to evaluate the programs and analyse results for both assignments and competitions.

Keywords— *e-classes; Programming Competition; Assignment; Grading; Model View Controller; PHP; Symfony*

I. INTRODUCTION

With the development of the technology and internet, the way of learning has changed. These days, there are many online classes that can be followed for learning. There are many courses on programming through e-learning systems. In those courses there are assignments that have to be submitted by the students and has to be graded by the instructors. If those submissions can be graded automatically a lot of time of instructors can be saved and there will be more time for instructors to focus on their teaching materials.

Furthermore it is better if those automatically grading systems can be used to have programming competitions among students of a class to improve their skills. There are many online programming competition platforms which have their own way of evaluating the programs written by contestant. But many of them does not provide the environment to have an online class environment. HackerRank [1] is one of the famous systems for online programming competitions. But it does not support for the functionalities of the e-learning system that has activities of submitting assignments and grading them. Therefore the Automated Programming Assignments Grading System

(APAGS) is the solution for both e-learning and competition support system.

The system allows the instructors to focus more on their teaching by saving their time of grading submissions. It also improves the students' performance by providing more competitive environment by hosting competitions. Those competitions can be created by the instructors and students can register to them. Even the people who do not have accounts in APAGS system can register to competitions by creating lower privileged student accounts. Moreover the system will help both instructors and students by analysing their performance as a class and as individuals. Automated Programming Assignment Grading System is mainly focused on automating the procedure of grading programming assignments and analysing the results. This system is very useful for the resource persons in online courses to handle their classes effectively.

The paper is structured as follows. Section II discusses the literature review relevant to this system and some other competition hosting platforms. Section III contains the details about the system model. It includes system requirements, system design and database design. Section IV describes the implementation procedure of the system and main interfaces of the system. In the section V the testing analysis of the system has been described. Conclusion and future works are described in the last section.

II. LITERATURE REVIEW

The development of the Automated Programming Assignments Grading System considers several theoretical aspects. APAGS currently based on Python language. Unlike compiled programming languages such as C, C++ or C#, Python uses an interpreter. The Python programs are first compiled automatically into a byte code used by Python virtual machine and then run on it [2]. Therefore APAGS has more concerned about this since it will be improved to have other languages C and JAVA.

There are systems that have the functionality of evaluating programs by running it automatically. Many of them are used for online programming competitions that have large number of contestants. Following is a list of them with an analysis of how they work.

HackerRank [1] is a web application for solving programming problems. It mainly supports for the online

programming competitions. It can handle large number of contestants at one time. It runs on a Ubuntu 14.04(LTS) AMD64 virtualized EC2 instance. It has a submission size limit of 64kB. All the submissions are evaluated comparing the test case inputs and outputs to programs outputs. Therefore test cases should be provided when creating the problem. This system supports for more than 30 programming languages but it does not have the features for e-classes.

Mooshak [3] and Stepic [4] are other systems that has the feature of grading programs automatically. Mooshak also supports hosting online programming competitions. Web-CAT [5] is a web application that gives the support for instructors to evaluate their students' programs. This system is not available for organizing competitions.

All the above mentioned systems have the ability of evaluating a programming assignment automatically and analysing the results. The APAGS is also capable of doing that but has additional features too. Mainly the APAGS system provides the platform for e-classes by giving the features to create classes and add students to them.

Coursera [6] is another e-learning system which has many e-courses. But it grades the assignment by peer reviewing. It does not support competition hosting. Code school [7] is another e-learning system which automatically grade the assignments. But it does not support creating classes and hosting competitions.

APAGS also supports for programming competitions among the class members. It can host competitions for outsiders. and provide an analysis of results of both class and individuals for assignments and competitions. Therefore APAGS is different from others by supporting both e-class and competitions platforms.

III. SYSTEM MODELS

A. System Requirement

This system's main functional requirements mainly depend on three actors. They are instructors, students and system admin. Instructor's main activities are creating classes, adding students to classes and creating assignments. Students are involved with submitting assignments, discussing problems and viewing their progress. The system admin is responsible for creating accounts, especially in e-class environments. The system is responsible for evaluating assignments and keeping track of progress of both classes and individuals. The system is well designed to support all these functional requirements.

The non-functional requirements of the system are concerned with the security aspects and performance aspects of the system. Security of the system should be very high since the main functionality (code evaluation) use many system resources. There is a risk of attacking the system through malicious code submissions. Therefore security is vital factor for the system. Performance is important since the system should be capable of handling a large number of

concurrent users without affecting the performance of the system.

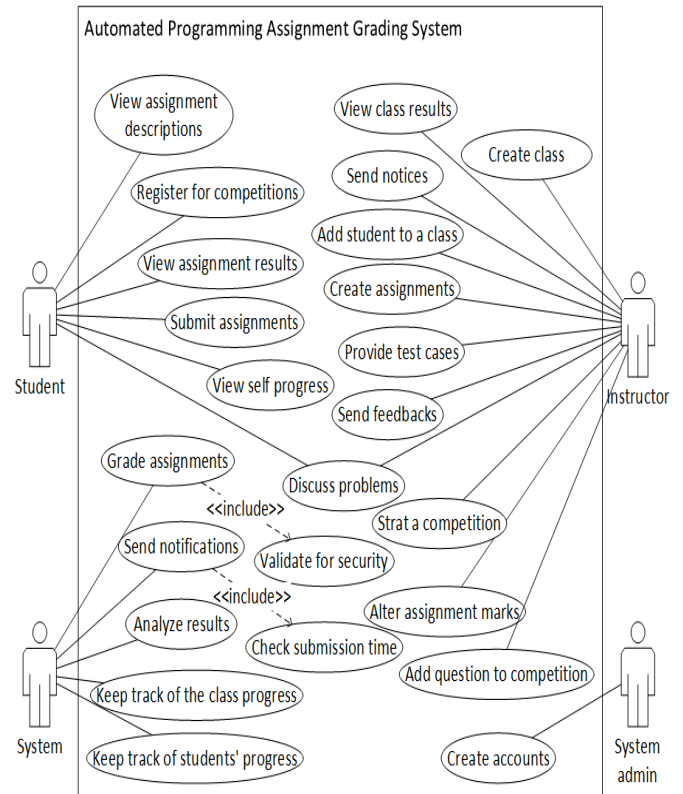


Fig. 1 Main use case diagram

Figure 1 shows the main use cases of the system. The main four actors are instructor, student, system admin and the system. They all are participants of main functionalities of the system.

B. System Design

The APAGS system follows the Model View Controller (MVC) architecture. In this design architecture the roles who interacts with the system are represented by the models. The user interfaces are represented by the views and the data processing and functional parts of the system represented the controller. This architecture ensures the maintainability of the system.

Figure 2 shows the logical view of the APAGS system. At the top of the diagram classes like Instructor, Student, System admin represent the actual entities and the classes such as controller represent the more functional classes in MVC architecture. File handle and DB connection classes are responsible for the handling the file system (created python file) in system and handling the data base connections.

But when it comes to the actual implementation there are slight changes because of the behaviour of Symfony framework. Most of the entity classes does not have any functions and are used only to hold attributes and the functions are implemented in respective controller classes.

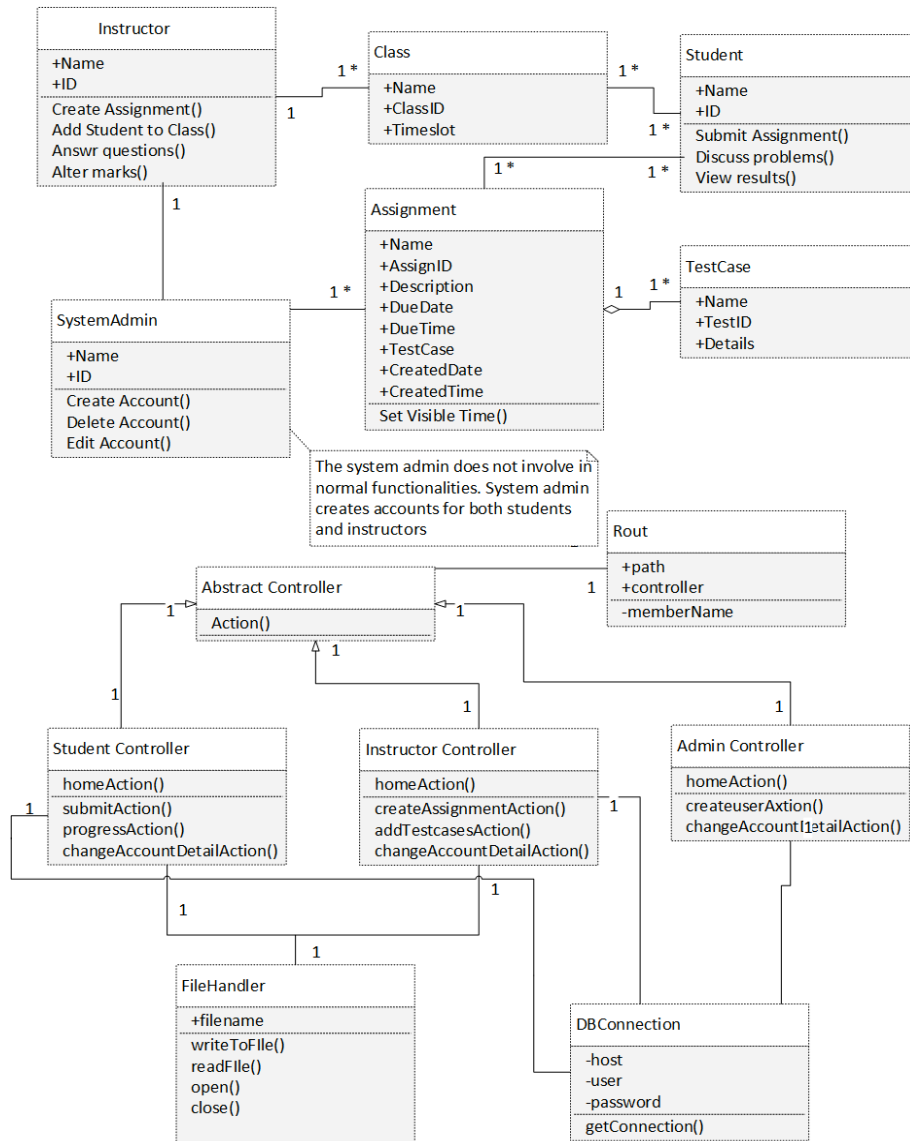


Fig. 2 Class diagram

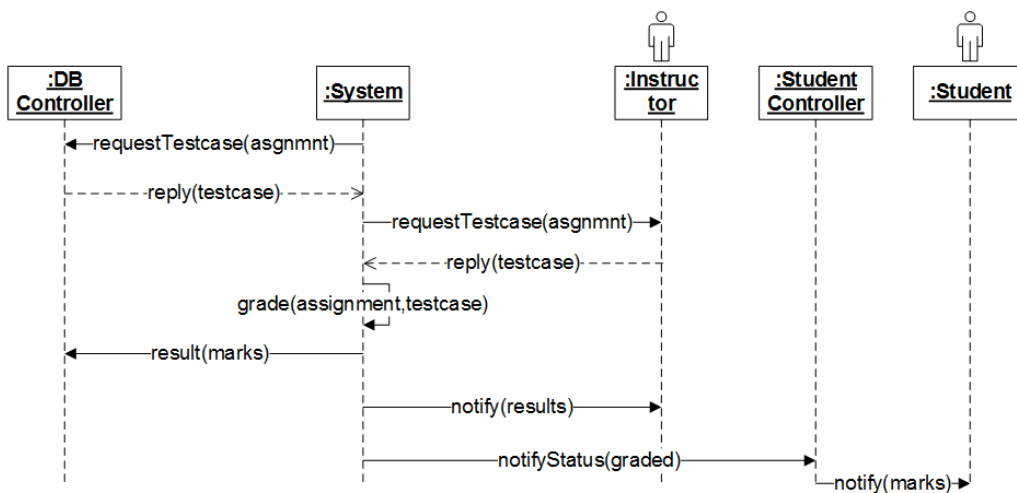


Fig. 3 Sequence diagram

Figure 3 shows the sequence diagram of how an assignment is graded when it pass the due time. First it will check for the test cases provide by the instructor in the data base. If there are no test cases provided, the system will request test cases from instructors. When the system gets the test cases it will grade assignments according to those test cases and save results to database. Then the instructors are notified with results and finally results will be sent to students through the student controller.

C. Database Design

Database consists of main entities of the system. There are tables for student, Instructor and system admin. The database contains table for classes created by instructor with the details of created time, name and class id. Also it has tables to store created assignments and submitted assignments with their results. Moreover it has test case table to store test outputs and results with respective assignment id

IV. SYSTEM IMPLEMENTATION

A. Implementation Procedure

The APAGS system was developed using the Symfony framework, which is a PHP framework that supports model, view, controller architecture. Therefore the source code can be easily managed according to the models in the system.

The system used PhpStorm [8] as the development environment and MySQL for the database. Also “phpmyadmin” used for the interactions with the database since it provides a graphical user interface with database.

As the software development methodology the “Rational Unified Process” was preferred since it provides a method to keep track of the development progress. According to the rational Unified Process there are four phases identified as below.

- Inception phase
- Elaboration phase
- Construction phase
- Transition phase

During each of these phases, considerable amount of work is completed according to the phase activities. During the inception phase the business case of the system was developed according to the domain problem. Also the requirement analysis was done during this phase of the project and the schedule of the project was created.

During the Elaboration phase the applicable risk for the project and methods of mitigating those risks were identified. At the beginning of the construction phase, actual development of the system started. Throughout the construction phase the development happens parallel with testing. The transition phase is the time that deploys the system and change the system into production environment.

```

Check assignment due date and time
If due time has passed {
    Deactivate submission link;
    If submitted {
        Run the assignment;
        If no syntax errors {
            Get the outputs;
            Compare outputs with test case
            results;
            Allocate marks for test cases
            and calculate total mark;
            Save data to database;
        }
        Else syntax errors {
            Save error to database with zero
            marks;
        }
    }
    Else not submitted {
        Save not submitted with zero marks
        to database;
    }
    Notify the instructor;
}

```

Fig. 4 Algorithm for assignment grading

Figure 4 shows the pseudocode of how an assignment is graded automatically. First the system checks the due time of the assignment and when it pass the due time it will start evaluation. Then it will check whether the students have submitted or not. If not submitted, it will be recorded in the database. If the assignments were submitted then the system will evaluate them by running them.

When the system tries to run the program it will be checked for syntax errors by a syntax validator tool. The tool will go through the code and identify the syntax errors. If there are syntax errors the errors will be saved to database. If there are no errors the program will be run and evaluated.

If the program is executed successfully without any runtime errors, then the outputs of the program will be checked against the test case results provided by instructors, marks will be allocated for each test case and total marks will be calculated. If the program does not pass some test cases it will not get the marks for those test cases. When calculating the total marks the weight of the test case is also concerned. After everything is finished, the instructor will be notified with student’s marks.

B. Main Interfaces

Figure 5 is the login interface of the system. It will identify whether the login attempt was made by an instructor a student or a system admin according to the user id provided. There is an option for creating users. It will only create accounts for student role which has lowest privileges, for the purpose of participating in programming competitions only. If an instructor wants to create an account he should contact system administrator.

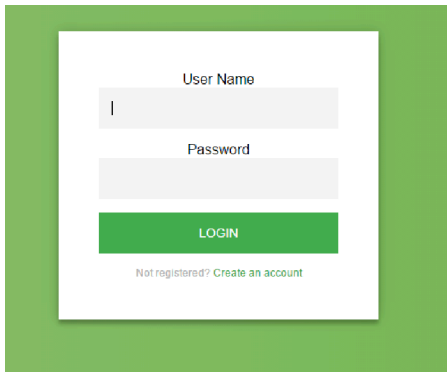


Fig. 5 Login interface

Figure 6 is the interface that shows all the classes for the instructor. There the instructor can create a new class and add students to a created class. Furthermore the instructor can create new assignments and view created assignments.

V. SYSTEM TESTING ANALYSIS

For testing approach the database is tested with much data and the functionality of retrieving data to the system with manually checking using the generated data. Furthermore the system is checked by altering some data and providing invalid data types. Some test cases failed due to constraints used in the database and then those were changed and identified the normalized database for the system.

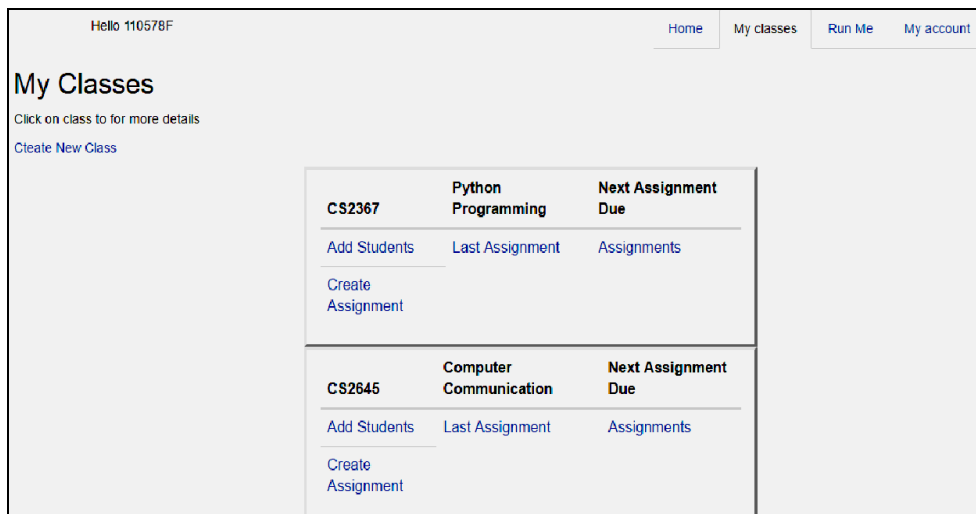


Fig. 6 Class interface

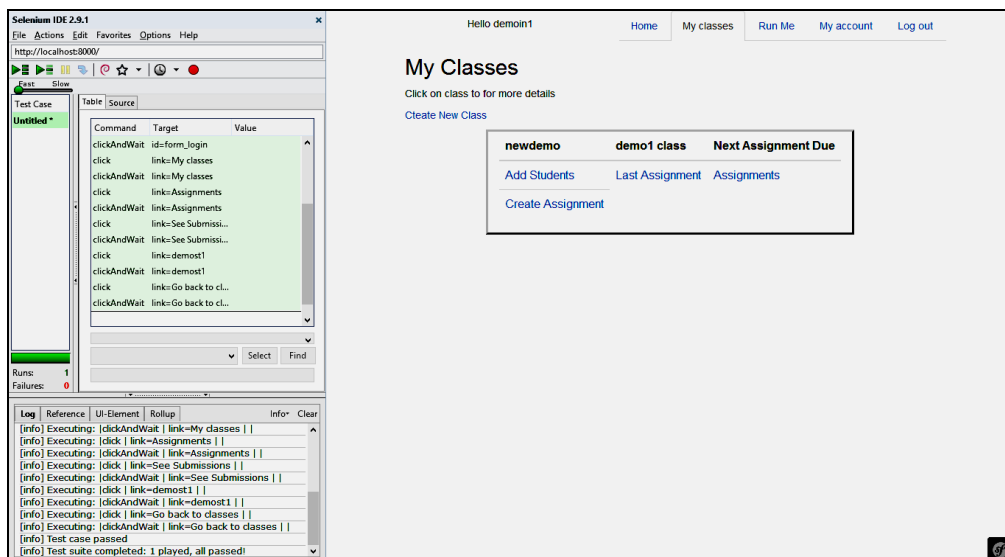


Fig. 7 Selenium IDE test results

The user interface components were tested using Selenium IDE [9]. With every change to the code, the test was run to check whether there are any problems with user interaction components. There were some situations where the test cases failed and it was easy to identify them with help of selenium IDE. Figure 7 shows the user interface test results which were done with Selenium IDE.

VI. CONCLUSION AND FUTURE WORK

The Automated Programming Assignment Grading System is mainly focused on automating the procedure of grading programming assignments and analyzing the results. The system is very useful for the resource persons in online courses to handle their classes effectively. It also provides the facility to have programming competitions in the e-learning environment to improve the quality of the e-learning experience.

This system will be a help both young teachers and students. With globalization and introduction of new technologies, everything has started to become online. Learning programming too has become easy with the technical developments. It is clear that it will be easier if this system is used for e-learning activities. Institutes can earn more profit and students can have superior learning experience with the use of APAGS system.

Currently the system is functioning properly for python programs, but it should be extended for other programming languages as well to get the full benefit of the system. Therefore the main future work is to develop the system for other languages.

Furthermore the system should be improved with a code analysing algorithm that check for malicious codes prior to execution of code. This code analysing algorithm should be able to read the code and identify what the code is doing. So this algorithm is the solution for identifying the wrong programs that produce the expected test case results mistakenly. Moreover to address the issue of rapid increase in the data volume of the database, system can be modified to use MongoDB. Since the system architecture has been created in a way, which is open to extension, those changes are realistic. Therefore APAGS system's reliability, performance and security can be further improved.

REFERENCES

- [1] "Compiling Python," [Online]. Available: <http://programmers.stackexchange.com/questions/136942/why-doesnt-python-need-a-compiler>. [Accessed 18 06 2016].
- [2] "HackerRank," [Online]. Available: <https://www.hackerrank.com>. [Accessed 07 03 2016].
- [3] "Mooshak," [Online]. Available: <https://mooshak.dcc.fc.up.pt>. [Accessed 07 03 2016].
- [4] "Stepic," [Online]. Available: <https://stepic.org/lesson/Step-Code-9173/step/3?unit=undefined>. [Accessed 07 03 2016].
- [5] "Web-Cat," [Online]. Available: <http://web-cat.org/group/web-ca>. [Accessed 06 03 2016].
- [6] "Coursera," [Online]. Available: <https://www.coursera.org>. [Accessed 8 10 2016].
- [7] "Code School," [Online]. Available: <https://www.codeschool.com>. [Accessed 9 10 2016].
- [8] PhpStorm, [Online] Available: <https://www.jetbrains.com/phpstorm>
- [9] "SeleniumHQ," [Online]. Available: <http://www.seleniumhq.org/>. [Accessed 05 06 2016].