Web application implementation of Android programming learning assistance system and its evaluations

Article *in* IOP Conference Series Materials Science and Engineering · February 2021

DOI: 10.1088/1757-899X/1073/1/012060

CITATIONS
8

READS
65

7 authors, including:

Yan Watequlis Syaifudin
Politeknik Negeri Malang
58 PUBLICATIONS **102** CITATIONS
SEE PROFILE

Minoru Kuribayashi
Okayama University
162 PUBLICATIONS **858** CITATIONS
SEE PROFILE

Mustika Mentari
Politeknik Negeri Malang
31 PUBLICATIONS **38** CITATIONS
SEE PROFILE

Pramana Yoga Saputra
Politeknik Negeri Malang
26 PUBLICATIONS **37** CITATIONS
SEE PROFILE

Web application implementation of Android programming learning assistance system and its evaluations

Web application implementation of Android programming learning assistance system and its evaluations

Y W Syaifudin[1,*], N Funabiki[1], M Kuribayashi[1], M Mentari[2], P Y Saputra[2], Y Yunhasnawa[2] and F Ulfa[2]

[1] Dept. Electrical and Communication Engineering, Okayama University, Okayama, Japan
[2] Dept. Information Technology, State Polytechnic of Malang, Malang, Indonesia

*p52m5jeb@s.okayama-u.ac.jp

Abstract. Currently, Android has been installed on more than 70 percent of the smartphone devices in the world, which has significantly increased the need for Android-based application engineers. Many IT departments in universities and professional schools have included the Android-based programming into the main subjects. To improve Android programming educations, we have developed Android Programming Learning Assistance System (APLAS), a platform to accommodate the students learning Android programming independently. APLAS adopts the test-driven development method to realize automatic validations of students' answers to the programming assignments. Previously, APLAS has been implemented as a desktop offline application. In this paper, we present a Web application implementation to manage the learning activities online. This implementation consists of three main parts, namely, Validator, Web Interface, and Database System. To evaluate the effectiveness, we offered three learning topics, Basic UI, Basic Activity, and Advanced Widgets, into this Web application, and asked 60 students in Indonesia to solve them. The results showed that the Web application has well been accessed and smoothly utilized by them. The reliability of the validator program was confirmed by handling more than 180 submissions.

1. Introduction

Mobile applications have become one of the main fragments in the global economy after enormous achievements of smartphones. Then, mobile application programmers become one of the most popular and highly demanded jobs in the IT sector. Currently, Android has been installed on more than 70 percent of the smartphone devices in the world. The statistics shows that Google Play is the leading application store and provides more than 2.5 million applications in the 1st quarter of 2020 [1]. With this demand, many professional schools and universities have carried out the mobile programming courses, especially for Android application developments.

To improve the effectiveness of Android programming by students, we have proposed the Android Programming Learning Assistance System or APLAS [2]. APLAS provides a self-learning platform with automatic validations of the students' answers for Java-based Android programming, by adopting a test-driven development method. Unit testing is applied to validate the requirement specifications in the source code from a student. For unit testing in Android application, JUnit and Robolectric are adopted.

In previous implementations, students must solve the learning assignments using Android Studio and validate source codes by themselves using given test codes [3]. Then, the students report the validation results and source codes to the teachers. In this case, the teachers must check the validity of the answers one by one. APLAS has offered 13 learning topics with 8-10 tasks for each. Each task contains one or two test codes. If a teacher has 50 students, he/she must execute at least 400 test codes for one learning topic. This situation makes the teacher difficult to handle the students with many learning topics.

In this paper, we present the web application implementation with the automatic validation process for APLAS. This web application can be used by the students to obtain the learning materials, manage the assignments, and submit the results. In the server, the validator program runs in the background to detect new submissions and validate them automatically. The result of the validation process can be accessed on the web interface by the students and the teachers.

To evaluate the effectiveness, we implemented three learning topics, Basic UI [3], Basic Activity [4], and Advanced Widgets [5]. Then, we asked 60 students in Indonesia to solve the assignments and submit the results to the web application. The validation process has taken a long time relatively, depending on the number of test files, test methods, and types of unit testing. The results confirmed the reliability of the validator program that has validated than 180 submissions.

The rest of this paper is organized as follows: Section 2 studies related works in literature. Section 3 explains the overview of APLAS. Section 4 presents the model of web application. Section 5 describes the online validation method. Section 6 presents the evaluation results of web application and validation process to 60 students with its discussion. Finally, Section 7 concludes this paper with future works.

2. Related works

In 2012, Bosnić et. al. introduced ORVViS, a module in the Moodle learning management system, to validate source codes from students [6]. However, it did not support the grading system, including checking the source codes according to the given specification or assignment from teacher. It focuses mainly on syntax.

In 2013, Funabiki et al. proposed a Web-based Java Programming Learning Assistant System (JPLAS) for Java programming language self-learning [7]. JPLAS adopts JUnit for unit tests to validate source codes as assignment answers from students. In this system, a student writes a source code by reading the statement and the test code, then it will be tested at the server automatically.

In 2014, Yulianto and Liem introduced a source code analyzer to grade programming assignments automatically [8]. Their study produced an application called SCAGrader consisting of a Java engine application and web-based user interface.

In 2016, Staubitz et.al introduced CodeOcean as a web-based platform to practice programming course exercises [9]. It is designed to be used in programming MOOCs. Also, it allows teachers to plan programming exercises, which can be graded automatically using unit tests as a quality measurement.

Khan et al. presented AUTOGRADER in 2019, a tool to decide the correctness of programming assignments automatically [10]. It also examines the semantically different execution paths the codes and the reference implementation. This tool successfully decreases the teacher's efforts in writing test cases and delivers the grading more flawless.

3. Review of Android Programming Learning Assistance System (APLAS)

In this section, we review APLAS in our previous studies [2,5].

3.1. System overview

APLAS is a students' self-learning platform for Android programming learning with automatic validation features [3]. To validate source codes from students, APLAS adopts a Test-driven Development method based on Java unit testing. The developed project is validated by executing the given test codes. To validate the correctness of the program codes and the UI components in an Android project, APLAS adopts JUnit and Robolectric for integration testing. On these tools, APLAS provides a set of learning assignments to be solved by students individually.

3.2. System specification
APLAS implements a learning platform for the students to solve the learning tasks on PCs. Using Android Studio, a student can build a specific Android application to solve learning tasks and validate the built projects by him/herself using the given test codes. Android Studio offers tools to facilitate Android developments to work on tasks using the combinations of Java and XML.

3.3. Learning model
In APLAS, the learning materials consist of several topics. Each topic represents one learning assignment. To cover the wide contents in Android application programming subjects, we grouped the learning topics into four stages: User Interface, Interactive Application, Content Provider, and Service Interaction. For a topic, students will get a package consisting of guide documents, test codes, and supplement files. The test code is applied to validate the code written in Android project.

4. Proposal of web application for APLAS
In this section, we explain the design and implementations of the proposed web application for APLAS.

4.1. System architecture
The web application for APLAS adopts a client-server model, and has two main sections, a client section and a server section, as illustrated in Figure 1. The client section interacts with students, contains the learning platform, and the web client. The server section contains the validator and the web server. The web server accesses the database to manage the data.

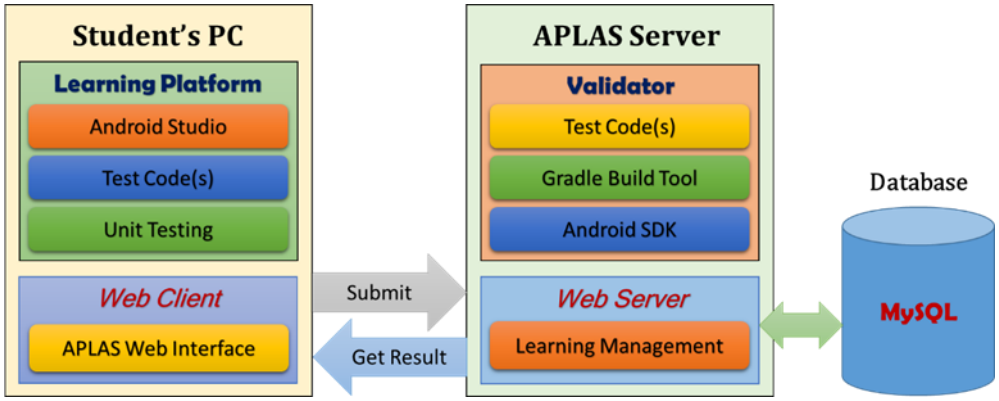

Figure 1. Improved system architecture of APLAS.

4.2. Learning platform
The learning platform is the part used by the students to solve the assignments on PCs. This part adopts *Android Studio* as the most popular *Integrated Development Environment (IDE)*, to create *Android* applications based on Java programming language. The project created by the student will be validated using given test codes that will run on *JUnit* and *Roboletric* manually. The student can get results directly in *Android Studio* when the execution of a test code finished.

4.3. Web application
To facilitate the students in learning managements and online assignment validations, the web application for *APLAS* provides the interface to students in Figure 2. The interface assists students in using the learning packages. After they finish a topic, they can submit the results of an assignment to this web application. Then, the system will validate them automatically, and record the learning progress in the database.

The validation process will be carried out by the validator, a thread program that validates any new submission from a student. The detailed validation results will be recorded in the database and will be

displayed on the web interface. If the validation results are incorrect, the messages will explain the incorrectness. Then, students should improve the answer codes and resubmit them to the web application. Students can monitor the achievements of their learning and track the history of their submissions.

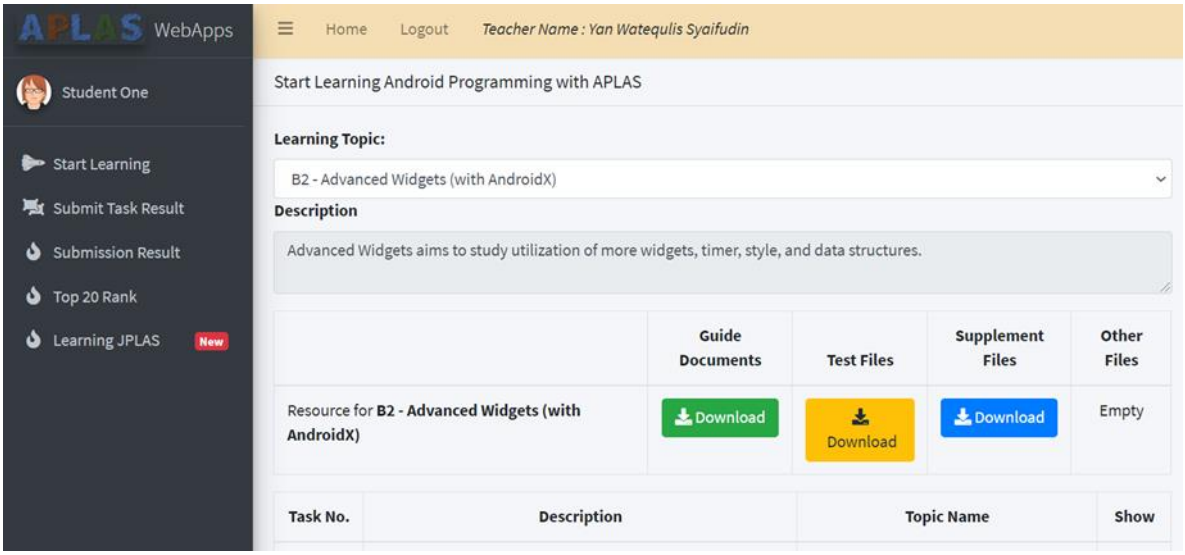

Figure 2. The web interface for students to access learning materials.

4.4. Validator

Validator is the background process written in Java and has the functions to detect new submissions from students and to validate each submission. The validation process is achieved by executing the given test codes using Gradle [11] in the form of an Android project. Gradle will integrate the Android project with the testing tools (JUnit and Robolectric) and Android SDK library. The validation results will be stored in the database so that it can be accessed from the web application. Each validation process will record the execution time, error messages (if any), and execution error messages (if occur).

5. Online validation model

In this section, we describe the student's learning process and the online validation model of their answers submitted using the web application.

5.1. Learning process

There are three steps for students to learn Android programming in APLAS using the web application, as shown in Figure 3. First, they have to open the web interface on a web browser and access the "Start Learning" page. As shown in Figure 2, they must choose a topic and download the learning materials that contain guide documents, test files, supplement files, and other files (if available). Then they use Android Studio to work with the assignment.

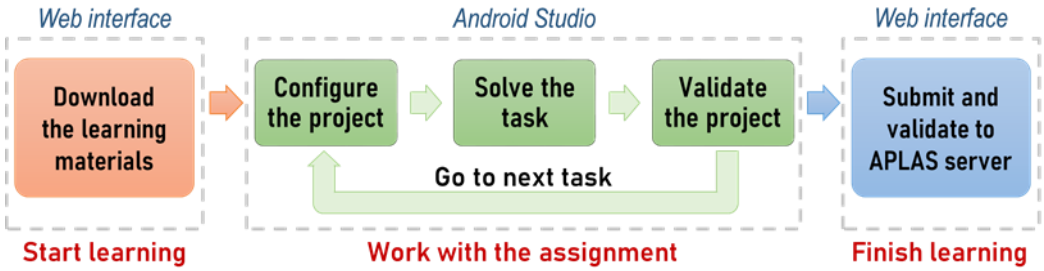

Figure 3. Learning process using web application.

A topic in APLAS contains several tasks that must be solved one by one sequentially, to finish the assignment for one topic. A student must follow the instructions in the guide document provided for each task. The student has to check the correctness of the answer using the provided test code(s) until he/she can solve all the tasks in the topic. Finally, he/she must submit the results and upload the necessary files on "Submit Task Results" page in the web application.

5.2. Validation method

After submitting the assignment results, the student has to wait for the results of the validation process in the APLAS server. As shown in Figure 4, there are five steps to validate a new submission. First, the validator receives a new submission from a student. Then, it prepares a folder that contains the Android project files associated with the submission. In this step, all the uploaded files will put in the folder. After the folder is completed, the validator will execute the test codes automatically using the Gradle command prompt, and get the results of validation. The results will be submitted into the database. The student can access the results using the web application in "Submission Results" page.

Figure 4. The validation process for a new submission by validator.

6. Evaluation

In this section, we clarify the effectiveness of the web application by evaluating it to students online.

6.1. Evaluation setup

To perform online implementation, we chose randomly 60 undergraduate students of a university in Indonesia that were taking a mobile programming course in an IT department. We asked them to download the learning materials for the three topics, Basic UI, Basic Activity, and Advanced Widgets. Then, they solved them on their PCs using Android Studio. Here, we provided two versions for each topic, Android support library version and AndroidX library version. The students can use one of them independently. For each student, we gave three days to solve three assignments. If some students failed to solve all or some of the assignments, we gave the second chance to solve and submit the results again one week later.

In this evaluation, we used the server that runs on Ubuntu Linux with VMWare virtual machine. This virtual server is with Processor AMD Opteron(TM) Processor 6238 4 Core 2.6 GHz, RAM 11 GB, Storage 100 GB, and internet address "aplas.polinema.ac.id".

6.2. Evaluation results

In general, the results for the three topics show that finally, all the students can pass all the tasks, as shown in Table 1. Only in solving Advanced Widgets topic, two students failed in the first chance. However, they can pass it in the second chance. The observation of students in accessing the web application shows that there is no serious problem on the registration, login, downloading, learning materials, submissions of assignment results, and getting validation results.

6.3. Discussion on evaluation results

All the students took various time to solve one learning topic from 63 minutes until 96 minutes on average. If compared with previous studies [2,5], the students' performances on online learning are faster than the results on in-class learning. Moreover, all the submissions can be validated by the validator successfully. It took time from 176 seconds until 294 seconds on average. It confirmed the effectiveness

of the online validation method. The detail of validation results can be accessed by students on the web application, including the messages to explain the incorrectness of the failed tasks.

In performance evaluations, we summarize the conditions that affect execution time:

- the number of test files and test methods has the big impact on the execution time,
- the number of assertion commands in a test file has the small impact on the total time, and
- the initial method of Robolectric takes around 85% of the total execution time.

Table 1. Validation results of students' submissions in three topics.

Topic	First chance		Second chance		Total		Avg. solving time	Avg. time to validate
	#Passed	#Failed	#Passed	#Failed	#Passed	#Failed		
Basic UI	60	0	0	0	60	0	96 min.	233 sec.
Basic Activity	60	0	0	0	60	0	63 min.	176 sec.
Advanced Widgets	58	2	2	0	60	0	78 min.	294 sec.

7. Conclusion

This paper presented the design and implementation of the web application with the online validation for Android Programming Learning Assistance System (APLAS). The students can download the learning materials from the web interface, manage the learning process, and submit the assignment's answers at any time, where the validator will validate it automatically. The evaluation results confirmed the effectiveness and reliability of our proposal. In future works, we will implement the web interface for the teachers, improve the accessibility of the web application, and the reliability of the validator.

References

[1] Clement J 2020 Google Play: number of available apps 2009-2020 *Statista GmbH*
[2] Syaifudin Y W, Funabiki N, Kuribayashi M, and Kao W C 2019 A proposal of Android Programming Learning Assistant System with implementation of basic application learning *International Journal of Web Information Systems* **16**(1) 115-135
[3] Syaifudin Y W, Funabiki N, and Kuribayashi M 2019 Learning model for Android programming learning assistant system *in Proc. IEICE General Conference 2019, Tokyo*
[4] Syaifudin Y W, Funabiki N, and Kuribayashi M 2019 An implementation and evaluation of basic activity topic for interactive application stage in Android programming learning assistance system *in Proc. FIT 2019, Okayama, Japan*
[5] Syaifudin Y W, Funabiki N, and Kuribayashi M 2020 An implementation and evaluation of Advanced Widgets topic for interactive application stage in Android programming learning assistance System *in Proc. 8th Int. Conf. Inf. and Edu. Tech., Okayama*
[6] Bosnić I, Mihaljević B, Orlić M, and Žagar M 2012 Source code validation and plagiarism detection - technology-rich course experiences in *Proc. 4th Int. Conf. Comp. Supp. Edu., Porto*
[7] Funabiki N, Matsushima Y, Nakanishi T, Watanabe K, and Amano N 2013 A Java programming learning assistant system using test-driven development method *IAENG Int. J. Comp. Sci.* **40**(1) 38-46
[8] Yulianto S V and Liem I 2014 Automatic grader for programming assignment using source code analyzer in *Proc. Int. Conf. Data. Soft. Eng., Bandung, Indonesia*
[9] Staubitz T, Klement H, Teusner R, Renz J, and Meinel C 2016 CodeOcean - a versatile platform for practical programming exercises in online environments in *Proc. Glo. Eng. Edu. Conf., Abu Dhabi*
[10] Khan I A, Iftikhar M, Hussain S S, Rehman A, Gul N, Jadoon W, and Nazir B 2020 Redesign and validation of a computer programming course using inductive teaching method *PLOS ONE* **15**(6) 1-21
[11] DiMarzio J 2017 *Beginning Android programming with Android Studio* (John Wiley & Sons) 2017