Ajax Based Exam Engine with Tagging System to Improve Learning

Asroni[1], Minhajuddin K. Abdurrahim[2], Cahya Damarjati[3]

[1,2,3]*Universitas Muhammadiyah Yogyakarta, Jln.Brawijaya, Tamantirto, Kasihan, Bantul, Yogyakarta 55183, Indonesia*
[1]*asroni@umy.ac.id,* [2]*minhajuddin.kasman.2013@ft.umy.ac.id,*
[3]*cahya.damarjati@umy.ac.id*

Abstract

In this paper, we propose exam engine software with tagging system to help students' study. With this tagging system, they can analyze their subject learning through exam they have done from time to time. When the students see their reports, the students will decide which subjects are low in grade and should be re-studied. We use AJAX technology to enrich the user experience of this exam engine. After we test all feature with unit testing, this exam engine is proven to runs well and do benefits for students.

Keywords: *Exam engine, Tagging System, AJAX*

1. Introduction

The assessment test is one of the activities held to perceive if the candidates are competent. Results from the test can be used as a benchmark for tests to come. Annually, most of the graduated senior high school student prefers to continue their study to the higher level like colleges. There are two ways to enroll in colleges those are the test and non-test way. Many choose test way instead of non-test because, in a non-test way, achievements will be used as consideration if the students passed the criteria or not while test way allows every student even those with no past achievements to enroll by completing tests. Test results depend on strategies made before the test is held like student's study at home, study time extension at school, etc. With these efforts, students should evaluate what they learned, recap evaluation results and categorize its answers to specific topics so that they can compare the results one another. With this activity, students can discover their weakness and strength in topics evaluated.

We have developed exam engine using ASP.NET MVC [1] to integrate Web Design exam, Database Design, Software Engineering, and Programming. MVC pattern is modular, with the model, the view, and the controller. Then using AJAX technology [2], it could achieve partial refresh and rapid on-demand access data. Combined AJAX with ASP.NET MVC technology, it could support an interactive platform for high-performance exam engine [3–5]. To achieve these objectives, having an automated system which has computer-based test features would be the solution. There are many parties who provide such systems. Although categorizing questions based on topics is important for achieving the objectives, the systems provided don't categorize test questions to specific topics. Moreover, according to the result of UNBK (Computer-based National Exam) 2015 candidate's questionnaire, there are worries within candidates about system failure and other obstacles when facing the UNBK. To quantitatively measure the quality of the assigned categories (tags) we adopted the exam name and exam score measures, which have been widely used in the area of information retrieval [6].

2. Method

In this project, we designed the system using UML (Unified Modelling Language) tool which allows us to visualize and document analysis results and design the system with syntaxes in visual modeling.

2.1. Activity Diagram (Teacher)

In Figure 1, teachers must authorize themselves before attempting to visit exam lists page. In exam lists page, teachers can do the following actions: (1) create an exam, (2) create a question and (3) complete exam details like adding candidates and exam questions. After creating an exam or question, teachers will be redirected back to exam lists page while exam detail process brings teachers to exam detail page where they can perform actions related to exam management.
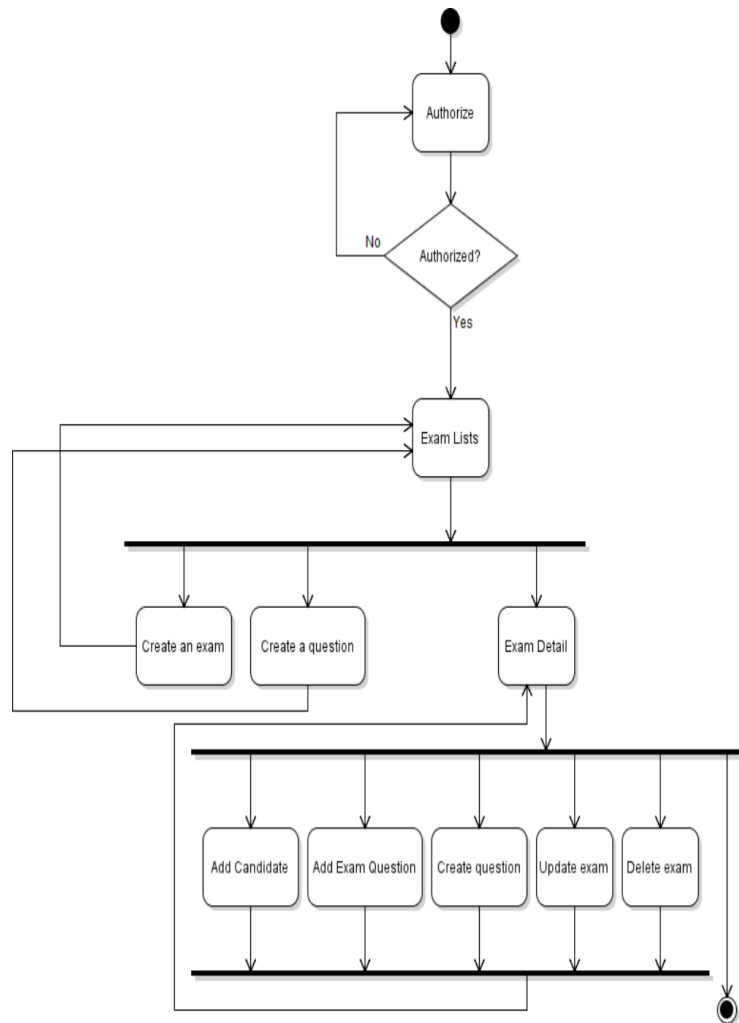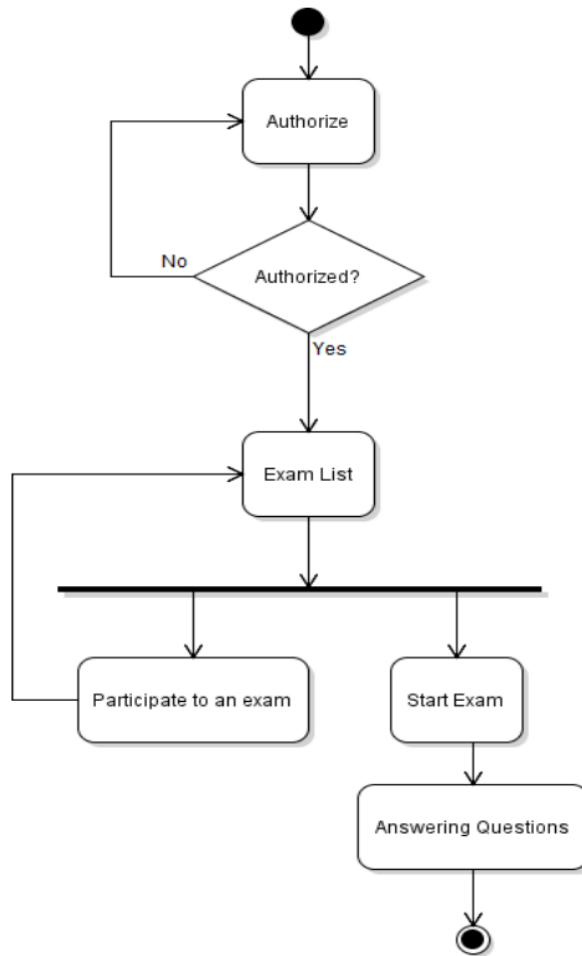

Figure 1. Activity Diagram (Teacher)

2.2. Activity Diagram (Student)

Same as teachers, figure 2 shows that students need an authorization before accessing exam list page. In students' exam list, there are two actions which can be performed by students those are: (1) start exam and (2) participate in an exam. Students registered as candidates can directly start an exam where unregistered ones have to participate in an exam before starting it.
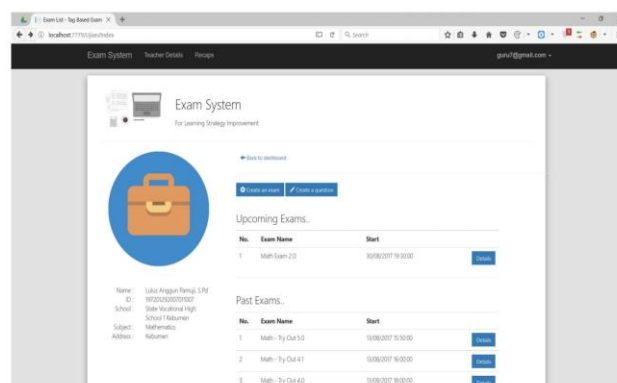
Figure 2. Activity Digram (Student)

2.3. Implementation

The following list shows figures of the system interfaces with implemented features:

a.   Exam List (Teacher)

Figure 3. Exam List (Teacher)

Figure 3 contains two lists of exams made by logged in teacher those are upcoming and past exams. Teachers can create an exam by filling a form shown after "Create an exam" button was clicked. Once the teacher has done filled the form, the system automatically adds an item to upcoming exams list according to the entered data.
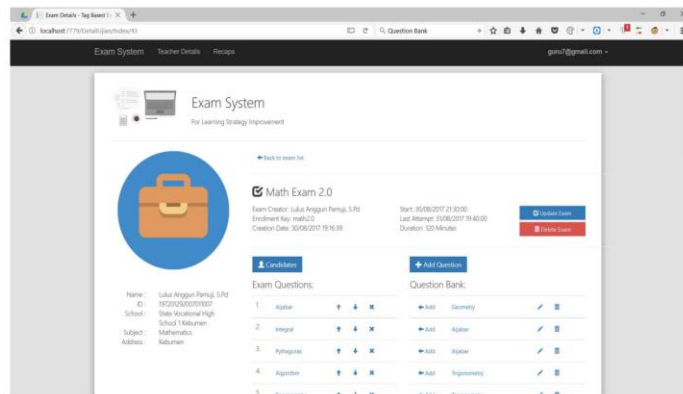
b.   Exam Detail

Figure 4. Exam Detail

Figure 4 contains data needed for the exam. Teachers can pick questions from Question Bank and include them into exam questions list. Teachers can also set up a list of candidates to the exam by entering both student or group name on a page shown after "Candidates" button was pressed.
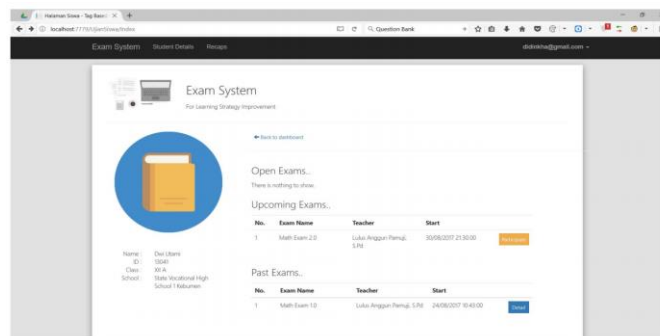
c.   Exam List (Students)

Figure 5. Exam List (Student)

Open, upcoming and past exams are included on Figure 5. Students registered as candidates in an exam can directly start the exam if its state is open. Students can participate in an exam even when they are not included in the list of candidates by entering enrolment key after clicking "Participate" button. Once the student enrolled, the "Participate" button will change to "Start Exam" if the exam state is open and "Detail" if isn't. If the student has taken an exam from these lists, the exam will be hidden.
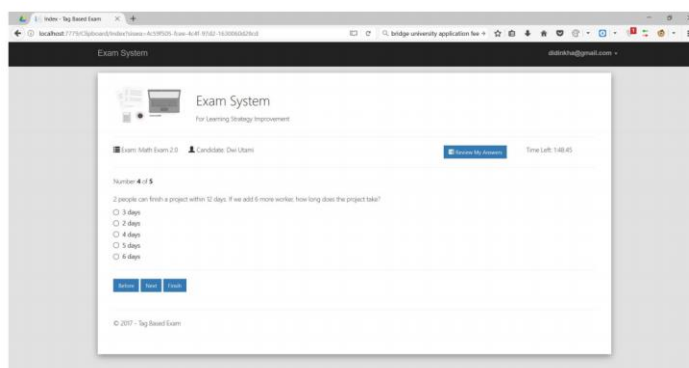
d.   Clipboard

Figure 6. Clipboard

```
$.ajax({
    url: "/Soal/Insert",
    data: JSON.stringify(events),
    type: "POST",
    contentType: "application/json;charset=utf-8",
    dataType: "json",
    success: function (result) {
        if (result == 1) {
            @Html.Raw(redirect)
        }
    },
    error: function (errormessage) {
        alert(errormessage.responseText);
    }
});
```

Figure 7. Snippet Code of AJAX

Figure 6 acts like a real clipboard, it holds answer sheets on top of it. In this page, students answer the questions given by teachers in Exam Detail page. Students can check their answer sheet for unanswered questions by clicking "Review My Answers" button. Numbers with a green background are answered and the ones with a white background aren't. The "Prev" and "Next" buttons execute functions which fire AJAX to get information from controller class. Figure 7 show snippet code of AJAX. This approach is to prevent the system from performing unneeded page reloads. Same goes to numbers in "Review My Answer" popup.

3. Result

After the features are implemented to the system, we want to make sure if the system runs well in some conditions. Unit testing [7] is the method we use to achieve this objective by testing every function which is included in our system especially the parts where user interacts with data. Figure 8 shows the result of the process.

Visual Studio [8] has built-in Unit Test which allows us to check the tested functions of our system. This feature marks a test case with cross mark in red if the case is failed, exclamation mark in yellow if the case is skipped and check mark in green if the case is passed. As shown in the figure, there are 18 passed test cases in 5 different classes.

Figure 8. Unit Test Summary

Visual Studio [8] has built-in Unit Test which allows us to check the tested functions of our system. This feature marks a test case with cross mark in red if the case is failed, exclamation mark in yellow if the case is skipped and check mark in green if the case is passed. As shown in the figure, there are 18 passed test cases in 5 different classes.
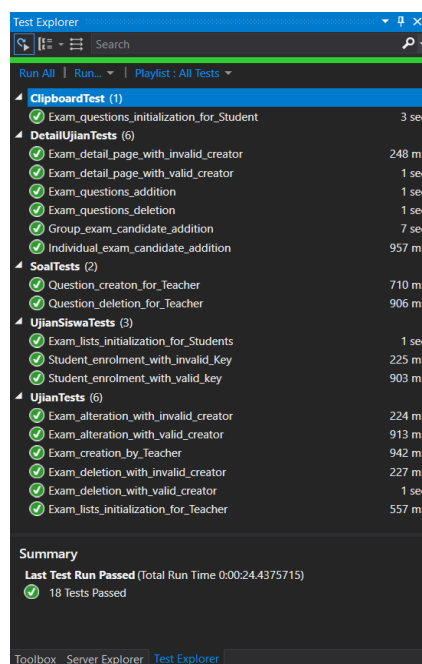
4. Conclusion

According to the research, we can pull some conclusions those are:
a. This research has been finished and the system can be used in school.
b. In the system, teachers acan acategrize qestions by its topics.
c. Students can now take their exams without worrying about their sheet being pirated or sabotaged

References

[1]   J. Galloway, P. Haack, B. Wilson, K. S. Allen, and S. Hanselman, "Professional ASP.NET MVC 4," in *Professional ASP.NET MVC 4*, 2012.
[2]   J. Kluge, F. Kargl, and M. Weber, "The effects of the ajax technology on web application usability," *Webist 2007 - 3rd Int. Conf. Web Inf. Syst. Technol. Proc.*, vol. WIA, no. May, pp. 289–294, 2007.
[3]   Y. P. Jin, Y. Liu, and G. Song, "Design and research of teaching resources management based on SSH and AJAX," *Proc. - 8th Int. Conf. Internet Comput. Sci. Eng. ICICSE 2015*, pp. 279–285, 2016.
[4]   A. Shaw, "Teaching students to design and implement social networks using MVC as a capstone experience," *Proc. 2013 10th Int. Conf. Inf. Technol. New Gener. ITNG 2013*, pp. 747–748, 2013.
[5]   J. Ou, Yun-Pei, X. B. Bai, and Q. X. Wu, "Interactive platform of project-teaching based on AJAX," *2011 IEEE 3rd Int. Conf. Commun. Softw. Networks, ICCSN 2011*, pp. 352–354, 2011.
[6]   M. Allahyari and K. Kochut, "Semantic Tagging Using Topic Models Exploiting Wikipedia Category Network," *Proc. - 2016 IEEE 10th Int. Conf. Semant. Comput. ICSC 2016*, pp. 63–70, 2016.
[7]   P. Runeson, "A survey of unit testing practices," *IEEE Softw.*, vol. 23, no. 4, pp. 22–29, 2006.
[8]   Microsoft, "Testing Tools." [Online]. Available: https://www.visualstudio.com/vs/testing-tools/. [Accessed: 12-Jul-2017].