

XÂY DỰNG HỆ THỐNG ONLINE JUDGE CHO VIỆC HỌC LẬP TRÌNH

Nguyễn Ngọc Thủy*, Võ Việt Dũng, Lê Mỹ Cảnh, Trần Việt Khoa

Khoa Công nghệ thông tin, Trường Đại học Khoa học, Đại học Huế

* Email: nnthuy@gmail.com

Ngày nhận bài: 16/7/2020; ngày hoàn thành phản biện: 21/7/2020; ngày duyệt đăng: 02/10/2020

TÓM TẮT

Các hệ thống E-learning, học tập trực tuyến ra đời ngày càng nhiều để đáp ứng nhu cầu tự học của sinh viên. Việc xây dựng một mô hình như vậy để hỗ trợ cho sinh viên ngành công nghệ thông tin là một nhu cầu cấp thiết, qua đó nâng cao trình độ kiến thức cho sinh viên. Việc đánh giá năng lực sinh viên sẽ trở nên dễ dàng hơn bằng việc đưa ra các bài tập sát với nội dung môn học và phù hợp với nhiều cấp độ, khả năng của người học. Trong bài báo này, với mục tiêu giúp cho người học nâng cao hơn kết quả học tập. Chúng tôi đề xuất một hệ thống lập trình trực tuyến với các tính năng đặc thù, phù hợp với sinh viên ngành Công nghệ Thông tin thuộc Đại học khoa học Huế. Các kết quả đánh giá, khảo sát thu được đã cho thấy tính hiệu quả, thu hút người học của mô hình này

Từ khóa: Máy chấm trực tuyến, kỳ thi, bài toán.

1. ĐẶT VẤN ĐỀ

Đặc thù riêng của ngành Công nghệ thông tin đó là sinh viên được dạy và học thực nghiệm rất nhiều trên máy tính với nhiều môn học liên quan đến lập trình. Có thể liệt kê ra một số môn học mà bất kỳ sinh viên ngành công nghệ thông tin phải trải qua và nắm bắt làm chủ, gồm:

- Kỹ thuật lập trình.
- Ngôn ngữ lập trình: C/C++, Java, ...
- Cấu trúc dữ liệu và thuật toán.

Nhận thấy nhu cầu thực hành các môn học trên cần có hệ thống bài tập, và quy trình chấm bài cho nên chúng tôi đã nghiên cứu và phát triển một hệ thống dạy và chấm bài trực tuyến nhằm phục vụ việc học, luyện tập kỹ thuật lập trình cho sinh viên ngành Công nghệ Thông tin của Đại học Khoa học Huế.

Nội dung của hệ thống đang xây dựng là một tập hợp các bài tập cho các môn học: Kỹ thuật lập trình I, II, ôn luyện OLP/ACM. Ngoài ra còn có tổ chức các cuộc thi luyện tập định kỳ dành cho tất cả mọi người và chúng tôi cũng đã thử nghiệm các kỳ thi giữa kỳ và kết thúc môn cho môn học đặc thù là Kỹ thuật lập trình I, II.

2. ĐỐI TƯỢNG VÀ PHƯƠNG PHÁP NGHIÊN CỨU

2.1. Đối tượng nghiên cứu

Bài toán (Problem hoặc Task): Đây là đối tượng quan trọng nhất của hệ thống, là sự chuyển tải nội dung môn học đến sinh viên, thể hiện quan hệ hai chiều giữa giảng viên và sinh viên. Nội dung của bài toán bắt buộc phải bám sát đề cương môn học và ít nhiều được giảng dạy lý thuyết ở trường. Bài toán sẽ được tạo ra bởi tác nhân là giảng viên.

Sinh viên: Sinh viên là đối tượng chính thứ hai nhưng cực kỳ quan trọng của hệ thống mà ta thường còn gọi là người sử dụng. Sinh viên sử dụng hệ thống phải giải quyết (lập trình) rất nhiều bài toán với nhiều thể loại, mức độ khó dễ khác nhau nhằm nâng cao kiến thức về lập trình. Khi gắn kết trách nhiệm, họ phải tham gia các kỳ thi để có một kết quả đánh giá, ví dụ họ phải tham gia thi giữa kỳ, kết thúc kỳ để lấy điểm cho môn học kỹ thuật lập trình.

Giảng viên: Giảng viên là tác nhân chính của hệ thống, công việc của họ là xây dựng bài toán cho hệ thống và triển khai các kỳ thi. Một bài toán sẽ được gắn liền với các thuộc tính như: thể loại hay dạng toán, điểm số, thời gian thực hiện, số lượng test case. Tất cả các giá trị thuộc tính trên đều được quyết định bởi giảng viên.

2.2. Phương pháp nghiên cứu

Trọng tâm của hệ thống chính là đối tượng bài toán, việc xây dựng được một bài toán chính là điểm quan trọng nhất, để xây dựng nó ta dựa vào các yếu tố sau:

- Dựa trên đề cương môn học: Đây là yếu tố cơ bản để xây dựng một bài toán, việc bám sát đề cương môn học giúp sinh viên tiếp cận tốt hơn kiến thức lý thuyết được truyền thụ trên giảng đường. Ngoài ra việc cô đọng đề cương môn học vào các bài toán cũng là một tiêu chí của khảo thí, của kiểm tra chất lượng môn học [1].

- Dựa trên tài nguyên của các website thi nổi tiếng: Tài nguyên trên Internet được xem như một kho báu, việc khai thác đúng và kịp thời đem lại cho ta nhiều lợi ích. Trong lĩnh vực lập trình trực tuyến, nhiều trường học, công ty đều đã tạo ra và duy trì một số website cho lập trình cạnh tranh, một vài website tiêu biểu như bảng 1:

- Dựa trên kiến thức, kinh nghiệm của giảng viên: Quan trọng hơn hết vẫn là kiến thức và kinh nghiệm của giảng viên, quá trình tích lũy trong thời gian dạy và

ngiên cứu giúp giảng viên đưa ra được những bài toán tốt nhất để sinh viên có thể tiếp cận tốt nhất môn học của mình.

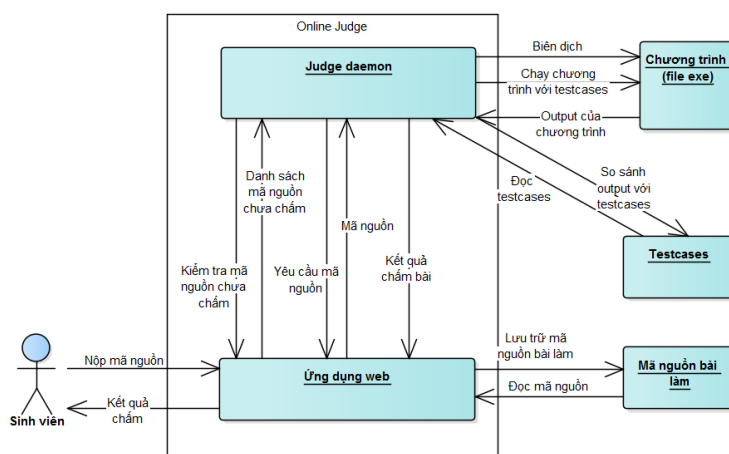
2.3. Phân tích, thiết kế và triển khai hệ thống

Sau khi tiến hành khảo sát, nghiên cứu chức năng người sử dụng (hầu hết các website trực tuyến đều có cấu trúc giống nhau), chúng tôi triển khai cài đặt hệ thống bằng phần mềm mã nguồn mở DOM Judge [3]. Đây là một phần mềm mã nguồn mở chạy ổn định được nhiều trường Đại học chọn làm nhân để cài đặt hệ thống. Một số thành phần chính của DOMjudge:

- Các chức năng dành cho bài toán (Problem).
- Các chức năng dành cho kỳ thi (Contest).
- Các chức năng dành cho thí sinh (Contestant) và đội thi (Team).

2.3.1 Người dùng hệ thống

1) Biểu đồ ngữ cảnh hệ thống (System Context Diagram).



Hình 1. Biểu đồ ngữ cảnh của hệ thống

2) Biểu đồ usecase của hệ thống thể hiện các chức năng chính của hệ thống gồm:

i) Người dùng Sinh viên:

- Tìm kiếm và liệt kê bài tập: Chức năng cho phép sinh viên liệt kê danh sách các bài tập theo thể loại.
- Nộp bài và xem kết quả: Nộp bài giải sau khi làm bài và xem kết quả, nếu bài chưa đúng thì cho phép làm lại.
- Tham khảo mã nguồn người khác: Sau khi nộp bài thành công, sinh viên có thể xem bài giải của người khác để học hỏi và rút kinh nghiệm.

- Tham gia các kỳ thi: Cho phép sinh viên tham gia bất kỳ cuộc thi nào.

ii) Người dùng giảng viên:

- Quản lý hệ thống bài tập: Là chức năng ra bài tập, có thể chèn, sửa, xóa, cập nhật bài toán lên hệ thống.

- Quản lý các kỳ thi: Tạo kỳ thi bao gồm chọn bài, thiết lập ngày, giờ thi, thời lượng cuộc thi, cách chấm và phân điểm cho từng bài thi.

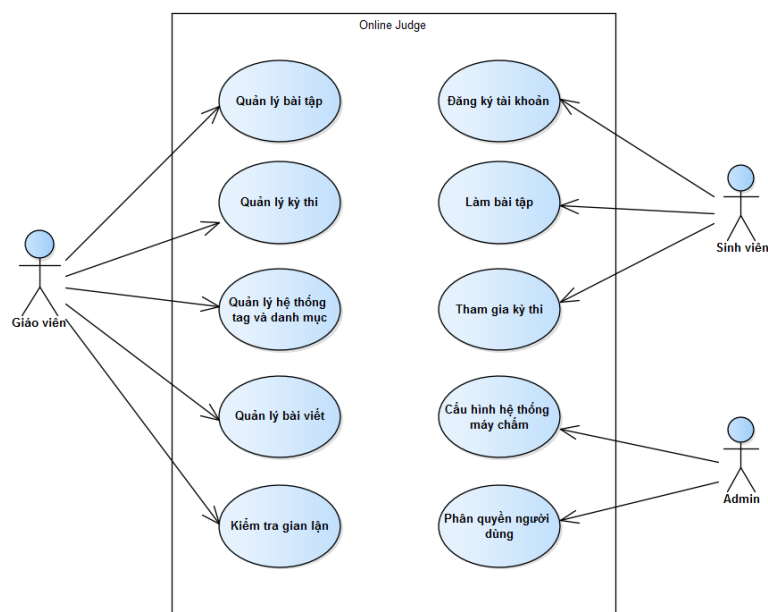
- Quản lý bài viết: Các bài viết hướng dẫn học tập hỗ trợ kiến thức cơ bản cho sinh viên.

- Kiểm tra gian lận: Kiểm tra thông tin gian lận dựa vào chức năng so sánh mã nguồn.

iii) Người quản trị:

- Cấu hình thông số của hệ thống máy chấm.

- Phân quyền người sử dụng.



Hình 2. Biểu đồ use-case của hệ thống

2.3.2 Quy trình xây dựng một bài toán

Các bài toán đều được xây dựng thông qua các bước sau:

Bước 1. Định nghĩa bài toán.

Bước 2. Xây dựng test case.

Bước 3. Kiểm tra test case.

Bước 4. Thực hiện việc kiểm chứng bài toán trên hệ thống bằng cuộc thi.

Việc định nghĩa bài toán thường dựa vào việc phân lớp bài toán theo chủ đề của các môn học, như vậy ta sẽ có các chủ đề cơ bản sau trong lập trình: Các bài toán Ad-hoc, tức là bài toán mang tính chất cài đặt để tìm hiểu về ngôn ngữ lập trình, lý thuyết số, tổ hợp, lý thuyết đồ thị, xử lý xâu, các bài toán hình học, các cấu trúc dữ liệu, các thuật toán tìm kiếm và sắp xếp.

Ngoài việc phân lớp bài toán theo chủ đề ta còn có thể định nghĩa bài toán theo từ khóa. Việc định nghĩa bài toán theo từ khóa giúp người học tiếp cận bài toán liên quan một cách cụ thể hơn, hình thành các kỹ thuật lập trình theo kỹ năng thực hành. Các từ khóa là các thuật ngữ trong tin học gắn liền với các thuật toán, các cấu trúc dữ liệu, ví dụ như: chia để trị, quay lui, tìm kiếm nhị phân, quy hoạch động, tham lam[2].

Sau khi định nghĩa bài toán với nội dung cần giải, không gian bài toán được xác định thông qua ràng buộc về mặt thời gian, dữ liệu. Quy trình tiếp theo là xây dựng testcase, quy trình này gắn liền với tác giả của bài toán. Ở đây anh ta phải thực hiện giải bài toán trên có thể thực hiện bằng cách lập trình chi tiết, có thể tính toán bằng tay, sử dụng công cụ tính toán và kết quả là các bộ testcase với dữ liệu vào ra tương ứng [4]. Đây chính là các bài toán cụ thể riêng biệt cho bài toán tổng quát mà ta đã ra ở trên.

2.4. Đề xuất bổ sung môn học

Sau quá trình thực nghiệm liên tục cho các môn học Kỹ thuật lập trình I, II trong bốn năm học với kết quả khả quan:

- Thứ nhất: nâng cao trình độ người học được đánh giá khi các bạn sinh viên ra trường đi làm với doanh nghiệp. Kết quả thi Olympic Tin học Sinh viên toàn quốc vô cùng ấn tượng với 3 giải nhì, 6 giải 3, 3 giải KK (tính từ 2016-2019 với mỗi lượt thi chỉ 3 sinh viên).

- Thứ hai: nâng cao cách dạy, đổi mới phương pháp và nâng cao kiến thức truyền thụ khi giáo viên ra đề thi và bài tập.

- Thứ ba: việc thi cử hoàn toàn công tâm, kết quả chấm chính xác, nhanh và tiết kiệm.

Với nhiều ưu điểm như vậy chúng tôi mạnh dạn đề xuất với Nhà trường, Khoa mở rộng thêm nhiều môn học đặc thù nữa và đồng thời nhận chuyển giao hệ thống từ chúng tôi để vận hành tốt hơn, đưa hệ thống đến với nhiều giảng viên hơn nhằm phát triển nội dung hệ thống tốt hơn, sinh động hơn.

3. KHẢO SÁT VÀ ĐÁNH GIÁ

Để phát triển tốt hơn các chức năng của hệ thống cũng như khảo sát tính thực tiễn của ứng dụng. Chúng tôi khảo sát sinh viên thuộc nhóm người dùng của hệ thống

Chúng tôi đã xử lý thống kê và đưa ra kết quả sau như sau:

- Khảo sát về quá trình học tập trên trang web.

Dữ liệu khảo thời gian, địa điểm làm bài tập, ngôn ngữ lập trình được sử dụng kèm với phân lý thuyết đã học cũng như độ khó dễ của các dạng bài tập trong trang. Kết quả như ở bảng 2 sau:

Bảng 2. Dữ liệu khảo sát về học tập trên hệ thống

	Địa điểm	Tỷ lệ (%)	Ngôn ngữ	Tỷ lệ (%)	Bài tập	Tỷ lệ(%)
A	Ở trường	6.40	C/C++	87.10	Dễ	2.40
B	Thư viện	7.40	Pascal	7.30	Khó	24.60
C	Ở nhà	80.80	Java	3.20	Vừa phải	65.10
D	Khác	5.60	Khác	2.40	Quá khó	7.90

Với dữ liệu bảng 2 cho thấy con số trên 80% đối với sinh viên tham gia làm bài tập trên hệ thống ở nhà, ngôn ngữ được sử dụng trong lập trình là C,C++ theo lý thuyết dạy ở Trường và với tỷ lệ 65% sinh viên cho rằng độ khó của các bài tập trong hệ thống ở mức độ vừa phải.

- Khảo sát về chức năng của hệ thống

Mục tiếp theo chúng tôi khảo sát về chất lượng các chức năng của hệ thống gồm: Tính chặt chẽ của hệ thống chấm bài, tính năng chống gian lận, tính ổn định của hệ thống, tính hiệu quả và cấp thiết của hệ thống.

Bảng 3. Dữ liệu khảo sát về tính năng của hệ thống

	Chặt chẽ	Tỷ lệ (%)	Ổn định	Tỷ lệ (%)	Hiệu quả	Tỷ lệ(%)
A	Rất chặt chẽ	73.60	Rất ổn định	91.30	Hiệu quả	78.60
B	Chặt chẽ	15.20	Ổn định	4.00	Không hiệu quả	4.80
C	Còn lỗi	10.40	Còn lỗi	2.40	Rất hiệu quả	11.10
D	Khác	8.00	Khác	2.40	Khác	5.60

Với dữ liệu ở bảng 3 cho thấy có trên 73% sinh viên cho rằng hệ thống chấm bài rất chặt chẽ, tỷ lệ 91% cho rằng hệ thống chạy ổn định với nhiều chức năng như: chấm bài theo testcase, refresh điểm ngay lập tức nhờ công nghệ nodejs, chức năng chống gian lận với các hàm API được cung cấp bởi trường Đại học Stanford đã đem lại hiệu quả cao trong học tập với tỷ lệ bình chọn 78%.

4. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

4.1 So sánh việc dạy trực tuyến bằng phần mềm của Google và hệ thống Online Judge

Thực ra việc học trên phần mềm của Google chẳng khác nhiều với học trực tiếp và có giới hạn hơn về mặt giao tiếp giữa người dạy và người học. Sự tương tác không tốt giữa người dạy và người học dẫn đến người dạy phải đầu tư nhiều về bài tập mới mong chuyển tải nội dung đến người học. Tuy nhiên, phần bài tập mà người dạy công phu chuẩn bị cũng không có kết quả tốt nếu việc chấm bài cũng thực hiện một cách thủ công. Một phần do đặc thù môn học mà việc kiểm tra kiến thức người học khác với các môn học khác (Google đưa ra các test dưới dạng trắc nghiệm).

Trong khi đó, hệ thống Online Judge của chúng tôi thực hiện việc kiểm tra kiến thức người học tốt hơn. Và chỉ cần tương tác với người học qua hệ thống khác như email, facebook chúng tôi có thể lập lịch học và thực hành các môn đặc thù của ngành một cách hiệu quả.

Tuy nhiên, nếu có sự kết hợp của hai hệ thống thì việc dạy và học trở nên tuyệt vời, và cụ thể học kỳ vừa qua chúng tôi đã thực hiện. Bỏ qua một số yếu tố khách quan như cơ sở vật chất của người học.

4.2 Kết luận

Qua việc xây dựng thành công hệ thống và 5 năm sử dụng. Chúng tôi đánh giá được một số kết quả sau:

- Chủ động trong việc phân bố lịch dạy, nội dung dạy: Chúng tôi đã dạy thực tế cho sinh viên và đã kết thúc thành công với việc học và thi trên hệ thống.

- Các kỳ thi, bài tập được chấm một cách công khai, minh bạch và có kết quả ngay lập tức: Việc phân loại thí sinh còn dựa vào chức năng chấm điểm theo testcase và là tính năng do chúng tôi phát triển. Do hệ thống có thiết lập chức năng phát hiện sao chép bài nên đã hạn chế tối đa tính gian lận trong khi thi.

- Sinh viên có thể chủ động học tập mọi lúc, mọi nơi: Do hệ thống chạy trên nền Web nên hầu hết các sinh viên đều có thể chủ động làm bài bất kỳ thời gian nào, nhật ký nộp bài hệ thống cho thấy hầu hết thời gian trong ngày 24/24 đều có người làm bài tập. Việc chủ động học tập và có bảng xếp hạng cao cho thấy được kết quả thi tốt của các sinh viên khi thi.

4.3 Hướng phát triển

Hệ thống được xây dựng dựa trên sự yêu cầu thay đổi về phương pháp dạy và học trong nhà Trường, tại thời điểm viết bài báo nó đã ít nhiều đáp ứng được một số lượng nhất định nhu cầu học tập, thực hành, tự học của Sinh viên Khoa Công nghệ Thông tin, Đại học Khoa học Huế. Trong tương lai chúng tôi sẽ đề xuất kinh phí để tiếp

tục triển khai thêm một số môn học mà thay vì học nhiều lý thuyết có thể chuyển đổi sang học trực tuyến trên hệ thống với nhiều ưu điểm nổi bật. Những môn học được đề xuất:

- Lập trình Java cơ bản.
- Lập trình hướng đối tượng.
- Ngôn ngữ truy vấn có cấu trúc (SQL).
- Cấu trúc dữ liệu và giải thuật.
- Ngôn ngữ Ruby.

TÀI LIỆU THAM KHẢO

- [1]. Jaap Eldering, Nicky Gerritsen, Keith Johnson, Thijs Kinkhorst, and Tobias Werth, *<https://www.domjudge.org/intro>*, Last modified: Sunday, 12-Feb-2017.
- [2]. Emil KELEVEDJIEV, Zornitsa DZHENKOVA, *Tasks and Training the Youngest Beginners for Informatics Competitions*, Olympiads in Informatics, 2008, Vol. 2 © 2008 Institute of Mathematics and Informatics, Vilnius, p 75-89.
- [3]. Sébastien COMBÉFIS, Jérémy WAUTELET, *Programming Trainings and Informatics Teaching Through Online Contests*, Olympiads in Informatics, 2014, Vol. 8 © 2014 Vilnius University, IOI, p 21-34
- [4]. Krassimir MANEV, Biserka YOVCHEVA, Milko YANKOV, Peter PETROV, *Testing of Programs with Random Generated Test Cases*, Olympiads in Informatics, 2010, Vol. 4 © 2010 Institute of Mathematics and Informatics, Vilnius, p 76-86

IMPLEMENTATION OF AN ONLINE JUDGE SYSTEM FOR PROGRAMMING TRAINING IN UNIVERSITY

Nguyen Ngoc Thuy*, Vo Viet Dung, Le My Canh, Tran Viet Khoa

Faculty of Information Technology, University of Sciences, Hue University

* Email: nnthuy@gmail.com

ABSTRACT

The number of E-learning system has increasingly emerged to meet student's self-study needs. It is inevitable to build an E-learning system for Information Technology (IT) students, therefore they are enhanced the knowledge.. Classifying students now becomes easier by giving tests which are close to the subject and suitable for levels and abilities of learners. . In this article, with the purpose of improving the learning results of learners, we propose an online judge system model in consideration of specific requirements of IT students of University of Sciences, Hue University. The results of assessments and surveys demonstrated the effectiveness and attraction of this system.

Keywords: contest, online judge, problems.



Nguyễn Ngọc Thủy sinh ngày 21/04/1976 tại Quảng Bình. Năm 1998, ông tốt nghiệp cử nhân ngành CNTT; năm 2006, ông tốt nghiệp thạc sỹ ngành CNTT tại Trường Đại học khoa học, ĐH Huế. Hiện nay ông là giảng viên tại Trường Đại học khoa học, ĐH Huế.

Lĩnh vực nghiên cứu: Mạng máy tính.



Võ Việt Dũng sinh ngày 2/01/1978 tại Hà Nội. Năm 1999, ông tốt nghiệp cử nhân ngành CNTT; năm 2017, ông tốt nghiệp thạc sỹ ngành CNTT tại Trường Đại học khoa học, ĐH Huế. Hiện nay ông là giảng viên tại Trường Đại học khoa học, ĐH Huế.

Lĩnh vực nghiên cứu: Mạng máy tính.



Lê Mỹ Cảnh sinh ngày 26/07/1988 tại Quảng Trị. Năm 2010, ông tốt nghiệp cử nhân ngành CNTT tại Trường Đại học Khoa học, ĐH Huế. Năm 2015, ông tốt nghiệp thạc sỹ ngành CNTT tại Đà Bắc, Đà Loan. Hiện nay ông là giảng viên tại Trường Đại học khoa học, ĐH Huế.

Lĩnh vực nghiên cứu: Công nghệ phần mềm.



Trần Việt Khoa sinh ngày 19/06/1972 tại Thanh Hóa. Năm 1995, ông tốt nghiệp cử nhân ngành Toán Tin; năm 2004, ông tốt nghiệp thạc sỹ ngành CNTT tại Trường Đại học khoa học, ĐH Huế. Hiện nay ông là giảng viên tại Trường Đại học khoa học, ĐH Huế.

Lĩnh vực nghiên cứu: Khoa học máy tính.