International Journal
of Applied Information Technology
http://journals.telkomuniversity.ac.id/ijait/

Software Architecture of E-assessment on Higher Education

Wardani Muhamad [a,*], Wawa Wikusna [b]

[a, b] *School of Applied Science, Telkom University, Bandung, Indonesia*

A R T I C L E I N F O

Received 07 September 2017
Revised 21 September 2017
Accepted 30 September 2017
Available online 07 December 2017

Keywords
software architecture, e-assessment, ADL, software development, learning process

A B S T R A C T

Computer technology has been used to support the learning process at the university. Learning process, generally involved students and teachers in order to learn about materials on subject courses and also evaluate student competencies regularly. Teachers can evaluate student competencies or knowledge by e-assessment. E-assessment is one of the domains of e-learning which involves the use computer in assessment, includes: setting, delivery, marking and reporting of assessments. The Major benefit of the e - assessment system is its flexibility in term of global access and devices used to access. When developing an e-assessment system, we have two focuses on multi-dimensional approach, such as user friendly and student centric nature. Because of its complexity, software architecture need to define so software developer will develop software properly. By designing software architecture, view of the system that includes the system components, the behavior of those components, and the ways the components interact could clearly define. Architecture Description Language (ADL) has been used to describe software, because it provides a concrete syntax and formal framework for characterizing architectures. As the result, the design of e-assessment system architecture can meet the needs of attribute quality. The use of notation to explain ADL is able to provide a complete description than simply explaining ADL is using text. Furthermore, the e-assessment system architecture design is expected to be used as a reference for software development in establishing an e-assessment system.

* Corresponding author at:
School of Applied Science, Telkom University,
Jl. Telekomunikasi No. 1, Terusan Buah Batu, Bandung, 40257
Indonesia.
E-mail address: wardani.muhamad@tass.telkomuniversity.ac.id

ORCID ID:
- First Author: 0000-0002-6420-1683
- Second Author: 0000-0001-7733-5376

https://doi.org/10.25124/ijait.v1i02.1030
Paper_reg_number IJAIT000010213 2017

1. Introduction

Nowadays, computers technology cannot be separated from daily activities in a university, especially to support the learning process. Learning process generally involved students and teachers in order to learn about materials on subject courses and computers technology have a contribution to share the subject course material easier. For further, discussing teaching and learning process could not be separated from assessment activities. The assessment used to evaluate the student competencies achievement regularly. In general cases, it also used for grading, to measure student ability to fulfill the subject learning objectives. In part of detailing learning objectives, teacher defined some learning outcome that will be tested when assessment. In other words, each assessment component is prepared with the intention of addressing a particular learning outcome and area or related areas of knowledge in the particular subject. Preparing assessments traditionally, based on paper and pen, are time-consuming and bind resource. To relieve teacher task on preparing an assessment, along with information and communication technology (ICT) growth that promising ease of use, consistency, and resource efficiency, there is a technology to organize assessment called e-assessment.

Joint Information Systems Committee (JISC) define e-assessment as an electronic assessment used ICT to support its process, include: presenting assessment activity and recording learner responses [1]. Web-based assessment tools commonly found as primary computer involved in e-assessment. By this tools, a teacher can be disburdened of manual effort. Along with the development of assessment concept, which is recently the main objective of assessment has shifted away from content-based evaluation to outcome-based evaluation, e-assessment also has to adopt in this situation.

As a computer application or software, developing e-assessment need an excellent strategy. The initial phase is usually done is planning a software architecture, where is, for the general case it started after requirement specification is completed and just before beginning design phase [2]. Software architecture also a step in Software Development Life Cycle which its aim is how to solve customer's problems by defining the solution that contains all functional and non-functional requirement and match costumer's needs [3]. By this viewpoint, because of the primary users are student and teacher, when designing a software architecture of e-assessment, we need to give concern on multi-dimensional approach will be faced, such user-friendly and student-centric nature. To make the requirement clearer, we assumed that the natural condition of traditional assessment could adapt to be a model of student nature. This paper aim is to propose e-assessment software architecture design that can be used as the reference for a software programmer to develop e-assessment for higher education or university.

This paper is organized as follows. Section two presents the related work or fundamental concept used to support the research. Then section three shows methodology used to design e-assessment software architecture. Section four shows the result or proposed the software architecture design. Finally, section five summarizes the conclusion and future work.

2. Related Work

This section describes related research that has been done and also fundamental concepts used to design software architecture of e-assessment, these are assessment and e-assessment, software architecture, client-server architecture, and architecture description language (ADL).

2.1. Assessment and e-Assessment

Normally, when organizing an assessment, it begins with the assessment component preparation and ends with the collation of assessment components to distribute the results [4]. These detail activities as follows.

- **Assessment preparation**. As the first activity when organizing assessment, this activity includes defining assessment component, setting scoring rubric, evaluating subject materials have been delivered, assessment scheduling, setting guidelines for task completion and submission, setting marking and evaluation criteria
- **Assessment execution**. Assessment is held on a predetermined schedule. Teacher distributes cases must be completed and student completed it according to the guidelines. Generally, if the assessment is carried out in writing, the student creates a solution for each case provided on the answer sheet provided.
- **Monitoring**. Assessment completion should be monitored including tracking submission. When it is submitted, identifying plagiarism is held manually or using plagiarism detection software.
- **Evaluation and marking**. The teacher gives a score based on submitted answer referring to the scoring rubric and assessment component. Further, teacher producing feedback based on student achievement.
- **Re-assessment and second marking**. When student achievement is below standard or it does not fulfill several competencies, the teacher can assign a re-assessment in accordance with the unfulfilled competence.
- **Collation and storage of assessment results**. Assessment components and results collecting and exporting in various media like web blog or bulletin board for result dissemination.
- **Post-assessment evaluation**. Teacher produces an academic report based on the comparison of results for completed assessments. This feedback will be used as the improvement notes for next semester learning process.

Support of computer technology on organizing an assessment gives positive impact especially on the ease it offers. Organizing assessment by involving computer technology become a basic terminology of e-assessment. There are some general procedures involved in taking an e-assessment [5], that are:

1. **Access (P1)**
   The user has access to the e-assessment system via a web link or by clicking an icon which corresponding to a desktop application.
2. **Authentication (P2)**
   Each user has their own username and password recorded in a database. By this way, when they log in the e-assessment system they will face specific features appropriate to a user group.
3. **Presentation the assessment activity (P3)**
   Based on a predefined schedule, system present questions or cases have to answer by the student and also submission guidelines or instruction. One page could be consisting about one question or several questions and also e-assessment has an ability to random the question orders.

4. **Answering the questions (P4)**
   Student gives answers to presented questions in accordance with the question type. For example, if the question type is multiple choice, the student has to choose the most appropriate answer from a set of answer choices provided. e-Assessment also completed with the time available to complete the assessment.
5. **Recording of responses (P5)**
   The system has the ability to record all the answers chosen or filled by students. Furthermore, the system should provide a guarantee from recording failure.
6. **Marking the responses (P6)**
   Several types of questions could automatically mark by the system, meanwhile others type has to mark manually by the teacher.
7. **Presentation of feedback (P7)**
   The system displays the feedback filled by the teacher as the response of student achievement on assessment. The feedback can also in form of marks gained based on correct answers.

2.2. Software Architecture

Bass et.al [6] defines the software architecture of a system is the collection of structures required to explain about the system, which cover software elements, their relationship, and properties of both. Philip et.al [7] gives another definition of a software architecture which is a view of the system that contains system's principal components, their behavior as seen in the whole of the system and describe interaction and coordination among components to achieve all of the system's goal.

Some researchers [2], [6], [8] proposed the key principles of software architecture, these are:

- A software system architecture should only clarify how customer requirements achieved, not in order to describe the detail of implementation
- Flexibility on future changes and maintenance requirements should be anticipated when designing software architecture
- All customer functional and non-functional requirements should capture precisely to make sure that software architecture guarantee future product quality developed based on all those requirements
- Software system architecture have to predict all of the possible future risks when developing software system
- Software system architecture should promote optimize utilization of system resources

There are several architectural patterns [6], namely: layered, share-data or repository, client-server, multi-tier, and competence center and platform.

2.3. Client-Server Architecture

Kekic et.al [9] were received US patent on client-server computer network management architecture. Currently, most of the application which uses the internet to access it is based on client-server architecture. This architecture divides the entire system into two essential parts, the first part is the server and other called as the client as shown in Figure 1.

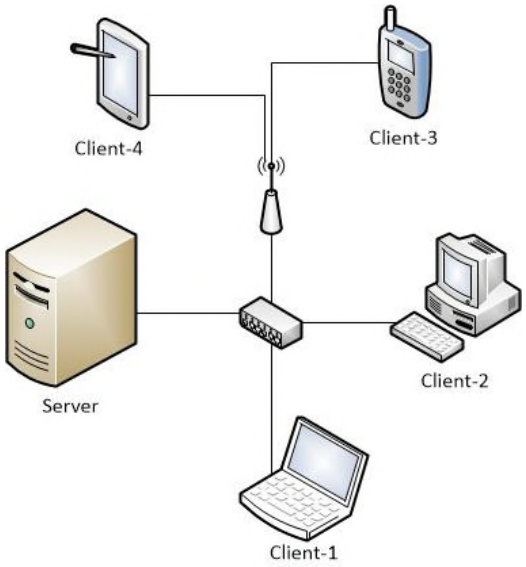

Figure 1 Client-Server Architecture

Web applications adopt client-server as bases architecture [2]. When we pay attention more detail on Figure 1, there are mainly three components that compose the architecture, namely: client, server, and medium of communication.

1. Client:
   Client act as service or information requester to the server, i.e.: web browser, email client, online chat client, etc.
2. Server.
   Server accepts requests from client then performs the processing to response it to produce a required information. In other words, server plays a role as service provider to generates the solution as fit as client's demand and send that solution to the client. The example of the server such as web server, email server, database server, etc.
3. Communication media between client and server.
   When the client interacts with the server, it needs a media to communicate (send request or response request). This communication facilitated by networking interface. The line that connected client and server is representing communication media which is bidirectional. Kind of communication media is i.e. internet, intranet, wireless network, etc.

Here is some common example of a system using the client-server architecture [6]:

- Information system accessed on local computer network where the clients are desktop applications (GUI based applications) and the server is a database management system
- Web-based application where accessed via web browser as a client and the servers are components running on specific purposes

2.4. Architecture Description Language

By architecture, many aspects of the system could describe clearer. Description of components that construct a system, behavioral specifications among components, and the pattern also mechanism on their interactions. Clemens [10] defines that architecture description language (ADL) as a formal language that explains or representing software system architecture. It has been developed to

avoid the weakness of specification methods on informal architecture. ADL use to representing software design at the architecture level in a formal way. To describe an architectural abstraction, ADL uses a descriptive notation. Architectural elements need to specified to support a mechanism for decomposing a system into components and connectors.

Several types of architectural elements to detailing ADL are [11]:

- User interface
    - GUI, that is kind of a traditional GUI client using Java Swing, C# or C++ as the programming language.
    - WebUI, that is application or user interface implemented in form of a set of web page use supporting languages like CGI scripts, PHP, or a Java web app
    - Command Line, that is a user interface implemented as a command line program, such as a Python script.
- Servers
    - Message Driven Server, when a server is operated if a message receipt from the message bus
    - Server, when a server is operated is driven by a mechanism other than messages like RPCs or temporal schedules is received
    - Batch Program is a program that is executed from a scheduler and processed in a single execution, automatically without any input from user or system element.
    - Data Loader is a program that transmitting any information to a destination as a result of data extraction from a source
- Data Stores
    - System database is supported database or a set of tables for a software system
    - File, that is a file on the file system

When connecting each component in the system, there is some fair connector generally use:

1. RPC is a synchronous inter-process procedure call (usually XML over HTTP)
2. Database Data Flow is data writing process to a database so another element could use it
3. File Data Flow is data writing process to a file system and allow another element use it
4. System Messaging, receive and release of messages through a messaging destination over the system bus

If comparing between ADL and modeling language, ADL concentration is a statement of components. A main advantage of ADL is its ability to facilitate formal analysis and verification of complex software architecture [12]

3. Research Methodology

The methodology used to design e-assessment software architecture is adopted from Attribute-Design Driven (ADD) version 2.0 [13] proposed by Wojcik et.al. The main characteristic of ADD is its design process when defining software

architecture based on the quality attribute requirements. ADD divided all step into three main stages, namely: input, process, and output.

Inputs in ADD are functional requirements, design constraints, and quality attribute requirement. Those all inputs item are strongly related to business and mission goals of system stakeholders prioritized. Process in ADD representing supporting activities needs to design a software architecture include defining artifact needed to describe the architecture such as ADL notation. Meanwhile, output in ADD is a system design that describes software architecture terminology includes the roles, responsibilities, properties, and relationships among software elements.

3.1. Functional Requirement

To define functional requirements, need to design e-assessment system, we have to pay attention to the activities on organizing assessment and general procedures of e-assessment described in section two.

Here are functional requirements for an e-assessment system:

1. **User management (F1)**
   The e-assessment system involves at least three groups of user, they are the student, teacher, and administrative staff. Administrative staff should manage student and lecturer data periodically that normally done every semester. The main purpose of data management is adapting the change of subject taught by the lecturer and taken by the students. Based on these data, the e-assessment system should ensure that the lecturers organize the e-assessment according to corresponding courses, as well as ensuring students carry out the assessment in accordance with the courses they take.
2. **User authentication (F2)**
   As a system that involves more than one user, the e-assessment system should ensure that users who can access the system are registered, as well as ensuring the conformity of the features presented with users logged into the system. To make it easier for users when having problems at login, the system should also facilitate users to manage their accounts even if they forget their passwords.
3. **Learning outcome management (F3)**
   The main purpose of the assessment is to measure the achievement of student competence during the learning process. At the beginning of the learning process, the lecturer will determine the learning outcomes relevant to the competencies that the student will achieve at the end of the semester. Competence or learning outcomes are always evolving as the industry needs for the competence of graduates. Therefore, the e-assessment system should be able to manage the mapping and determination of learning objectives by the lecturer.
4. **Question management (F4)**
   How to create a tool to measure student achievement? The answer is only one, which is to provide questions or cases that must be completed by the student during the assessment. These questions must necessarily have a very strong correlation to a predetermined list of learning objectives. Considering the number of students who must be tested, the lecturer must also increase their creativity by providing a sufficient number of questions. One indicator of adequacy that can be used as a reference is the availability of several packages of questions given to students different from one another. The e-Assessment system should be able to facilitate and facilitate

lecturers in preparing packages of questions to be presented during the assessment.

5. **Assessment scheduling (F5)**
   The most significant moment in the assessment is when will the assessment be done? The e-assessment system should facilitate lecturers in scheduling assessment and automatically inform students. When an assessment needs to be completed in more than one day, the e-assessment system should at least be able to facilitate to determine the submission deadline. In this way, the collection of answers or implementation of the assessment becomes orderly in accordance with the time specified. To improve the capability of the e-assessment system, a predetermined schedule becomes a constraint on the features for working on a package of questions or solutions submission accessible by students.
6. **Presenting assessment activity (F6)**
   In accordance with a predetermined schedule, the student will conduct an assessment or submit a solution. The e-assessment system becomes the main interface used by students to meet all of these. The e-assessment system should be able to present packets of pre-made questions according to its type. For example, if the question type is a multiple choice, then the e-assessment system should present a selection of answers that students can choose. If the questions presented should be answered by uploading a file or script, then the e-assessment system should be able to facilitate the user to access the resources on the client computer in finding the appropriate file or running a certain software that is allowed. As a complement, the e-assessment system should be able to show the remaining time available to complete the package of questions to be solved. Ease of use becomes a significant indicator so that users will not have difficulty in using this feature.
7. **Scoring (F7)**
   Deciding the achievement of competence (pass) cannot be done in vain. One component that can be used objectively to determine whether or not a student passes is a score. The e-assessment system should be equipped with a scoring feature to make it easier for the lecturer to determine whether or not the student achieves the learning competency. In general, the scoring mechanism of the e-assessment system can be divided into two groups, manual and automatic. For the type of multiple-choice questions or similar variants, we can apply the scoring automatically. When creating a question, at the same time we can determine the exact answer of the alternative answers given, as well as determine the weight of scores obtained by students if answered correctly. Meanwhile, for the type of essay requires manual scoring by lecturers. The e-assessment system should be able to present the answers that have been filled by the student and submit the scoring done by the lecturer. Compilation scores are then informed to the students.
8. **Feedback (F8)**
   Are all students always able to achieve learning competencies in one assessment opportunity? Probably not. When the learning process is intended to equip students so as to be able to meet certain learning competencies, assessment should be used as a primary tool in measuring achievement. What if some students cannot meet the learning competencies? Lecturers should provide reassessment opportunities and submit feedback to students related to learning competencies that have not

been achieved. The e-assessment system should be able to facilitate lecturers in providing reassessment and deliver feedback to students.

3.2. Design Constraint

Here are some constraints defined for an e-assessment system software architecture design:

- The user interface used for teacher and administrative staff is webUI. Its choose because when managing various data via a form, the easiest technology to adopt is the web.
- GUI used as the user interface for students. Programming languages that support to develop a GUI application like C# have an ability to manage the computer client resource (such as restrictions on the use of hotkeys or turning off some features on keyboard) and make any limitations so students only run software permitted (if required).
- The data storage mechanism can be a problem if not properly managed. The common technology that is still widely used is RDBMS. Choosing a popular RDBMS like MySQL, Oracle, or MS SQL Server can be the perfect solution to help store and process data.

3.3. Quality Attribute Requirement

To ensure the resulting software architecture meets the quality, the objective reference used is the general procedure of e-assessment as described in Chapter 2.

3.4. ADL Notation

Notation determination is intended to facilitate the reader's understanding of the resulting software architecture. Usually, the notation in graphical form is easier to understand than textual. Figure 2 [11] shows the notation used to construct software architecture design.

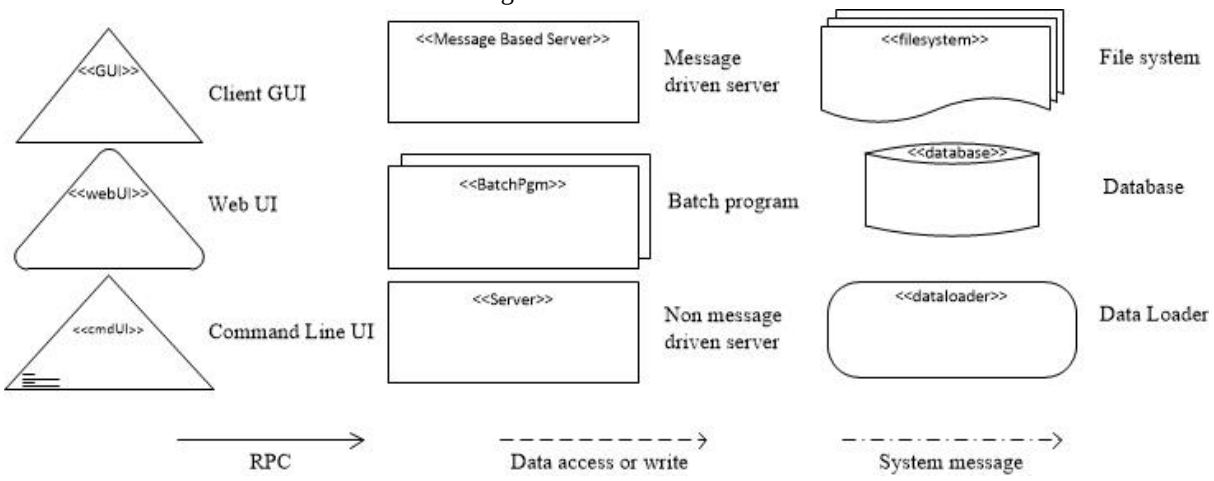

Figure 2 ADL Notation

The graphical notation is used to represent elements and connectors as the basic elements of software architecture. To facilitate the making of notation used basic stencils on Microsoft Visio and to clarify the name of each notation added stereotype, copied from UML. Every type of element (server, user interface, data store) has its own base shape and every subtype denoted using the variation (dimension, line, or texture) of each base shape. Meanwhile to denote the connectors are used lines, which are commonly used as connecting symbols.

A triangle was used as a base shape to represent user interface. The triangles were then modified slightly to denote each sub-type of an interface by modifying each corner of the triangle or adding specific information where is traditional GUI client having a sharp corner, web page user interface (webUI) having rounded corners, and command line program having a graphical represent a command line inside the triangle. Rectangles used to represent a basic shape of server elements. The addition of stereotypes and line modifications to the rectangle are used to confirm each subtype of the server elements. Drum and its variants used to represent data store elements. This notation commonly used in general cases so it's easy to understand.

3.5. Output

The output is a software architecture design that describes the user roles, responsibilities, and also software elements. The detailed output will discuss in section four.

4. Result

One of the popular learning management system software is Moodle [14]. As a learning support software, Moodle is equipped with an administrative module to manage users and their roles, as well as general modules that play a role in organizing learning content and conducting assessment in the form of quizzes, assignments, and workshops. Furthermore, Moodle can be operated in three different technologies: web technology, desktop, and mobile devices. Completeness of features and media access offered by Moodle certainly very helpful lecturers and students. However, one major drawback that Moodle does not have is its inability to conduct assessment in the form of practices that involve accessing resources on client computers.

After analyzing functional requirements, the weakness of Moodle, and elements of software architecture, we proposed a software architecture to support organizing an assessment using computer technology as shown in Figure 3.

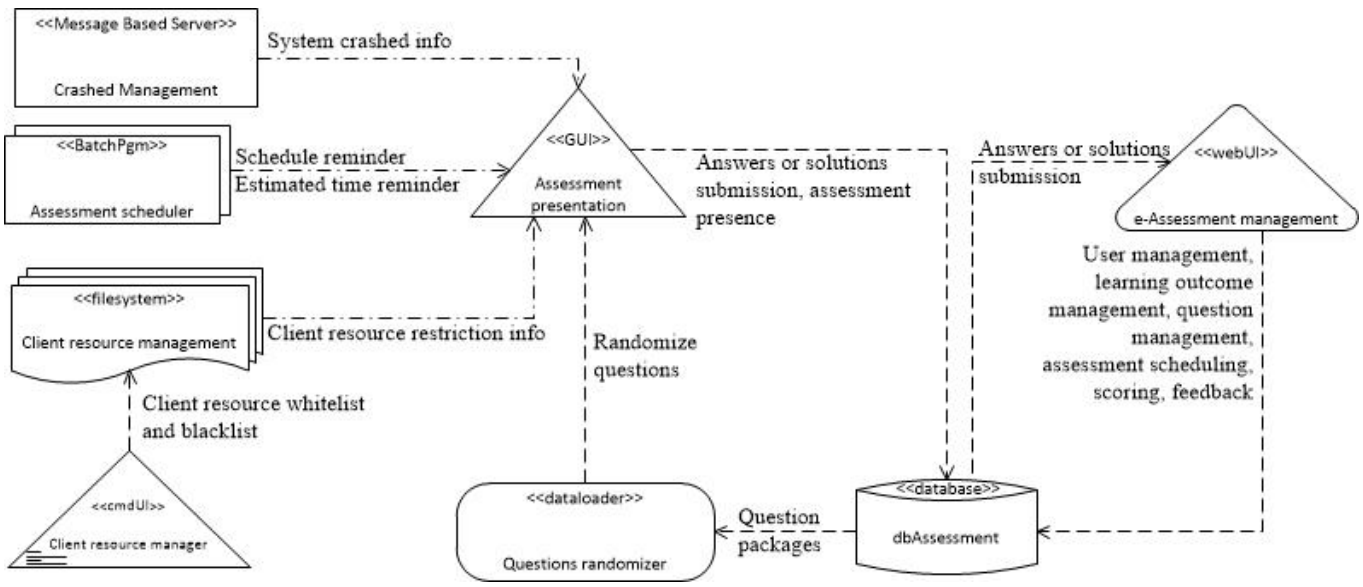

Figure 3 E-assessment Software Architecture Design

The adoption of client-server architecture patterns in the proposed e-assessment system architecture is noticeable through the involvement of user interface

elements: the database, GUI interface, and web interface. The database represents the role of a server in charge of receiving requests from the GUI and web interface and returning information in accordance with its request. Client-side is represented by GUI and web interface. In the client-server architecture pattern, the client becomes the part that interacts with the user.

In proposed software architecture does not present users explicitly when the user is an element of a system that contributes significantly, therefore it is necessary to explain again that the e-assessment system is accessed by students, lecturers, and administrative staff. Those users interact with the e-assessment system through the GUI and web interface, where students interact through the GUI interface, while the web interface is accessed by lecturers and administrative personnel.

The GUI interface plays a major role in presenting an assessment. Through this interface, students will be faced with a set of questions or cases that must be resolved in accordance with a predetermined schedule. This means that students send answers to all questions presented through this interface. To support the assessment presentation, the GUI interface is equipped with the ability to randomize questions, display remaining time, restrict client computer resources, and anticipate when a system crash occurs. Randomized questions can improve the quality of the assessment so as to prevent students from deception such as plagiarism or cheating other students' answers. The completion of the assessment must be time-limited so that the remaining time information is needed by the students. The e-assessment system should not only be equipped with the remaining time available but should also be equipped by automatic submission if the available time reaches zero. To increase the anticipation of cheating, the e-assessment system should also be able to limit access to the computer resources used to run the GUI interface. The restrictions are made to anticipate students running other software that is not allowed, for example, a web browser to access the internet, word processing application that allows the storage of subject material, etc. Furthermore, the anticipation of the crash can meet the reliability aspects of a system. The e-assessment system should be able to save the state when it crashes, so students still have the opportunity to complete the assessment without having to repeat from the beginning or get a different set of questions than before. The e-assessment system should also be able to retain the remaining time information available to complete the assessment in the event of a crash. No less important, the GUI interface can also be equipped with the ability to calculate scores automatically.

The web interface becomes an element accessed by faculty and administrative staff. This interface is served in organizing and preparing assessment instruments. The capabilities of e-assessment systems supported through the web interface include user management, management of learning competencies, question management, assessment scheduling, scoring, and feedback. Of all the abilities mentioned above, administrative staff plays a role in user management. Academic support data such as student data taking courses and lecturer data teaching course must be managed regularly because at normal each semester the data is changed. The lecturer runs the e-assessment system to use the remaining capabilities. Lecturers must manage the learning competencies periodically due to the possibility of curriculum change. The competencies become the main reference of the learning process and become an objective indicator that will be assessed its achievement through assessment. In addition, the creation of questions must also be aligned or can be mapped with learning competencies. In general, the questions type presented are multiple choice and essay. The e-assessment system is not only equipped with the ability to keep a list of questions to be presented in the assessment, but also provides a mapping of questions with learning competencies.

If there is a learning competency that does not have a question, then the system must be able to provide alerts and learning competencies cannot be used for assessment. Assessment should be done on schedule; therefore, the e-assessment system should be equipped with schedule management capability. When scheduling an assessment, the lecturer must determine the learning competencies to be tested as well as the estimation of time provided to complete the assessment. After completing the assessment, answers or solutions that have been given by the students will be given a score. If the e-assessment system has been completed with automatic scoring, then the lecturer does not need to give a score manually.

Finally, to complete an assessment process, lecturers should provide feedback in accordance with the achievement of learning competencies to students including determining the passing of a course.

To store data used the database and file system elements. Database elements serve to store all data involved in all assessment organizing transactions, while the file system is used to store whitelists and blacklists associated with client computer resource management.

Finally, to ensure the adequacy of software architecture design according to the quality requirement attribute, mapped functionality compliance to general e-assessment procedures as presented in Table 1.

Table 1 Functionality Compliance to E-assessment General Procedures

	P1	P2	P3	P4	P5	P6	P7
F1	√						
F2	√	√					
F3	√		√				
F4	√		√				
F5	√		√				
F6	√		√	√	√		
F7	√					√	
F8	√						√

5. Conclusion

Based on the mapping of functionality with general e-assessment procedure, it can be concluded that the design of e-assessment system architecture can meet the needs of attribute quality. A functional can fulfill more than one general procedure, and vice versa. The use of notation to explain ADL is able to provide a complete description than simply explaining ADL using text. Furthermore, the e-assessment system architecture design is expected to be used as a reference for software developers in establishing an e-assessment system. As a further work, this design needs to be evaluated and improved so as to involve RPC connectors so that the ability of web and GUI interface elements can involve XML technology.

Bibliography

[1] Joint Information Systems Committee (JISC), “Effective Practice with e-Assessment,” Information Systems Journal, 2007. [Online]. Available: http://www.jisc.ac.uk/publications. [Accessed: 14-Aug-2017].

[2] A. Sharma, M. Kumar, and S. Agarwal, “A Complete Survey on Software Architectural Styles and Patterns,” Procedia Comput. Sci., vol. 70, pp. 16–28, 2015.

[3] J. Bosch, “Software Architecture : The Next Step,” Lect. Notes Comput. Sci., vol. 3047, pp. 194–199, 2004.

[4] L. S. Prakash and D. K. Saini, “E-assessment for e-learning,” AICERA 2012 - Annu. Int. Conf. Emerg. Res. Areas Innov. Pract. Futur. Trends, 2012.

[5] T. Dube, M. Ma, and Z. Zhao, “Tasks, processes, and tools: A Design Methodology Management approach to design and development of e-assessment,” Emerg. eLearning Technol. Appl. (ICETA), 2011 9th Int. Conf., pp. 211–216, 2011.

[6] L. Bass, P. Clements, and R. Kazman, Software Architecture in Practice (SEI Series in Software Engineering), 3rd Editio. New Jersey: Pearson Education, Inc., 2012.

[7] A. Philip, B. Afolabi, O. Adeniran, G. Ishaya, and O. Oluwatolani, “Software Architecture and Methodology as a Tool for Efficient Software Engineering Process: A Critical Appraisal,” J. Softw. Eng. Appl., vol. 3, no. 10, p. 934, 2010.

[8] Å. Lindström, P. Johnson, E. Johansson, M. Ekstedt, and M. Simonsson, “A survey on CIO concerns-do enterprise architecture frameworks support them?,” Inf. Syst. Front., vol. 8, no. 2, pp. 81–90, 2006.

[9] M. M. Kekic, G. N. Lu, and E. H. Carlton, “Client-server computer network management architecture,” U.S. Patent No. US6664978 B1, 2003.

[10] P. C. Clements, “A survey of architecture description languages,” Proc. 8th Int. Work. Softw. Specif. Des., no. March, pp. 16–25, 1996.

[11] E. Woods and R. Bashroush, “Using an architecture description language to model a large-scale information system - An industrial experience report,” Proc. 2012 Jt. Work. Conf. Softw. Archit. 6th Eur. Conf. Softw. Archit. WICSA/ECSA 2012, pp. 239–243, 2012.

[12] A. Choutri, F. Belala, and K. Barkaoui, “A Tile Logic Based Approach for Software Architecture Description Analysis,” J. Softw. Eng. Appl., vol. 3, no. 11, pp. 1067–1079, 2010.

[13] R. Wojcik et al.s, “Attribute-Driven Design (ADD), Version 2 . 0,” 2006.

[14] Moodle, www.moodle.org