Tests Generation Oriented Web-Based Automatic Assessment of Programming Assignments

Yann Le Ru, Michaël Aron, Jean-Pierre Gerval and Thibault Napoleon

Abstract This paper describes a Web-based system that is aimed to help both academic staff and students in the task of automatic assessment of programming assignments. The server implements LAMP (Linux, Apache, MySQL and PHP) architecture powered by the Laravel Framework and designed by Twitter bootstrap. The system focuses on adaptability, scalability and simplicity. It also enables teachers with few skills in the field of software testing to use such a tool. This system implements various types of technologies: databases, Web technologies, object oriented programming, responsive Web design, design patterns, and others.

Keywords Web-based system · Automatic assessment · Programming · Tests

1 Introduction

In this paper, we propose a Web-Based System that is aimed to help both academic staff and students in the task of automatic assessment of programming assignments.

The benefits of using such a tool are reviewed in [1]. Automatic assessment of programming assignments is of interest for all educational organisations to improve the quality in teaching and involve students in the learning process. Moreover,

Y. Le Ru (✉) · M. Aron · J.-P. Gerval · T. Napoleon
Institut Supérieur de l'Electronique et du Numérique, Brest 20 Rue Cuirassé Bretagne, CS 42807 - 29228 Brest Cedex 2, France
e-mail: yann.le-ru@isen-bretagne.fr

M. Aron
e-mail: michael.aron@isen-bretagne.fr

J.-P. Gerval
e-mail: jean-pierre.gerval@isen-bretagne.fr

T. Napoleon
e-mail: thibault.napoleon@isen-bretagne.fr

V.L. Uskov et al. (eds.), *Smart Education and Smart e-Learning*,
Smart Innovation, Systems and Technologies 41,
DOI 10.1007/978-3-319-19875-0_11

assessing students is more objective as the teacher is not involved in the grading process. This is also faster because a repeating task is more adapted to computers.

The features to be assessed include but are not limited to functionalities, efficiency, testing-skills, robustness, quality by means of code analyses, etc. [2]. Sometime tools also provide anti-plagiarism features [3].

The goal of this paper is not to discuss about the utility but to deal with the tools and especially on the functionalities part of the assessment.

In the next sections, we will provide an overview of the existing tools' functionalities and explain the motivation of developing a new tool. Then, we will present an overview of developed Web-based system and details about our Automatic Assessment of Programming Assignments implementation.

2 An Overview of Tool's Functionalities

2.1 Web-Based Systems

First, most of the existing solutions in automatic assessment of programming assignments are web-based [4, 5]. Indeed, it provides an autonomous way for students to assess their own program, and an easy way for teachers to supervise. Usually the software includes two different interfaces. From student's interface, the user uploads his work, it runs on the server side and provides a feedback. From the teacher's interface, the user access to a dashboard to supervise all the works.

2.2 Programming Languages

Tools exists for many languages such as C, C++, Python, Pascal, shell, Assembler, etc. and Java is the most represented [2]. Some tools are specialized for one of programming languages; others implement few of them by the mean of plug-ins.

2.3 Tests Implementation

To assess the functionality part of a programming work, a majority of the systems use xUnit tests frameworks. Tools such as Web-Cat [6] focus on assessing the student's performance at testing his or her own code by writing their own xUnit tests. The idea is to make the student aware of the importance of testing. Others grade only the skills of programming, the teacher write the xUnit test for the specific exercise. In both cases, the tool requires skills in writing tests for teachers and eventually for students.

3 System Overview

3.1 Motivations

Several ideas motivated the creation of a new software.

The first one is to simplify the process of programming assignment creation. During this step, the teacher must define program functionalities he or she wants to assess in the case tests are not students' charge. Usually, it is done by writing xUnit tests that require specific skills. Teachers in charge of basic programming do not all master this domain and give up the tool. To avoid this fact, the idea is to provide a test generator requiring just a limited knowledge in testing.

The second is to make the tool smart. Thanks to the web-based scheme, compiling, executing and testing program take place on the server side and therefore is transparent for the end user. There is no need for special tool on the client side, only a browser and an editor to write the code. The student can test his program from everywhere and every device.

Last, an application is used on a mobile device only if it is designed especially for these kinds of target, time saving is essential for mobile users. This requires a special attention on software design, namely ergonomics and performances.

The software is aimed to address three criteria: adaptability, scalability and simplicity.

3.2 Technical Overview

An overview of the system (Fig. 1) is presented hereafter. It points out the main components: server, database and roles: teacher and student.

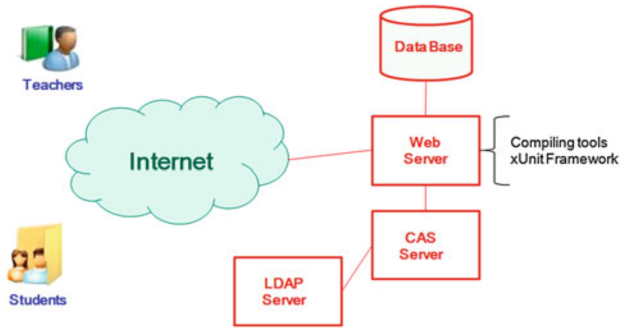


Fig. 1 System overview

Server and database implement Linux, Apache, MySQL and PHP. This architecture has come to be known simply as LAMP. The application is powered by Laravel Framework [7]. Status and information about users come from CAS server connected to a LDAP server.

3.3 Conception

We have two different roles: teacher and student.

3.3.1 Teachers

Academic staff module's main functionalities (Fig. 2) are project's management and project's life.

The first one consists of project creation and management including a user-friendly xUnit-tests generator (more information in Sect. 2.4). Once a project is created, only owners can access the project and his management.

From the projects life module, teachers are able to launch tests for a student and have access to the results pages. Different views are available for the results depending on the information needed. Student result view shows detail information for every tests. Group view shows only the project marks for all students, details are accessible by clicking on the student name.

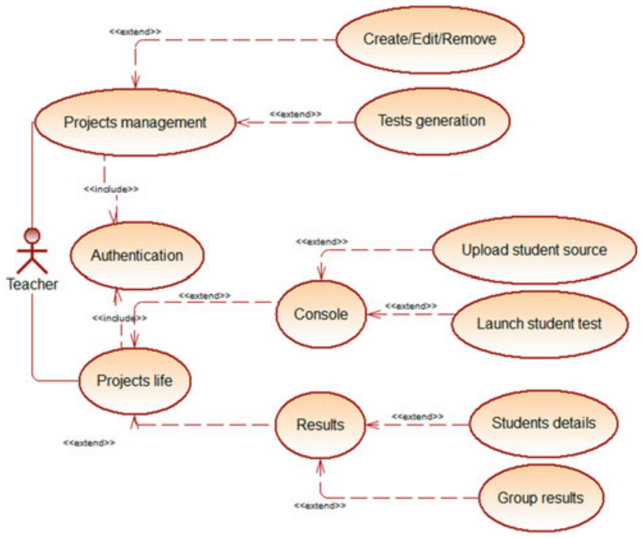


Fig. 2 Academic staff user case

3.3.2 Student

Student module (Fig. 3) implements only projects life with a different level of information in the results view.

Students can test the projects corresponding to their class and have access only to their own results without details. They cannot see the tests details but only if the functions tested work or not.

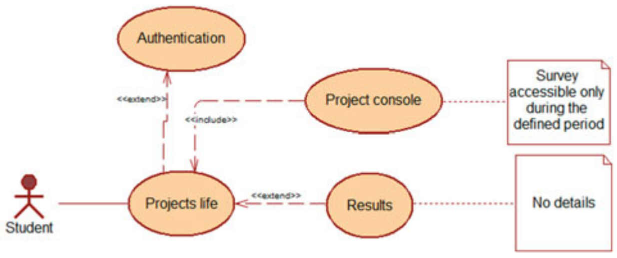


Fig. 3 Student user case

3.3.3 Authentication

The software must be adaptable to every university academic organisation requirements. To respect this criterion, there is no link with our information system apart from the CAS server, which is easy to adapt to any kind of authentication.

3.3.4 Responsive Web Design

The emergence of the "Bring Your Own Device" is a wide phenomenon and is even stronger in student population [8]. Academic organisations have to adapt their tools for this new trend and therefore must be compatible with any kind of devices (computer, smart phones, tablets and certainly in the future other smart machines, etc).

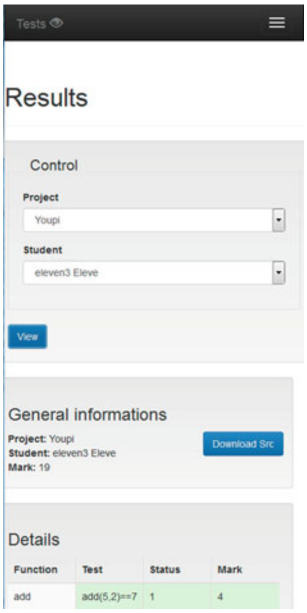
The software is responsive design (Fig. 4), it uses the twitter bootstrap framework version 3.

3.3.5 Adaptability

The software is adaptable to any programming language by means of modules.

Currently, it is efficient for C and Java languages.

Fig. 4 Results mobile design



Results

Control

Project: Youpi

Student: eleven3 Eleve

General informations

Project: Youpi
Student: eleven3 Eleve
Mark: 19

Details

Function	Test	Status	Mark
add	add(5,2)==7	1	4

3.4 *Tests Design*

3.4.1 Considerations

The tool focuses only on unit tests and especially on limit values or exceptions, for instance the test of the division by zero. The pedagogical purpose is to make students aware of these problems.

3.4.2 Tests Conception

A project contains functions to test (Fig. 5). Each function contains several tests. A test consists of verifying an assertion for the specific function. For instance, if the function is the division, the tests could be for the first one the verification of the operation 8/4, the second 9/4 and the third 9/0.

The implementation uses weak relationships to improve the database queries performances.

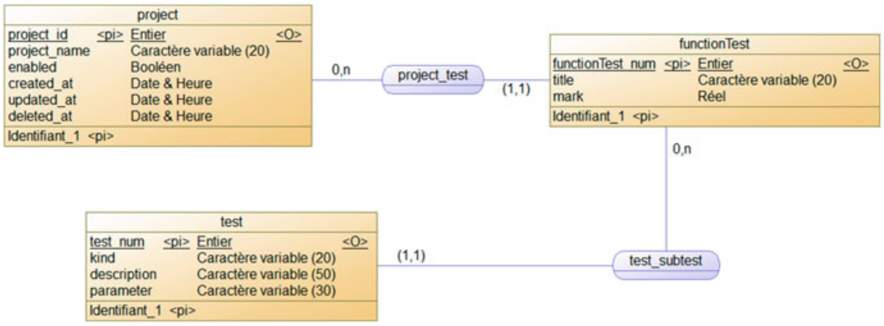


Fig. 5 Tests database

3.4.3 Test Generator

A test generator limits the xUnit skills required as explained in Sect. 2.2. The tool proposes a limiting set of xUnit function to the user, depending on the language chosen for the project (Fig. 6).

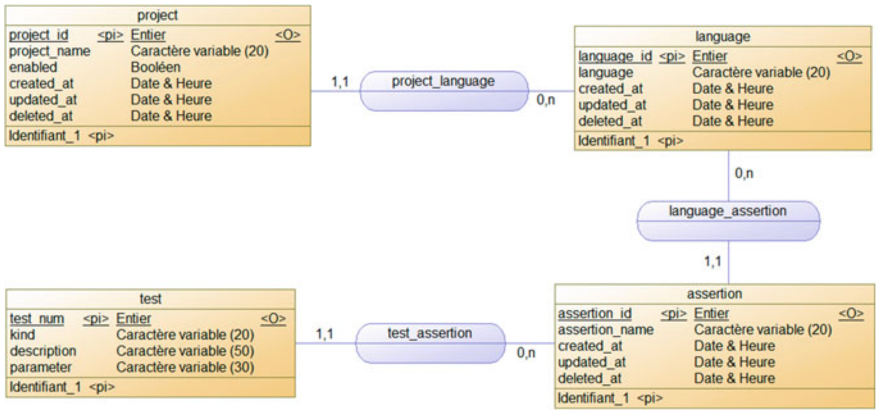
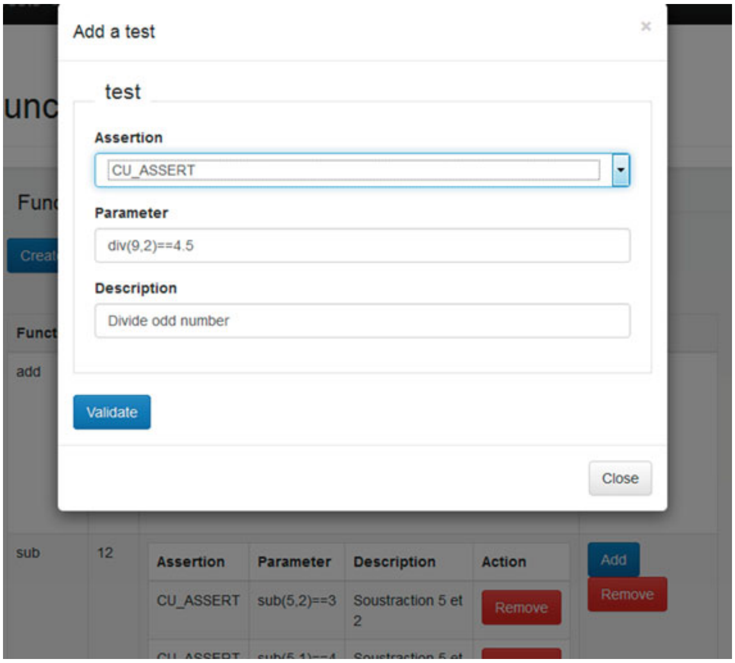


Fig. 6 xUnit assertions and language database

The knowledge required to create a test suite is just the parameters to pass to the assertion function as shown in Fig. 7.

A structure of a xUnit program includes:

- a list of tests to run
- parameters to indicate the form of the results
- a runner


Add a test

test

Assertion: CU_ASSERT

Parameter: div(9,2)==4.5

Description: Divide odd number

Assertion	Parameter	Description	Action
CU_ASSERT	sub(5,2)==3	Soustraction 5 et 2	Remove

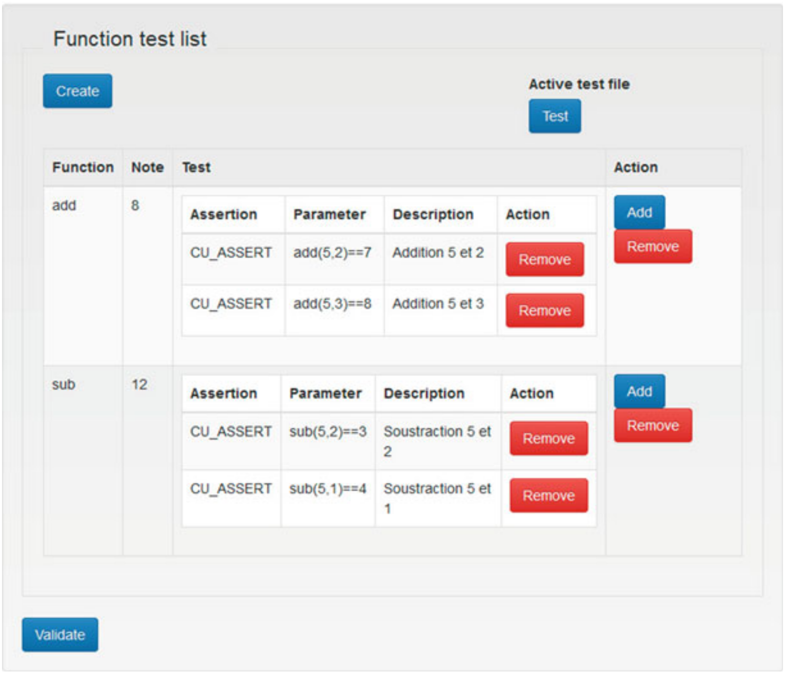
Fig. 7 Test creation for a specific function

The generator (Fig. 8) masks everything for the end user apart from the tests specification as explained above. On test suite completion, the running test file is generated. The teacher can access it if he or she wants to understand the generation process. The project is now ready to use.

3.4.4 Results

The results generated by a xUnit program are either text or XML based. Our choice is to use the last one because it is easier to parse to feed the database.

The feedback depends on the status of the user (Fig. 9): student on the left side and teacher on the right side.



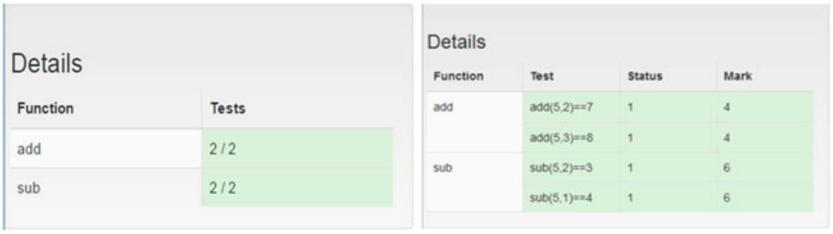
Function test list

Function	Note	Test	Action
add	8		
sub	12		

Assertion	Parameter	Description	Action
CU_ASSERT	add(5,2)==7	Addition 5 et 2	Remove
CU_ASSERT	add(5,3)==8	Addition 5 et 3	Remove

Assertion	Parameter	Description	Action
CU_ASSERT	sub(5,2)==3	Soustraction 5 et 2	Remove
CU_ASSERT	sub(5,1)==4	Soustraction 5 et 1	Remove

Fig. 8 Test generator



Details

Function	Tests
add	2 / 2
sub	2 / 2

Details

Function	Test	Status	Mark
add	add(5,2)==7	1	4
	add(5,3)==8	1	4
sub	sub(5,2)==3	1	6
	sub(5,1)==4	1	6

Fig. 9 Results from student and teacher's view

4 Experimentation

4.1 General Considerations

The software was efficient at the end of November 2014. It is currently testing by a teacher with xUnit skills. The plan is to use the tool on March for a programming assessment with first year students who are beginners in computer science.

The experiment was to adapt an existing assessment to integrate it into the tool.

4.2 From Teacher Point of View

The integration has been successful and has not required too much time. The test generator is simple to use but needed some explanations. It could be more instinctive or need more documentation.

The feeling of the teacher is that the tool is easy to use and will save times for correction. The idea is to use it also during lessons.

The experimentation identified limits. First, the tool cannot be used for creative projects but it was not the initial target. Another problem appeared, the tool is not able to deal with interactions. It is a common task for beginners to create a function, which requires user interaction, enter a variable by answering a question for instance.

4.3 From Students Point of View

The experimentation will start on March for assessment and over the rest of the year for lessons.

5 Conclusions and Future Work

This web-based system has been successfully experimented with a confirmed computer science teacher. Other experimentations will start on teachers with no tests skills.

We are thinking about evolutions for next year. First, we want to introduce the interaction possibility because the software first targets are computer science beginners. Also, we want to introduce quality tools such as sonar [9] and an anti-plagiarism tool. These tools can be very useful for teachers and will be completely transparent in terms of complexity for the end-user. Indeed, the main purpose is to make a tool easy to use.

References

1. Ihantola, P., Ahoniemi, T., Karavirta, T., Seppala, O.: A survey of automated assessment approaches for programming assignments. Comput. Sci. Educ. **15**(2), 83–102 (2005)
2. Ihantola, P., Ahoniemi, T., Karavirta, T., Seppala, O.: Review recent systems for automatic assessment of programming assignments. Comput. Sci. Educ. (2010)
3. Subba, L.T.: An anti-plagiarism add-on for web-CAT. Masters thesis, National University of Ireland Maynooth (2013)

4. Shah, A.: Web-CAT : A web-based center for automated testing. Master thesis, Faculty of Virginia Polytechnic Institute and State University (2003)
5. Spacco, J., Hovemeyer, D., Pugh, W., Hollingsworth, J., Padua-Perez, N., Emad, F.: Experiences with marmoset: Designing and using an advanced submission and testing system for programming courses. In: ItiCSE '06: Proceedings the 11th Annual Conference on Innovation and Technology in Computer Science Education. ACM Press, New York (2006)
6. http://marmoset.cs.umd.edu
7. http://laravel.com/
8. Emery, S.: Factors for consideration when developing a bring your own device in higher education. Master thesis, University of Oregon (2012)
9. http://www.sonarqube.org/