ISSN 2347 - 3983
Volume 8. No. 4, April 2020
International Journal of Emerging Trends in Engineering Research
Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter55842020.pdf
https://doi.org/10.30534/ijeter/2020/55842020

Improving Moodle Architecture and Learning Features in Cloud Server Ecosystem Using Kubernetes and Gamification

Yesun Utomo[1], Gede Putra Kusuma[2], Evan Kristia Wigati[3]
[1]Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11480, yesun.utomo@binus.ac.id
[2]Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11480, inegara@binus.edu
[3]Computer Science Department, BINUS Graduate Program - Master of Computer Science, Bina Nusantara University, Jakarta, Indonesia, 11480, evan.wigati@binus.ac.id

ABSTRACT

LMS has been used by many universities and schools to improve learning efficiency and effectiveness. One of the many LMS is Moodle. This LMS has many learning features and can be extended by installing "plugins". However, Moodle is monolith, which means all its resources like front-end, files, and database are installed in the same server and requires high cost server to do so. This study presents Moodle LMS architecture hosted in Google Cloud Platform to solve this problem. The core LMS learning features are extended by adding location-based learning and object-recognition learning as gamification element. Both features are implemented in the form of Moodle plugins that utilizes external service via API. Location-based learning utilizes Leaflet interactive maps and Google Maps place autocomplete search. While object-recognition learning utilizes an object recognition service (ORS) hosted in Google Kubernetes Engine to classify, store, and create image models. Evaluation of the architecture uses the Architecture Trade-off Analysis Method (ATAM) to evaluate performance and scalability as quality attributes. Experimental results showed that Moodle core LMS could support 480 concurrent users with average response time of 11.37 seconds with 100% availability. While ORS could support 960 concurrent users with availability above 98%. Thus, educators could use this architecture as an example to support many students. In the future, this solution may be evaluated if it could increase student motivation and achievement.

Key words : E-Learning, Gamification, Location-based Learning, Object Recognition, Cloud Deployment.

1. INTRODUCTION

Education is part of our life that must be taken since we are small until we become young adult. Students receives education from teachers in schools and conduct self-learning to discover something new and then passes that knowledge to the next generation. However, students often feel bored in class and lose motivation that causes in-effective learning. Educators and students must speak the same language for an effective learning to take place [1].

One possible solution for this matter with students especially millennials is using the concept of gamification [1]. Gamification means adding game elements such as levels, reward systems, scores, or other game elements into non-game context such as education [2]. The purpose of gamification in education is to increase motivation in learning. It is because when one does anything that he/she likes, one will do his/her best to get better result. Gamification proved to be effective motivator as proposed by [3]-[4].

Gamification could be implemented along-side other learning method such as Learning Management System (LMS). Since personal computers or PC spreads, they have been used to aid education process by giving basic instructions. Today it is called LMS and requires internet to use. LMS provides not only instructions but enables teachers to track student progress and manage learning courses. [5] suggests that almost every universities seems compelled to have an LMS and it will keep spreading. LMS could increase the efficiency of teaching and is a growing technology expectation [6]. LMS could benefit students nowadays because they grew up using internets, computers, and smartphones [7].

Combined work of gamification LMS have been done and proved effective. Many of them implements gamification on Moodle. One example is done by [8] that uses gamification element similar to Role Playing Games (RPG) like player level, leveled up system by finishing tasks, progress bar to track progress, titles that evidenced the mastery of a skill or subject, and player versus player of crossword puzzle. The gamified Moodle was tested at a higher education institute for a semester and results in decreased dropout rate by 10%. Another similar test was conducted by [9] which uses gamification elements: player level, level up system by completing tasks, progress bar to track progress, and leaderboard. Students engagement raised significantly after using the gamified Moodle compared to when before using it.

Both experiment results showed that using gamified LMS could increase student motivation and achievement compared to student that does not use gamified LMS. However, these gamified LMSs lacks gamification elements that connects students to the real-world environment. According to survey by [10], learning outdoor could improve learning achievement. Another problem is Moodle gamified LMS' resources are implemented in the same server which consumes more resources. Therefore, this study proposes the solution by using Moodle LMS and extend its features by utilizing external services to increase performance and scalability. Learning features used in this study are location-based learning and object-recognition learning which will be discussed in Chapter IV. Performance and availability tests are performed to ensure that both learning methods could be conducted without any technical issues using Architecture Trade-off Analysis Method (ATAM).

2. RELATED WORKS

2.1 MDA Framework

Developing games is quite complex because there are many components that needs to be kept in mind. To help analyzing previous works, MDA framework created by [11] will be used. MDA stands for Mechanic, Dynamic, and Aesthetics. Mechanic describes components of the game at data and algorithm level. Dynamics describes the behavior of mechanics during run-time and how it acts upon receiving player inputs. While aesthetics describes emotional responses of players when they are experiencing the game. Aesthetic divided into 8 categories:

- Sensation: players experiencing the game as sense-pleasure.
- Fantasy: players experiencing the game as alternate reality.
- Narrative: players experiencing the good story of the game.
- Challenge: players experiencing the game as fun problem-solving activity.
- Fellowship: players experiencing the game with fellow players inside the game.
- Discovery: players experiencing hidden features of the game.
- Expression: players experiencing the game as platform to express their personality.
- Submission: players playing the game to past time.

2.2 Related Works on Gamified LMS

Some researches choose to implement gamified LMS aside from Moodle LMS for various reasons. Reference [12] proposes a learning method using Technology-Enhanced Training Effectiveness Model (TETEM) combined with gamification. TETEM was chosen because it is developed for training in virtual worlds in organizational context.

Reference [13] proposed using gamified Microsoft SharePoint. The reason they use Microsoft SharePoint is because Moodle and Blackboard lack a built-in notification system to notifies students about assignments and notes that may pop up.

Reference [14] developed a customizable learning platform called OneUp. This platform allows teachers to determine which game elements to use for each course. OneUp allows instructors to insert static or dynamic problems which will be graded automatically. Static problems are multiple choice questions, true/false questions, and fill-in-the-blanks questions, and matching questions. Dynamic problems are problems could be solved by several different methods if those methods give the correct answer. Reference [15] implemented points and quiz on a web platform called Rain Classroom. This platform is used because it is simple to use. Reference [16] compared blended learning and blended learning + gamification on argumentative writing.

Researchers that uses Moodle LMS usually implements extra features on the LMS called "plugins". Reference [8] combined gamification elements with Moodle and tested it at a higher education institution. This platform was tested for a semester and decreased dropout rate by 10%. Reference [17] uses Moodle LMS combined with gamification on Computer Programming called "Batch and Stack". Moodle Log files were collected to analyze student performance. Analysis of Moodle log files suggest that Experimental Group have higher log entries than Control Group.

Another work that combines Moodle LMS with Gamification was done by [18]. Game elements used in this work are player level, experience points, leaderboard, Quizzes, item prizes, virtual currency, and player avatar. Aside from gamification, few extensions are integrated in LMS. The first extension is to allows LMS to include educational computer games or other gamification features. The second extension is to connect securely to external learning applications or tools. Another basic functionality of LTI includes: Single Sign On where learner must login only to LMS they are registered in. The third extension is xAPI which extends assessment and self-assessment functionalities by tracking activities in simulated environment or mobile devices.

Table 1 shows the overview of the findings on studies of gamified LMSs. Based on previous works, we can conclude that the most used game elements are points, leaderboards, player levels, and player avatar. Points and player levels are used to track student progress. Leaderboard are used to make students feel the sense of competition with each other. For players to express themselves, player avatars could help them. Enabling players to change their avatar with their own picture or other picture will make them gain interest of using the platform. Table 1 also shows that gamified LMSs do not have gamification elements to encourage outdoor activities which this study proposed.

2.3 Architecture Trade-offs Analysis Method (ATAM)

ATAM is a framework to asses' tradeoffs between quality attributes in a software architecture. Every software has quality attributes such as availability, performance, security, etc. Each quality affects each other, and trade-offs must be made for a software / system to work. ATAM helps stakeholders to make better design decisions to mitigate or solve problems before they become more expensive to fix as development progresses [19]. There are steps to conduct ATAMS, and they are:

- Collect scenarios
  System scenarios are collected to fulfill functionality and quality requirements. Another reason is to facilitate communication between stakeholders and to develop common vision in the development team.
- Collect requirements / constraint / environment
  This step is interchangeable with step one because requirements could occur before any scenarios are thought.
- Describe architectural views
  Based on information gathered in step 1 and 2, a few possible architecture designs are proposed. This architecture describes architectural elements like system process, data placement, etc. that have relevance to important quality attributes.
- Attribute-specific analyses
  Each possible architecture is analyzed whether it could fulfill determined requirements and scenarios by focusing on important quality attributes.
- Identify sensitivities
  Any modeled values like response speed of a server that significantly affected by a change in the architecture are considered sensitivity point.
- Identify trade-offs
  An example trade-offs point is as follows: the average respond time of an architecture is higher when more server is used to handle request. However, this means more cost is required to buy extra servers.

Scenario-based tests must be carried out to evaluate quality attributes and understand trade-offs between each attribute. Scenarios can be organized from specific scenario into a general scenario to show quality attributes that in concern of the software development process [20]. Once the scenarios have been presented, quality attributes and must be analyzed in isolation in any order.

Table 1: Related works of gamification

Previous Works	E-learning environment	Game Mechanics	Game Dynamics	Game Aesthetics
[9]	Moodle	Player Level, Points Leaderboard, Progress bar	Completing given task rewards players with points. Once certain point is collected, player level increases. Player levels displayed in a leaderboard.	Sensation, Challenge, Expression, Fellowship
[8]	Moodle	Player Avatar, Progress bar, Player Level, Points, Leaderboard, Instant Feedback, Battles,	Avatar progress with player levels. Points earned from submitting tasks could increase player level.	Sensation, Challenge, Expression, Discovery, Fellowship
[15]	Rain Classroom	Points, Quiz	Points are earned by answering quiz correctly and giving reasonable explanation. Points could be deducted by answering incorrectly or giving unreasonable explanation.	Sensation, Challenge, Expression, Discovery
[16]	Edmodo	Points, Leaderboard	Points are earned by posting arguments with the correct label and those points are showed on leaderboard.	Sensation, Challenge, Fellowship, Expression
[14]	OneUp	Points, Player Level, Leaderboard, Achievement, Player	Higher player level unlocks more Assignments. Each task will have direct feedback that tells whether the answer is	Sensation, Challenge, Expression, Discovery, Fellowship

		avatar, Virtual currency, Immediate feedback, Freedom to Fail	true or false.	
[17]	Moodle	Player Level, Player Avatar, Quiz, Points, Achievement	Finishing quiz and assignments could earn points and achievements.	Challenge, Fellowship
[12]	TETEM	Points, Leaderboard	Points are earned by giving correct answers or behaving properly as a team leader. Leaderboard will show each player's earned points.	Challenge, Expression, Fellowship
[13]	Microsoft SharePoint	Player level, Player Avatar, Leaderboard	Earned points will be shown on leaderboard. Top 3 players will have more decorated avatars than the rest of the players.	Sensation, Challenge, Expression, Discovery, Fellowship

3. PROPOSED MODELING

This study will use Moodle LMS as the base platform combined with gamification elements. Moodle architecture is monolith, which means that all its resources like front-end, back-end, database, and storage files are placed inside the same server. This means better hardware is required with higher cost and non-scalable. The overview of research steps is shown in Figure 1.

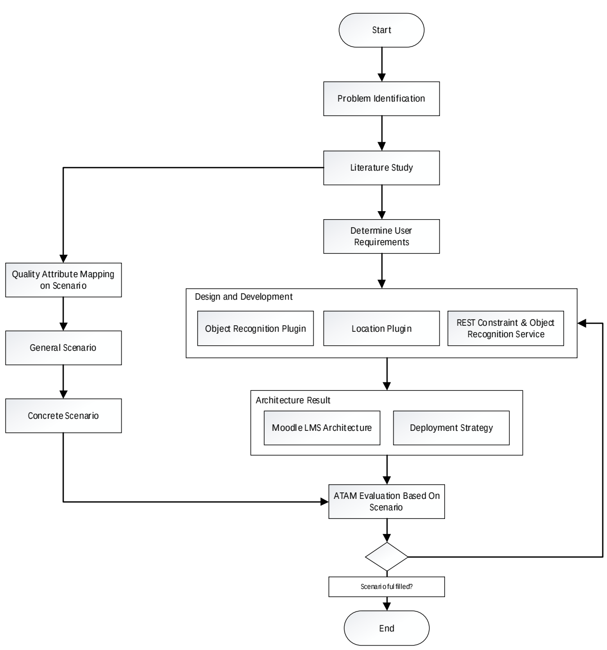

Figure 1: Research steps

The main challenge is how to implement location-based learning and object-recognition learning into Moodle as extended features especially translating raw PHP and JavaScript into Moodle framework. However, by getting familiar with the logic operators, relational operators, and other concepts, this challenge could be overcome [21]. Developing Moodle plugins is the best choice because tinkering with Moodle core codes could create more technical issues. This study seeks to extend Moodle LMS features using plugins which utilizes external service to make the system scalable, affordable, and has good performance. The Literature review was discussed in previous chapter.

The next step is to determine user requirements and what is the suitable plugin type to implement location-based learning and object-recognition learning based on those requirements. Moodle has many plugins types that ranges from assignment submission plugin, assignment feedback plugin, block plugin, etc. Each plugin has specific purposes to aid learning process and different development process.

After examining all possible plugin types based on user requirements, it is decided that location-based learning will use availability plugin type which restrict student from accessing an activity before fulfilling pre-requisite conditions and block plugin type which. While object-recognition learning will use assignment submission plugin type which requires student to submit images which will be automatically classified by the system, assignment feedback plugin type which display submitted image classification results to teachers and admins, and block plugin type which enables admins to upload image to create object recognition model which will be discussed further in the next chapter.

The initial architecture is as follows: Moodle LMS is implemented in Google Cloud Platform (GCP) and both gamification elements are implemented in form of Moodle plugins. The ATAM evaluation will be used to examine quality attributes of the LMS.

As mentioned before, both learning method is implemented as gamification element in the form of Moodle plugins that utilizes external services. Using external services has several benefits such as easy to maintain and grow [22]. Overview of the proposed architecture can be seen in Figure 2.

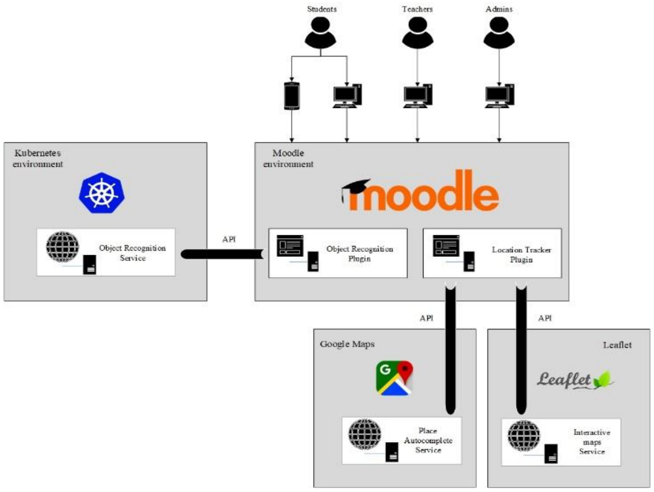

Figure 2: Proposed gamified Moodle LMS architecture

The following are proposed gamification elements:

- **Player levels** that allows students to keep track of their progress. Each level starts from level one, so every student will start at level one for Math, History, etc. To level up, students must earn experience points by finishing each task that comes with attached experience points (set by the teacher) and submitting the correct answer on time. This element comes from *block plugin*: Level Up!

- **Location tracker**. This element come in the form of *block plugin* called 'Geolocation Block' and *availability plugin* called 'Geolocation Availability' to support location-based learning. The *Availability plugin* could restrict tasks so that it can only be accessed once the student is within certain range of the task location as shown in Figure 3.

Figure 3: Geolocation Block restricting access to assignment

The *block plugin* can show the locations in a course by list or on an interactive map from Leaflet. 'Geolocation Block' is improved by adding Google Maps search feature to help pinpoint destination easier as shown in Figure 4. User location is stored in the current session and is used by the *availability plugin*.

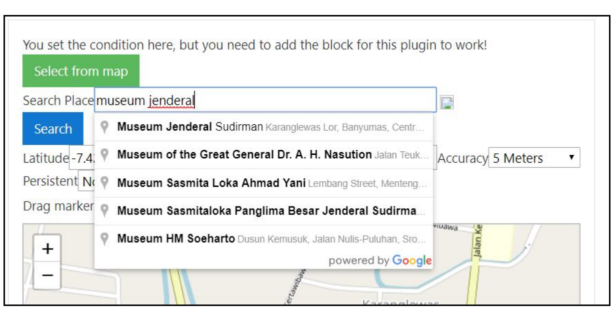

Figure 4: Geolocation search feature

- **Object-recognition**. The algorithm used for this element is called "Object Instance Recognition Using Best Increasing Subsequence" (BIS) proposed by [23]. The algorithm is compiled using C++17 in Windows. Before applying BIS, an image must be filtered before having its features taken and finally select all geometrically consistent pairs. The core of this element comes in the form of *assignment submission plugin* called 'Object Recognition'. During real practice, teachers will give assignment activity that utilizes this plugin as shown in Figure 5.

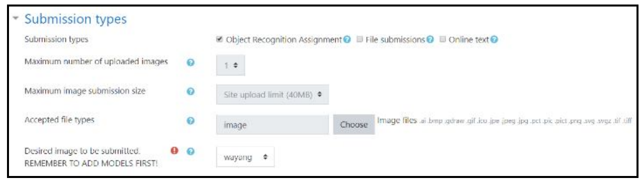

Figure 5: Utilizing object recognition plugin

The assignment requires students to upload a picture according to teacher instructions. If a student uploads an image, the plugin will call the Object Recognition Service (ORS) hosted in Google Kubernetes Engine (GKE). This service will classify the image result and returns it to the Moodle website. Teachers and admin could view all submitted image classification results via 'View All Submission' page Figure 6. This page uses *assignment feedback plugin* called 'Object Recognition Feedback' to retrieve submitted image classification results. Students can't view the classification result, instead they could see if they completed the assignment via 'Submission' page Figure 7 once their submitted image has the same classification result with the desired image classification.

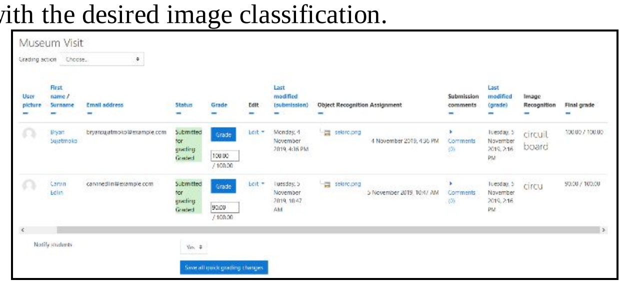

Figure 6: 'View All Submission' page

This gamification element could be implemented together with location tracker. For example: "visit the location of a national monument and submit a picture of it".

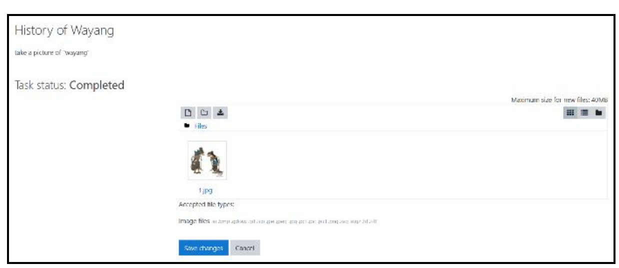

Figure 7: 'Submission' page

Teachers and admins could not add Object Recognition Assignments unless an existing model is available. This is where *block plugin* called 'Recognition Model Block' comes in to add or delete object recognition models. Only admins could access this plugin. The UI as shown in Figure 8 has simple design so admins could easily create models only by clicking few buttons. This slightly increases the website quality [24]. Instructions on how to prepare image files are displayed when admins clicked “create model” button. This plugin also stores categorized images locally in Moodle server to enable admins see an image sample of an object by clicking “Sample” hyperlink text. All existing categorized images used to create image models are displayed in tables separated by grades and subject name as shown in Figure 9.

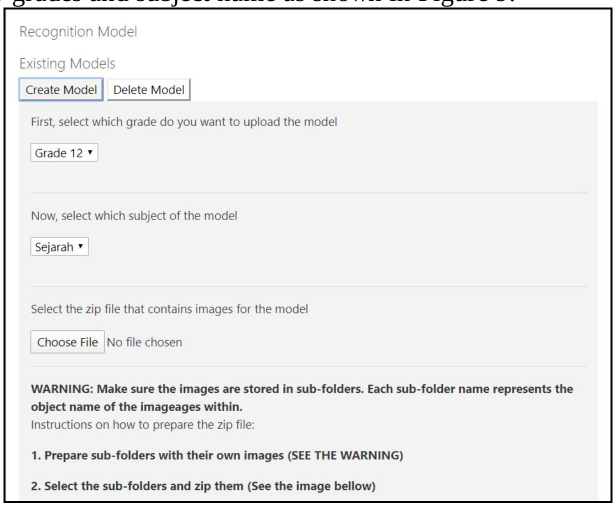

Figure 8: Create model display page

From the proposed architecture design, ATAM evaluation will focus on the quality attribute Performance (PE) and Scalability (SC). Table 2 shows evaluation scenario based on ATAM

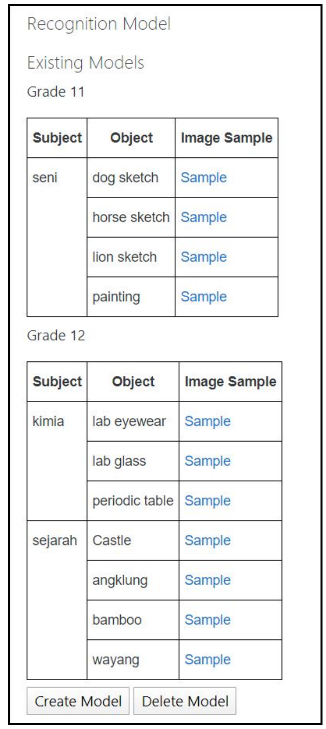

Figure 9: Existing image models displayed in table form

Table 2: ATAM evaluation scenarios of Quality Attributes

Quality Attribute (QA)	General Scenario QA	Concrete Scenarios
Scalability (SC)	**SC1 -** ORS-side stores data such as images and image models outside the web server engine to support horizontal scalability	Object Recognition Service will store submitted image and image models sent by admin. The service is hosted in GKE.
	SC2 – Object Recognition Service can easily scale up and scale out horizontally as the number of requests increases	30 number of user requests simultaneously to the Object Recognition Service. The number of concurrent users increases with each test iteration until it reaches 960 concurrent users. GKE Pod will be duplicated if the total CPU Pod usage reaches 50%.
Performance (PE)	**PE1 –** The LMS can be accessed by N number of concurrent users and does not reduce user experiences.	30 concurrent users visit Moodle LMS home page. The number of users increases with each test repetition until it reaches 480 users. System availability remains 98% and the average response type is under 30 seconds.
	PE2 – The Service can be accessed by N number of concurrent users and does not reduce user experiences.	30 concurrent users will simulate object recognition assignment submission to the Object Recognition Service and will be classified by the service. The number of users increases with each test repetition until it reaches 960 users. System availability remains 98% and the average response type is under 1 minute.

4. RESULT AND DISCUSSION

In the process of developing ORS, API is needed for Moodle LMS to connect to the ORS. The API serves to classify, create models, and delete models. Here are the lists of the API:

- /api/classify (POST): this API serves to receive image file from requests, classify the image, and return the classification result.
- /api/createModel (POST): this API serves to receive zipped file that contains categorized images, grade value, and subject value. The service will unzip the categorized images and stores them into structured folder based on grade value and subject value and create image models based on the categorized images.
- /api/deleteModelByGrade (DELETE): this API serves to delete all categorized images and image models based on grade value in the request.
- /api/deleteModelByGradeAndSubject (DELETE): this API serves to delete all categorized images and image models based on grade value and subject value in the request.

As mentioned in Table 2, performance testing is required to determine whether Moodle LMS and ORS could handle N concurrent user (CU) requests at once. Performance testing is performed using JMeter. The tests will record average response time (latency) in seconds and availability in percentages. For each error responses, it will reduce availability percentages count.

4.1. Moodle LMS Performance Test

Moodle LMS is hosted in GCP Computing instance with N1-standard-1 machine type with 1 CPU and 3.75 GB of RAM.

The first test starts with 30 concurrent users submitting GET type request for Moodle LMS home page. For each request, the server will return necessary home page data to the requester. The same test is repeated until the number of concurrent users reaches 480. Table 3 summaries the load test result.

Table 3: Moodle LMS test result

CU	Response time (s)	Availability (%)
30	0.743	100
60	1.453	100
120	2.889	100
240	5.621	100
480	11.368	100

4.2. ORS Performance Test

ORS is hosted in GKE with N2-standard-2 machine type with specification: 2 CPU and 7.5 GB of RAM. ORS load testing is divided into 2 tests: load test with Horizontal Pod Scaling (HPA) and load test without HPA. The first test starts with 30 concurrent users submitting POST type request for image classification. For each request, ORS will receive an image file, a subject value, and a grade value. The last 2 values are used to determine which image model to use to classify sent image file. After classifying sent image, ORS will return the classification result to the requester in JSON format. The same test is repeated until the number of concurrent users reaches 960.

A. ORS performance test without HPA

This test uses only a single pod in GKE node. In other words, only one application service that handles all request. The test result is shown in Table 4.

Table 4: POST /api/classify test result without HPA

CU	Response time (s)	Availability (%)
30	1.265	100
60	2.898	100
120	6.302	100
240	13.061	100
480	26.277	100
960	50.097	93.125

B. ORS performance test with HPA

Initially only a single pod handles all request. But when that pod utilizes more than 50% of CPU resources, it will replicate itself and all incoming request will be load balanced between all replicas / application service. Table 5 shows the HPA configuration while Table 6 shows the test result.

Table 5: HPA configuration to replicate pods in GKE

Autoscaler / Deployment	Parameter name	Value
Deployment	CPU	2
	RAM	7.5 GB
Autoscaler	Min Replica	1
	Max Replica	10
	CPU target utilization	50 %

Table 6: POST /api/classify test result with HPA

CU	Response time (s)	Availability (%)	Running Pod
30	1.255	100	1
60	2.827	100	1
120	6.437	100	1
240	12.997	100	1
480	26.457	100	1
960	53.060	99.06	2

The highest average response time is 53 seconds which failed to meet expected response time in the concrete scenarios. However, the all response time aside from 960 CU is under 30 seconds, which meets desired response time of concrete scenarios. Moreover, the availability is higher compared to no HPA because there are 2 pods / service that serves all incoming requests. The response time for 960 CU with HPA here is 3 seconds higher compared to no HPA because the server requires time to duplicate existing service.

4.3. ATAM Output

ATAM output is produced in the form of analytical results from the determined scenario. Concrete scenarios could be used to analyze the architecture, risks, and trade-offs of quality attributes. Table 7 shows the analytical results of concrete scenarios.

Table 7: ATAM Output for ORS scalability and performance

Scenario Summary	POST image classification requests are sent to the ORS that contains an image file, a subject value, and a grade value. The system will respond each request with image classification result under 1 minute.
Business Goal(s)	Make a scalable microservice
Quality Attributes	Scalability (SC1), Scalability (SC2), Performance (PE2)
Architectural Analysis	ORS is hosted in GKE. The service stores categorized images in structured folder along with image models created from the stored images.
Risk	Could increase rental cost of cloud storage.
Tradeoff	By separating ORS and Moodle LMS, it increases scalability, performance, and availability when scaling horizontally (scale out) even tough ORS need to clone images and models. However, this adds to the cost of renting cloud storage.

5. CONCLUSION

This study has implemented location-based learning and object-recognition learning as learning feature extensions in Moodle that utilizes external services like Leaflet, Google Maps, and Object Recognition Service.From LMS and ORS deployment architecture, it can be concluded that using containerization in GKE could support both location-based and object-recognition learning process. Admins and teachers could add location restriction or object recognition models using simple clicks and ORS will handle the rest.

Test results concludes that Moodle LMS is responsive enough to support up to 480 concurrent users with response time 11.37 seconds. The average response time for ORS classification result when accessed by 960 concurrent users is 50.097 seconds without HPA. While the average response time with HPA is 53.060 seconds. Even though ORS with HPA response time is lower, it has higher availability compared to ORS without HPA. For future works, it is suggested that image files and image models are stored separately in another server aside from ORS to reduce scale time and increase performance. Moodle LMS resources like database and file storage is placed separately instead storing them in the same server to increase scalability and performance.

This architecture may increase student's motivation and achievement during their study based on previous literature study. However, this claim is not evaluated in this study and should be done for the future works.

REFERENCES

1. L. A. Gibson and W. A. Sodeman, **Millennials and Technology : Addressing the Communication Gap in Education and Practice**, *Organization Development Journal*, vol. Winter, no. December, pp. 1–14, 2014.
2. S. Deterding, D. Dixon, R. Khaled, and L. Nacke, **From game design elements to gamefulness: Defining 'gamification'**, in *Proceedings of the 15th International Academic MindTrek Conference: Envisioning Future Media Environments, MindTrek 2011*, 2011, pp. 9–15, doi: 10.1145/2181037.2181040.
3. B. Perry, **Gamifying French Language Learning: A Case Study Examining a Quest-based, Augmented Reality Mobile Learning-tool**, *Procedia - Social and Behavioral Sciences*, vol. 174, pp. 2308–2315, 2015, doi: 10.1016/j.sbspro.2015.01.892.
4. T. Mailula, **Applying gamification in education to drive participation and engagement : A case study on grade 5 science learners at Lyttelton Primary School**, *Academia*, 2016.
5. H. Coates, R. James, and G. Baldwin, **A critical examination of the effects of learning management systems on university teaching and learning**, *Tertiary Education and Management*, vol. 11, no. 1, pp. 19–36, 2005, doi: 10.1007/s11233-004-3567-9.
6. K. C. Green and S. W. Gilbert, **Great Expectations: Content, Communications, Productivity, and the Role of Information Technology in Higher Education**, *Change: The Magazine of Higher Learning*, vol. 27, no. 2, pp. 8–18, 2010, doi: 10.1080/00091383.1995.9937733.
7. K. T. Pardue and P. Morgan, **Millennials considered: A new generation, new approaches, and implications for nursing education**, *Nursing Education Perspectives*, vol. 29, no. 2, pp. 74–79, 2018, doi: 10.1097/00024776-200803000-00007.

8. K. A. C. Castro, Í. P. H. Sibo, and I. hsien Ting, **Assessing gamification effects on E-learning platforms: An experimental case**, in *Communications in Computer and Information Science*, 2018, vol. 870, pp. 3–14, doi: 10.1007/978-3-319-95522-3_1.
9. H. F. Hasan, M. Nat, and V. Z. Vanduhe, **Gamified Collaborative Environment in Moodle**, *IEEE Access*, vol. 7, pp. 89833–89844, 2019, doi: 10.1109/ACCESS.2019.2926622.
10. C. Becker, G. Lauterbach, S. Spengler, U. Dettweiler, and F. Mess, **Effects of regular classes in outdoor education settings: A systematic review on students' learning, social and health dimensions**, *International Journal of Environmental Research and Public Health*, vol. 14, no. 5, pp. 1–20, 2017, doi: 10.3390/ijerph14050485.
11. R. Hunicke, M. LeBlanc, and R. Zubek, **MDA: A Formal Approach to Game Design And Game Research**, *Workshop on Challenges in Game AI*, pp. 1–4, 2004, doi: 10.1.1.79.4561.
12. R. N. Landers and M. B. Armstrong, **Enhancing instructional outcomes with gamification: An empirical test of the Technology-Enhanced Training Effectiveness Model**, *Computers in Human Behavior*, vol. 71, pp. 499–507, 2015, doi: 10.1016/j.chb.2015.07.031.
13. A. Mohd Amir and S. Dalbir, **Schoolcube: Gamification for learning management system through microsoft sharepoint**, *International Journal of Computer Games Technology*, vol. 2015, 2015, doi: 10.1155/2015/589180.
14. D. Dicheva, K. Irwin, and C. Dichev, **Oneup learning: A course gamification platform**, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10653 LNCS, no. Gala, pp. 148–158, 2017, doi: 10.1007/978-3-319-71940-5_14.
15. Z. G. Ge, **The impact of a forfeit-or-prize gamified teaching on e-learners' learning performance**, *Computers and Education*, vol. 126, pp. 143–152, 2018, doi: 10.1016/j.compedu.2018.07.009.
16. Y. W. Lam, K. F. Hew, and K. F. Chiu, **Improving argumentative writing: Effects of a blended learning approach and gamification**, *Language Learning & Technology*, vol. 22, no. 1, pp. 97–118, 2017, doi: 10125/44583.
17. A. Bernik, D. Radošević, and G. Bubaš, **Introducing gamification into e-learning university courses**, in *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2017 - Proceedings*, 2017, no. May, pp. 711–716, doi: 10.23919/MIPRO.2017.7973515.
18. G. Tuparov, D. Keremedchiev, D. Tuparova, and M. Stoyanova, **Gamification and educational computer games in open source learning management systems as a part of assessment**, *2018 17th International Conference on Information Technology Based Higher Education and Training, ITHET 2018*, pp. 1–5, 2018, doi: 10.1109/ITHET.2018.8424768.
19. R. Kazman, M. Klein, M. Barbacci, T. Longstaff, H. Lipson, and J. Carriere, **The Architecture Tradeoff Analysis Method**, *Security*, 1998.
20. L. Bass, M. Klein, G. Moreno, and S. E. Institute, **Applicability of General Scenarios to the Architecture Tradeoff Analysis Method**, *Carnegie Mellon University Technical Report*, no. CMU/SEI-2001-TR-014, ESC-TR-2001-014, 2001.
21. A. D. M. Africa, G. Ching, K. Go, R. Evidente, and J. Uy, **A comprehensive study on application development software systems**, *International Journal of Emerging Trends in Engineering Research*, vol. 7, no. 8, pp. 99–103, 2019, doi: 10.30534/ijeter/2019/03782019.
22. M. Halim, N. Adadi, D. Chenouni, and M. Berrada, **Web Services Composition in E-Learning platform**, *International Journal of Emerging Trends in Engineering Research*, vol. 8, no. 2, pp. 525–532, 2020, doi: 10.30534/ijeter/2020/41822020.
23. K. David Harjono and G. P. Kusuma, **Object instance recognition using best increasing subsequence**, in *Proceedings - 11th 2016 International Conference on Knowledge, Information and Creativity Support Systems, KICSS 2016*, 2016, doi: 10.1109/KICSS.2016.7951432.
24. B. V. Priya and J. K. R. Sastry, **Assessment of website quality based on appearance**, *International Journal of Emerging Trends in Engineering Research*, vol. 7, no. 10, pp. 360–375, 2019, doi: 10.30534/ijeter/2019/017102019.