Implementation of POLIMDO Academic Information System Webservice Based on Microservice Architecture

Diane Tangian, Edwin Lumunon and Maksy Sendiang
Manado State Polytechnic, Indonesia

Keywords: Microservices, Web Service, Academic Information System, REST.

Abstract: The research sought to provide POLIMDO academic information system web services utilizing a microservices architecture. The JavaScript Object Notation (JSON) response format and the Representational State Transfer Protocol (REST) were used in the development of the web service. Meanwhile, the Domain-driven Design (DDD) methodology was used to create the microservices architecture. Business process analysis was used to model the research using the Unified Modeling Language (UML), which was then used to model academic system business processes. The next step was to create a modest, responsible service for a function using the bounded context. The system was created using the javascript technology, and the database-per-service data management architecture. Each endpoint was put to the test using the Postman program according to the white-box testing methodology. The academic system web service can function as predicted based on the findings of the evaluation procedure. The outcomes also demonstrate the system's great resiliency. It denotes that the system is flexible and has few interdependencies among its components.

1 INTRODUCTION

An academic information system is a tool designed to process official and informal academic data and information in educational settings from elementary school through college. The academic information system, in essence, is a program that can speed up the processing of data and information pertaining to academic issues (Farid Suryandani, 2017). The employment of academic information systems can overcome obstacles faced and ensure that data and information get to the user as soon as possible. Another advantage of employing the Academic Information System (SIA) is that universities can enhance the efficacy and efficiency of managing academic administration, as well as the caliber of services provided to stakeholders, particularly students, and the competitiveness of universities (Yindrizal,Sri Suwitri, 2019).

The usage of an Academic Information System (SIA) can add value to the decision-making and problem-solving processes and have competitive advantages that considerably enhance corporate activities and universities (Kroenke in Kadir ,2008). Academic information systems that allow quick access to academic information can be advantageous to students as end users. According to the findings of a study done in 2016 by Firma S. B., Muhammad U.S., and Ovide D.W.A., academic information systems are necessary in universities since they can improve performance and make documentation easier. Manado State Polytechnic has implemented an academic information system but it still has many weaknesses, including the system architecture is still traditional (monolithic).

A standardized method of spreading communication between clients and servers is through web services. Web services are typically created using the Simple Object Access Protocol (SOAP) and the Representational State Transfer Protocol (REST) (Tihomirovs & Grabis, 2016). The data format for REST can be either Extensible Markup Language (XML) or JavaScript Object Notation (JSON). In contrast, the sole data format that SOAP may use is XML. Better performance, more simplicity, quick execution, more scalability, more loose coupling, and less memory usage are benefits of REST over SOAP. However, SOAP is more dependable and safe. Due to its greater emphasis on security and dependability, such as in banking, finance, and telecommunications services, SOAP is best suited for long-haul working projects. Due to its emphasis on simplicity and performance, REST is best suited for large-scale initiatives like web chats and mobile services (Soni & Ranga, 2019).

In the past, monolithic architecture was typically used to build systems. A single-tiered software application with a monolithic design combines the user interface and data access code into a single program from a single platform is known as such. The system will continue to undergo changes over time as a result of modifications to or expansions of the business procedures that increase the complexity and size of the application. The capacity of monolithic architecture to adapt when the system's requirements change, particularly in managing its code complexity and code maintainability, is one of its weaknesses. Due to its high degrees of dependency, it will cause issues during the distribution process (Munawar & Hodijah, 2018). Changes to one section of the code will have an impact on the remaining sections. Consequently, the other components of the code must also be modified.

The system will be created using a microservices design to address that issue. Microservices are tiny services. A large service is divided into smaller services using microservices. Each service has responsibilities and runs separately. As a result, there are little dependencies between services (loose coupling). Since there are less dependencies between services, the system can change more easily (Munawar & Hodijah, 2018). Additionally, the developer will find it simpler to change a few services and distribute them without altering the entire system.

A method for creating a microservices infrastructure is called domain-driven design (DDD). It concentrates on the domain, which comprises ideas, connections between domains, and current business procedures. On the microservice architecture, services are identified using bounded context. Grouping business processes by their functions is a good idea. Each group will develop into the system's bounded context (Newman, 2015). In the microservices architecture, bounded context is then transformed into discrete services.

2 RESEARCH METHODOLOGY

Figure 1 depicts the research methodology that was employed. Functional analysis is the first stage. It involves locating and examining business processes. Then, information is gathered to determine the features that will be offered by the system. There is an actor who can access each feature's function.

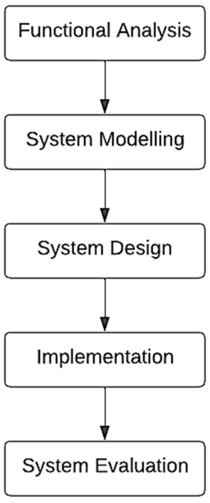

Figure 1: Research Method.

System modeling is done in the second step. In accordance with the findings of the functional analysis, the system is modelled. The Unified Modeling Language is used to model it (UML). The outcome, in the form of a use case diagram, illustrates how actors and the system interact. System design is the third step. The microservices are designed during this phase. The findings cover the technology employed and the setup of the system. The system is built using the javascript technology namely react.js dan express.js. Next, a database-per-service design approach is used to create the database. This design pattern enables the selection of an appropriate database for each service, preserves the data required by connected services, and forbids direct database access by other services. Only API contact between services will allow for data exchange.

Implementation is the fourth step. At this stage, the system is constructed using the chosen technology. Using the database-per-service design, the database is the first component to be developed and built. There is a database for each service. The web service is coded after the databases are created. The transport layer, endpoint layer, and service layer are the three levels that make up. The transport layer is where requests first enter the system, then they move down to the service layer.

System evaluation is the last action. The system is assessed to make sure it can function as intended and has a high level of resilience. White box testing is used to test each system endpoint. Following that, test outcomes are contrasted with predictions. The endpoint will be if the test results and expectations are similar.

3 RESULT AND DISCUSSIONS

The system needs to do a functional analysis on the produced data in relation to the features. According to the findings of the functional analysis, there are five actors in the system: the WADIR (vice director for academic affairs) the KAJUR (head of department), the Kapus P3M (head of research and community service centre), the Kapus P4M (head of quality management) and the LSP (professional certification body). WADIR is a component of the system, and one of its responsibilities is developing work plans for the academic sector. The research and community service centre, the centre for quality assurance and learning development, departments, and professional certifying unit will eventually put the work program into action. Each of these organizations will put into practice work programs that are pertinent to their individual primary duties and responsibilities. Table 1 displays the findings of the functional analysis. Each actor's activity within the system is displayed.

Table 1: Result of functional analysis.

Actor	Activity
WADIR	o Manage a POLIMDO work program in academic field o See & print performance reports o Manage feedback information o Manage account information
KAJUR, Kapus P3M, Kapus P4M & LSP	o Manage performance reports
KAJUR	o Manage students' information o Manage lectures information o Manage laboratory information
Kapus P3M	o Manage research information o Manage community service information
Kapus P4M	o Manage quality assurance information o Manage learning information
LSP	o Manage certification information

A use case diagram is created using the information in Table 1 as a basis. The interaction between the system and the actors is depicted in the use case diagram. A use case diagram for the interactions between the actors and the system is shown in Figure 2.

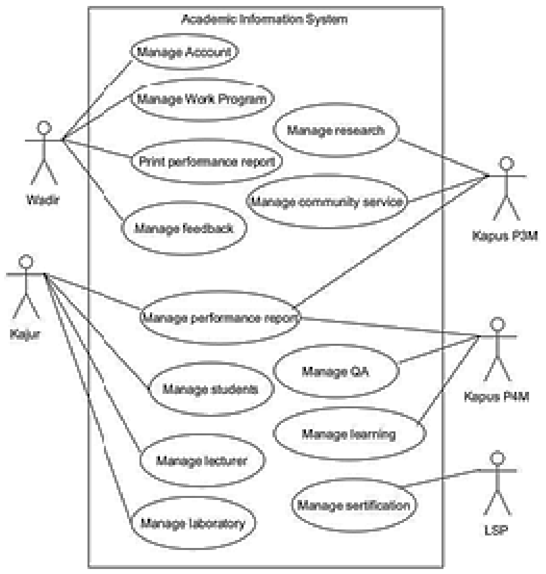

Figure 2: Use case diagram.

In this system WADIR can manage account, manage work program, print performance report and manage feedback. Kajur can manage performance report, manage students and lecturer, and finally can menage laboratory. In other hand, Kapus can manage research, manage community service, manage QA and manage financing. The last LSP manage well the certification.

The bounded context in the system is established based on the business process after producing the use case diagram. Implementing a business process in the system is the responsibility of a logical constraint. Bounded context is used by the domain-driven design (DDD) methodology to identify services in the microservices architecture. The system's bounded contexts are displayed in Table 2. The system has six services: account, report, research, quality assurance, LSP and department.

The account service is responsible for managing all system users, including granting access rights according to their respective roles. The Report service is responsible for managing reports made by each work unit including displaying the required data in graphic form in real time. The research service is responsible for managing research activities and community service within the Manado State Polytechnic. This activity starts from the process of submitting proposals, evaluating proposals by reviewers, announcement of winners to the annual monitoring and assessment process for research and service activities carried out. The quality assurance service is responsible for managing the quality assurance system at the Manado State Polytechnic.

Table 2: Bounded Context.

Bounded context	Description
Account	This service is responsible for managing account data, registering a new account, and handling the users' login process.
Report	This service is responsible for managing report data, and display report in graphics form
Research	This service is responsible for managing research and community service data, manage research and community service proposals, manage grant, and manage fund.
Quality assurance	This service is responsible for managing QA data, manage accreditation and manage learning process
LSP	This service is responsible for managing LSP data and manage certification.
Department	This service is responsible for managing department data, manage students data, manage lecturer data and manage laboratory data

It includes the internal audit process, routine monitoring of the implementation of quality assurance by the quality control team, accreditation of institutions and study programs and learning development. The LSP service is responsible for managing all certification activities carried out at the competency unit at the Manado State Polytechnic. LSP also manages assessor data and manages certification documents. The department service is responsible for managing activities at the department level, especially those related to academic activities. The department manages the data of lecturers, students, curriculum, laboratories and everything related to the teaching and learning process.

REST architecture is used to build web services. There are five HTTP (Hypertext Transfer Protocol) methods that can be utilized with REST design. The POST, GET, PATCH, DELETE, and PUT methods are among them. A new resource is created using POST. GET returns a particular resource. To update a specific resource, use PATCH. A specific resource can be deleted, and all specific resources can be replaced. The endpoints and their function in the system is shown below:

Table 3: The Endpoints.

Service	Endpoints - Method	Function
Account	/signup - POST	Create customer account
	/signin - POST	Sign in to existing account
	/user/{user_id} – GET	Show list of user
	/user/id/{user_id} - GET	Show information about selected user
	/user – POST	Add new user
	/user – PATCH	Change selected user
	/user/{id} – DELETE	Delete selected user
Report	/report – POST	Add new report data
	/report – PATCH	Change the selected report data
	/report/{user_id} – GET	Show list of report made by an account
	/report/id/{id} – GET	Show information about selected report
	/reported/id – GET	Show list of report
	/report/{id} - DELETE	Delete selected report
Research	/research – POST	Add new research data
	/research – PATCH	Change the selected research data
	/research/id/{id} - GET	Show information about selected research
	/research/id - GET	Show list of research
	/research/{user_id} - GET	Show research made by an account
	/research/{id} – DELETE	Delete selected research
	/research/{user_id} – DELETE	Delete research made by an account
Quality assurance	/quality – POST	Add new quality assurance data
	/quality – PATCH	Change the selected quality assurance data
	/quality/{id} – DELETE	Delete the selected quality assurance data
	/quality – GET	Show list of quality assurance information
	/quality/id/{id} – GET	Show a selected quality assurance data
LSP	/lsp – POST	Add new LSP data
	/lsp – PATCH	Change a selected LSP data
	/lsp – GET	Show list information about LSP
	/lsp/id/{id} – GET	Show list information about the selected LSP
	/lsp/{id} – DELETE	Delete the selected LSP data
Department	/department – POST	Add new department data
	/department – PATCH	Change the selected department data
	/department/id – GET	Show list of department
	/department/id/{id} – GET	Show a selected department data
	/department/{id} – DELETE	Delete the selected department

An API Gateway is used to simplify access to each of the current endpoints. In a microservice design, API Gateway serves as a single point of entry. The internal systems created behind the API Gateway can be isolated because it also functions as middleware. The API Gateway is set up to handle load balancing, logging, authentication, rate limitation, caching, and other tasks. The open source Kong API Gateway is employed in this study. The following figure depicts the design of this system's microservice architecture:

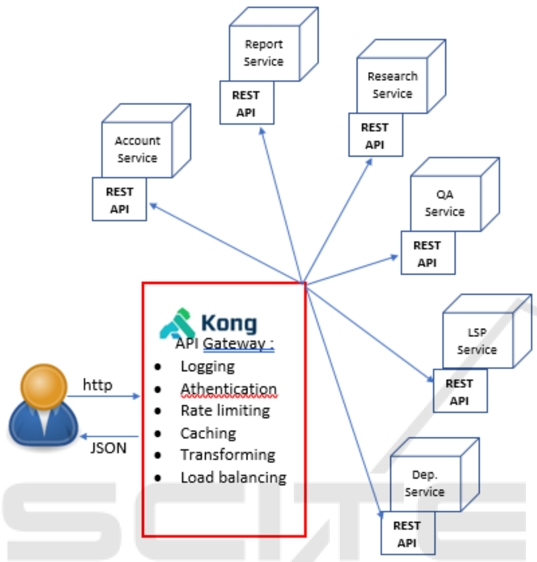

Figure 3: Microservice Architecture.

Microservices are the blocks of application and perform different services, while REST APIs work as the glue or the bridge that integrates these separate microservices. The system was developed utilizing HTTP and REST. Every request results in a response that is transmitted as JSON. The system is then outfitted with a security layer to safeguard user data and system-to-user communication protocols. By employing the BCrypt hashing algorithm, passwords are safeguarded. The blowfish encryption algorithm serves as the foundation for BCrypt. It satisfies three security requirements that are thought to be enough to safeguard a system. Additionally, it has a second preimage resistance, enough salt capacity (in cryptography) to thwart pre-computational attacks, and flexible costs (Provos & Mazieres, 1999). When users want to access the system's endpoints, they employ JSON Web Tokens (JWT). By comparing the token claims, JWT will make sure that only specific accounts have access to the data.

The system is then assessed using the white-box testing methodology. The outcome of the some system evaluation is displayed in Table 4.

In order to implement system evaluation, a system failure simulation is used to assess the system's resilience. Randomly, a number of service nodes are terminated. Tests are conducted on service nodes that are still operational. The test's findings demonstrate that service nodes that do not encounter interference can operate normally.

Table 4: Result of System Evaluation.

Examination	Expectation	Reality	Result
WADIR create a new user account and fill in the required information	Creating a new account, encrypting the password, and storing all of the data in the database	Creating a new account, encrypting the password, and storing all of the data in the database	Success
WADIR creates a new user account and do not fill in the required information.	The system returns an error message	The system returns an error message	Success
Users input the correct email and password combination.	Users log in to the system	Users log in to the system	Success
Users input incorrect email and password combination	System returns an error message indicating the incorrect combination of email and password.	System returns an error message indicating the incorrect combination of email and password.	Success
Users see their personal information. The user_ID is sent to the system	The system receives their user_ID and shows users' information	The system receives their user_ID and shows users' information	Success
WADIR select one user and change the user information	System changes the selected user information	System changes the selected user information	Success
WADIR select one user and delete it	The selected user is deleted from the database	The selected user is deleted from the database	Success
Users create a new report and fill in the required information	Creating a new report, and storing all of the data in the database	Creating a new report, and storing all of the data in the database	Success
Users select one report and change the report information	System changes the selected report information	System changes the selected report information	Success
User select report based on user_id	The system receives user_ID and shows the appropriate report	The system receives user_ID and shows the appropriate report	Success
User displays the relevant report, select one report and delete it	The selected report is deleted from the database	The selected report is deleted from the database	Success
KAJUR creates a new department data and fill in the required information	Creating a new department data, and storing all of the data in the database	Creating a new department data, and storing all of the data in the database	Success
KAJUR select one department data and change the department information	System changes the selected department information	System changes the selected department information	Success
KAJUR select one department and delete it	The selected department is deleted from the database	The selected department is deleted from the database	Success

4 CONCLUSIONS

The POLIMDO academic information system web service can function as predicted based on the findings of the evaluation procedure. The evaluation's findings also demonstrate the system's strong resiliency. It denotes that the system is flexible and has few interdependencies among its services. KONG API gateway provides a lot of convenience in managing microservice-based systems. Features that are common in every service such as load balancing, logging, authentication, rate limitation, caching can be pulled to be handled directly on the API Gateway. This greatly impacts the access speed of the existing endpoint.

REFERENCES

Farid Suryandani, Basori, Dwi Maryono, "Pengembangan Sistem Informasi Akademik Berbasi Web Sebagai Sistem Pengolahan Nilai di SMK Negeri 1 Kudus", jurnal Ilmiah Pendidikan Teknik dan Kejuruan vol 10 no 1 2017

Firma Sahrul B, Muhammad Asri Safi'ie, Ovide Decroly W A., 2016, "Implementasi Sistem Informasi Akademik Berbasis Web Menggunakan Framework Laravel", Jurnal Transformasi, Vol. 12, No. 1, Juni 2016 : 46 – 50

Kadir, Abdul. 2008. Pengenalan Sistem Informasi. Penerbit ANDI: Yogyakarta.

Munawar, G., & Hodijah, A. (2018). Analisis model arsitektur microservice pada sistem informasi DPLK. Sinkron: Jurnal dan Penelitian Teknik Informatika, 3(1), 232-238.

Newman, S. (2015). Building microservices: Designing fine-grained systems. USA: O'Reilly Media, Inc.

Soni, A., & Ranga, V. (2019). API features individualizing of web services: REST and SOAP. International Journal of Innovative Technology and Exploring Engineering, 8(9S), 664-671.

Tihomirovs, J., & Grabis, J. (2016). Comparison of SOAP and REST based web services using software evaluation metrics. Information Technology and Management Science, 19(1), 92-97.

Yindrizal,Sri Suwitri,Nufransa Wira Sakti, "Impact of the Use Academic Information System to Improve Student Satisfaction of Academic Administration Services in Universitas Andalas Padng",Internationa Journal of Scientific and Engineering vol.10 issues 5 2019

Provos, N., & Mazieres, D. (1999, June). A future-adaptable password scheme. In *USENIX Annual Technical Conference, FREENIX Track* (Vol. 1999, pp. 81-91).