

Implementation of Model View Controller Architecture in Designing Outcome-Based Education (OBE) Curriculum Management Information System

Fortia Magfira*, Tekad Matulatan, Novrizal Fattah Fahmitra, Feri Irawan, and Rifaldi Herikson

Informatic Engineering Department, Universitas Maritim Raja Ali Haji, Indonesia

Abstract. The development of 21st century education demands the integration of practical and collaborative skills that are relevant to global needs, where the Outcome-Based Education (OBE) approach is the key to achieving measurable and clear learning outcomes. This study focuses on the design of an OBE-based curriculum management information system by implementing the Model View Controller (MVC) architecture and the Laravel framework which aims to overcome the constraints of manual curriculum management at Universitas Maritim Raja Ali Haji, which requires significant time to collect, organize, and analyze data, thus hampering lecturer productivity and interaction with students. The development of the system not only integrates various components of the OBE curriculum, but also creates a collaborative platform in curriculum management. Based on the results of the study that have gone through usability testing with the explorative test method, involving all sub-system designers and stakeholders, this system shows a high level of effectiveness, efficiency, and user satisfaction. Thus, this study is expected to improve the efficiency and effectiveness of OBE curriculum management, which contributes to improving the quality of education at the institution.

Keywords: Outcome-based Education, System Design, Curriculum Management Information System.

1 Introduction

The development of the world of education has now entered the 21st century, marked by the era of information technology that continues to develop and affects almost all aspects of human life, including education. The 21st century education emphasizes the importance of skills that are not only academic but also practical, collaborative, and adaptive, which can support students in facing challenges that are developing globally. In this case, the Outcome-Based Education (OBE) approach is a relevant approach to facing the development of 21st century education that is able to direct the learning process to achieve clear and measurable results. The Outcome-Based Education (OBE)-based educational approach is the basis for the education system to achieve educational goals (output), so that when they graduate, students must achieve these goals [3]. The Outcome-Based Education approach is one form

* Corresponding author: fortiamagfira@umrah.ac.id

of educational approach that emphasizes the sustainability of an effective learning process, has innovation in building good interactions, which are important in the learning process, curriculum design, learning strategies and achievements, assessment procedures, educational ecosystems, and others that are oriented towards the future to produce outputs that have better skill specifications [13]. In the context of modern education, Outcome-Based Education (OBE) plays an important role with the emphasis on achieving specific and relevant learning outcomes and aligning the curriculum with industry needs. OBE also contributes to increased accountability and transparency. This approach not only makes education more focused and effective, but also prepares students with the skills and knowledge needed to succeed in the workplace and everyday life. By supporting personalized learning, improving assessment systems, and encouraging continuous improvement, OBE significantly contributes to the development of better and more responsive education [10].

The implementation of the OBE approach requires infrastructure support that can facilitate the management and monitoring of competency achievement effectively. In this case, the OBE-based curriculum management information system plays a crucial role with the integration of information technology that not only integrates various components of the OBE curriculum, but also creates a collaborative platform that allows collaboration between lecturers in compiling and managing curriculum components, optimizing the assessment process according to the learning outcomes set, minimizing errors, increasing the efficiency and effectiveness of curriculum data management, thus providing direct benefits to lecturers and students, especially at Universitas Maritim Raja Ali Haji where the implementation of the OBE-based curriculum still relies on manual methods without information system integration in several study programs.

The implementation of the OBE curriculum without the integration of this information system is less efficient and effective in compiling the OBE curriculum components that must be integrated in the study program. In fact, lecturers often find obstacles in organizing curriculum data, especially the Semester Learning Plan (RPS) to align with other OBE curriculum components such as Sub-CPMK, CPMK, and CPL, both directly and indirectly. Difficulty in organizing curriculum data manually can cause inconsistencies in the preparation of RPS and difficulties in ensuring that all curriculum components are well integrated. For example, if the RPS does not match the desired learning outcomes, this can have a negative impact on the learning process and student competency achievement. The manual process takes a lot of time to collect, organize, and analyze data, which hinders lecturer productivity and reduces the time they can allocate for teaching and interacting with students, supervision and monitoring of student competency achievement becomes difficult. Without an adequate system, lecturers find it difficult to systematically track student progress, which is important for assessing whether learning objectives are achieved, resulting in a lack of accountability and transparency in the evaluation and assessment process. Without an integrated system, it is difficult to clearly demonstrate student learning achievements, which can reduce stakeholder confidence in the effectiveness of the study program.

This research focuses on designing a curriculum management information system based on Outcome-Based Education by implementing the Model View Controller (MVC) architecture with the Laravel framework as a web application and using the Application Programming Interface (API) as a backend. The Application Programming Interface (API) acts as a communication channel between subsystems that allows the exchange of data and instructions between the two. This research is expected to help study programs in managing data and analyzing the OBE curriculum matrix easily, can help lecturers describe graduate learning outcomes (CPL) into course learning outcomes (CPMK) and create Semester Learning Plans (RPS) collectively and collaboratively and can bring significant changes in the curriculum management process practically.

2 Research Methods

2.1 Related Research

There are several research journal literatures that apply MVC architecture to their research. Furthermore, this research is also related to management information systems that involve users directly with system analysis and design and are effective for system correction. Thus, with this system, it can help stakeholders in managing data and producing accurate, up-to-date, secure and available information for users [7]. Other research has produced a website and has been tested with a success rate of 66.7% indicating that the website has the ability to handle data [1]. Other research shows that the design of the Task Management System with the implementation of the MVC architecture is good and is in accordance with the wishes and needs of users, with an average test score of 89 points [12]. Similar research also shows that MVC is effective in understanding, discussing, presenting, and implementing the research topic, namely interactive graphic applications [4]. The website-based OBE assessment application was developed using the waterfall method, in accordance with the stages in the software development life cycle that is able to display final grades based on courses and graduate learning outcomes, and has shown black-box test results with an achievement rate of 86% in terms of design, ease of use, and function [14]. The development of a web-based information system as a study program information media can help meet the needs of the community for information related to study programs and support study programs in publishing this information. The development process of this system uses the waterfall method, followed by black-box testing to ensure that the system functions as expected before finally being handed over to end users [2]. The application of the PIECES analysis method and system design aims to improve and increase the effectiveness of industrial work practice data management. This system has been tested using the black-box testing method, and the results show that the system functions well. In addition, this system can support the needs of industrial work practices more effectively, in accordance with the Merdeka Curriculum [11].

2.2 Methods

This study focuses on designing an integrated OBE-based curriculum information system, so that the stages of this study use the system development life cycle method with its development model, namely Prototyping. The Prototyping model is a technique for collecting certain information about user information needs quickly [17]. This method will produce a system prototype that functions as an intermediary between developers and users, allowing them to interact during the information system development process [18]. The application of the prototype method in this study is because this method will produce a system prototype that functions as a communication bridge between developers consisting of system analysts and programmers with users through direct interaction during the development process. So that user needs and expectations will be easier to understand. The prototype that has met user needs will be used as the basis for developing the final system. The prototype method used in this study can be seen in Figure 1.

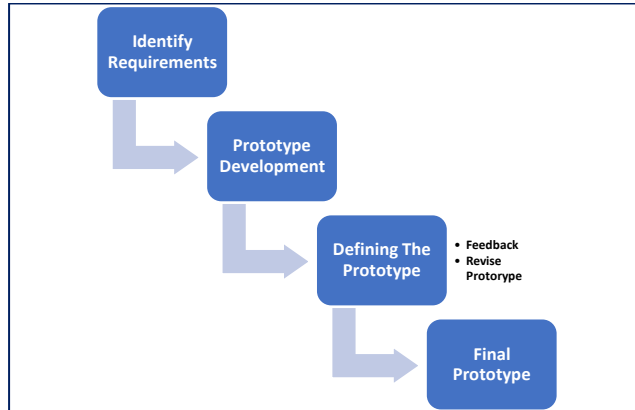


Fig 1. Prototype Method

The following are the stages of the prototyping method applied in this study [16]:

1.2.1 Identifying user needs

At this stage, the system analyst conducts a feasibility study and a study of user needs. The stage where all system needs are identified and defined. Needs are obtained from literature studies of documents related to the preparation of the OBE curriculum and interviews with relevant stakeholders who are directly or indirectly involved in the preparation of the OBE curriculum in the study program. In this case, the reference document for the OBE curriculum for the Informatics Engineering study program as a reference for system design and lecturers from the Informatics Engineering study program at Universitas Maritim Raja Ali Haji as resource persons to answer the needs of system users later.

1.2.2 Prototype development.

The system analyst works with other sub-component programmers to develop a system prototype to show stakeholders the system modeling to be built.

1.2.3 Determining the prototype

At this stage, the system analyst will determine to what extent the modeling he has created is acceptable, what improvements stakeholders want, or whether it is necessary to completely overhaul the existing modeling. This stage includes prototype testing, feedback and revision.

1.2.4 Final prototype

Once the prototype is approved and meets the requirements, the final prototype is used as the basis for developing the final system.

The application of the prototype method in this study provides many conveniences in facilitating the identification of user needs by allowing system analysts to interact directly with stakeholders, such as lecturers and related parties, to understand their expectations and needs. The prototype method provides flexibility in design, allowing changes and adjustments to be made iteratively according to user input that may change during the process. In addition, the prototype allows for early testing of the system, so that problems and additional needs can be identified early, reducing the risk of failure in the final stage.

The use of prototypes increases collaboration between analysts, programmers, and users, creating a better shared understanding of the system to be built. Thus, this method contributes to creating an information system in terms of designing a more effective system and in accordance with user needs in the context of OBE.

2.3 Model View Controller Architecture

The MVC concept divides the design of an information system into three layers, namely model, view, and controller. The model is used to organize information and notify observers when information changes. Only the model contains data and functions related to data processing. The view is responsible for mapping graphics to the device. The controller receives input from the user and instructs the model and view to perform actions based on that input [5]. The MVC concept is a pattern that has proven to be more effective in developing a software project, by selecting components in the project that will be used as models, views, and controllers [15]. The MVC concept can be seen in Figure 2.

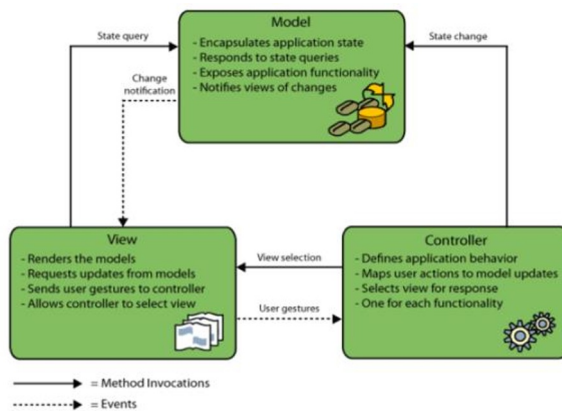


Fig 2. Model View Controller Architecture

In addition to the MVC architecture, there are several architectures that can be used such as MVVM (Model-View-ViewModel), MVP (Model-View-Presenter), Microservices, RESTful Architecture, and Event-Driven Architecture. Here are the differences between these architectures in Table 1.

Table 1. Table of Difference between MVC, MVVM, MVP architecture

Architecture	Advantages	Disadvantages
Model View Controller	Facilitates maintenance and unit testing, separation of responsibilities.	Initial complexity and overhead in development
Model View View Model	Supports a cleaner separation between UI and business logic, and allows the use of data binding to automatically update Views.	More complex, requires in-depth understanding
Model View Presenter	Separates presentation logic from the Model, making it easier to perform unit testing.	Interaction management can be tricky

The gap in existing curriculum management systems generally lies in the inability of the system to fully support the Outcome-Based Education (OBE) approach. Existing systems often focus on curriculum input (courses, credits) rather than output (competency achievement). In addition, lack of flexibility, limitations in reporting and data analysis, and lack of integration with other systems are also problems. The implementation of the Model-View-Controller (MVC) architecture in an OBE-based curriculum management system can overcome this problem by separating business logic, views, and data so that scalability increases, ease of maintenance, with a clear separation between business logic, user interface, and application flow control, the development team can more easily identify and fix problems. It also makes it easier to make updates and changes, because each component can be handled separately. The separation of concerns carried out by the MVC architecture allows developers and designers to focus on specific aspects of the application. Developers can concentrate on backend logic and data processing, while designers can work on the user interface, thereby increasing the efficiency of the team as a whole. This allows the system to be more flexible, easy to adapt, and able to produce more comprehensive reports. Thus, MVC can provide better support in achieving the OBE goal, which is to ensure that students truly master the expected competencies. The model view controller architecture is implemented together with the laravel framework as a web application.

2.4 Application Programming Interface (API)

Application Programming Interface serves as an interface between various software components, allowing them to communicate and interact with each other [6]. The purpose of using API is to share data between different applications, to speed up the application development process by providing a separate function so that developers no longer need to design similar features [9]. In this study, the role of API in connecting subsystems is to become a link between various components in the OBE curriculum management system. For example, API can connect RPS with learning process analysis and student assignment plans, ensuring that all elements are integrated with each other. This allows relevant information to be easily shared between subsystems, increasing the effectiveness of curriculum management. API supports scalability which is very important for curriculum adaptation. When there is a change in education policy or the addition of new components, such as additional learning modules or new formats for lecture contracts, API allows developers to integrate these changes without having to overhaul the entire system. In terms of flexibility, the system can easily adapt to changes needed in the learning process. For example, if teachers want to update student assignment plans or improve elements in RPS, they can do so with minimal impact on other components. This makes the system design more adaptive and responsive to user needs.

3 Analysis and Discussion

This study focuses on the analysis of the needs of several main business processes of the OBE-based curriculum management information system, namely the preparation of learning process planning.

3.1 System Requirement

Based on the feasibility study and needs study, an information system architecture or framework is identified that describes the structure, components and relationships between elements in the information system that will be built later, as in Figure 3.

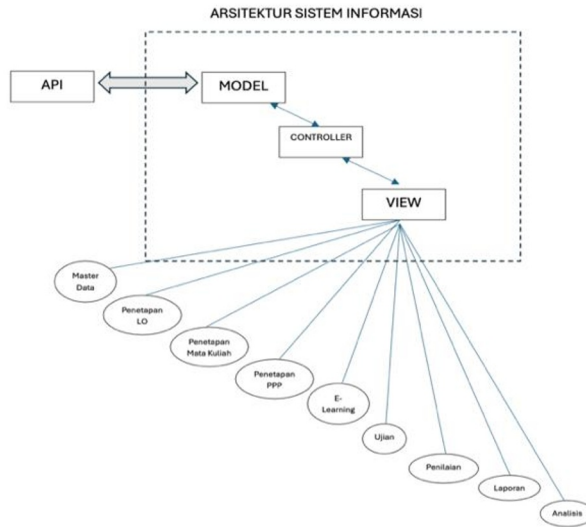


Fig 3. Architecture system of OBE curriculum management information system.

This architecture encourages separation of responsibilities, meaning each component has a clear, specific task. This makes it easier to develop, test, and maintain applications. In addition, API's allow applications to quickly adapt to changing business needs and integrate with new services without affecting other components. This architecture serves as a guide for the development, implementation, and maintenance of information systems, ensuring that all elements work harmoniously to meet the needs of the organization. In addition, a system connectivity architecture is also identified that regulates how various components in the system to be built will interact and communicate with each other as in Figure 4.

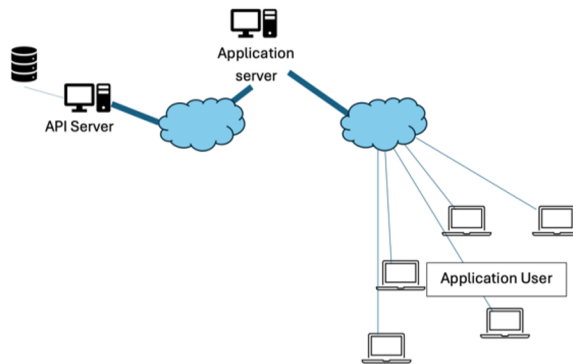


Fig 4. Connectivity Architecture of OBE curriculum management information system.

Communication System between user and application through Application server in internet connectivity where application does not have direct connection to database but to API server. API server only allows connection with Application Server, where direct database access request is not possible. This disconnects database from outside world, so hacker is impossible to hack.

3.3 Analysis and Design

In the analysis and design stages for each module of the curriculum management information system, a system development document is created consisting of use case diagram, class diagram and activity diagram.

3.2.1 Use case Diagram

Use case diagrams provide a clear picture of how users will interact with the system to achieve a particular goal [8]. Details of the interaction between the user and the system can be seen visually in the use case diagram in Figure 5.

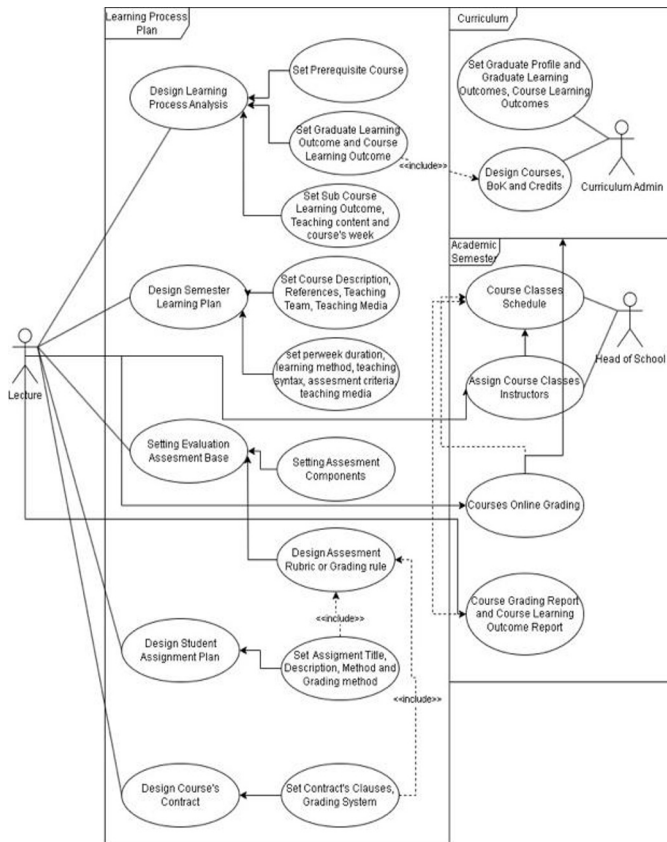


Fig 5. Use case Diagram of OBE curriculum management information system.

The use case diagram in Figure 5 describes the interaction between the lecturer, curriculum admin, and head of school actors who are users and are tasked with managing the OBE-based curriculum in the system that will be built later.

3.2.2 Activity Diagram

Activity diagrams visualize the flow of work from one activity to another in a system [8]. Details of the system's work flow, from the initial stage to the final stage, can be seen in the activity diagram in Figure 6 below.

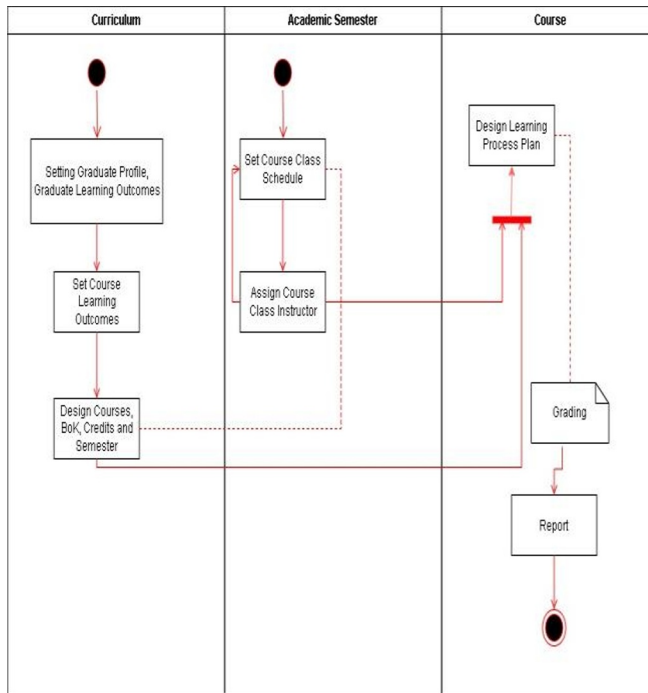


Fig 6. Activity Diagram of OBE curriculum management information system.

Figure 6 describes the activities in the system to be built with several process stages, including:

- **Start:** The process begins by determining the graduate profile and the learning outcomes to be achieved. This is an important initial step in designing a relevant curriculum.
- **Curriculum Design:** Once the graduate profile is determined, the next step is to design the curriculum. This involves designing courses, determining credit weights, and determining the semester in which the course will be offered.
- **Create a Class Schedule:** Once the curriculum is designed, a class schedule can be created. This involves determining the time, classroom, and lecturers who will teach each course.
- **Lecturer Assignment:** Lecturers are then assigned to teach the scheduled courses.
- **Learning Process Design:** Each course will have a specific learning plan. This plan includes the material to be taught, teaching methods, and assessments.
- **Assessment:** After the learning process is complete, an assessment of student learning outcomes is carried out.
- **Report:** The assessment results are then summarized in the form of a report. This report can be used to evaluate the effectiveness of the learning process and as material for future improvements.
- **Completed:** The process ends after all stages are completed.

3.2.3 Class Diagram

A class diagram describes in detail the classes that will be used in the system, including the attributes and operations that each class has, as well as the relationships between the classes [8]. Figure 7 presents a class diagram model that shows how data will be stored and manipulated in the system.

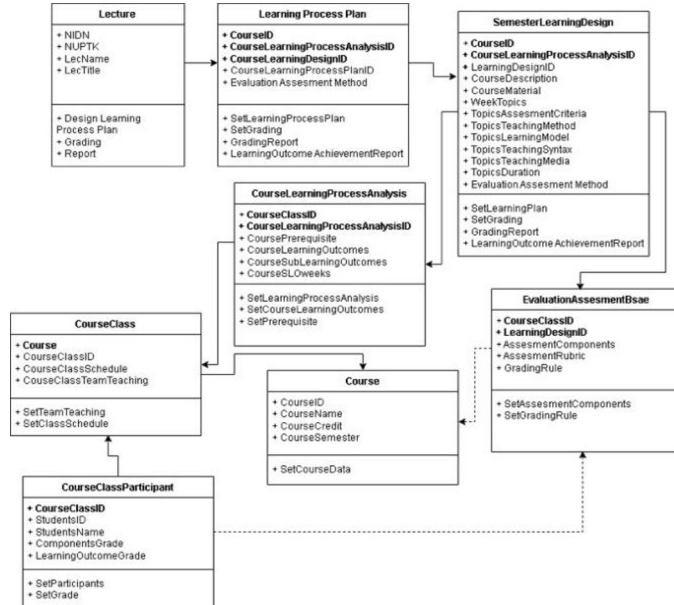


Fig 7. Activity Diagram of OBE curriculum management information system.

This diagram illustrates the structure of a complex curriculum management system. Some of the entities that are clearly visible are:

- Lecturer: Has basic information about the lecturer such as name, NUPTK, and course title.
- Course: Has information about the course such as code, name, semester, and organizer.
- Course Class: Is an instance of a course in a certain semester, with information such as schedule, lecturer, and participants.
- Class Participants, Represents students who take a course, with information such as component scores, final scores, and others.
- Learning Plan, Describes the learning plan for a course, including learning objectives, teaching methods, and assessments.
- Learning Process Analysis, Results of the analysis of the implementation of the learning plan.

In the analysis and design process, there is a mockup or user interface display of the system that will be built later as well as a visualization of the system so that it is easy to understand and as a basis for testing by stakeholders, programmers, and other sub-components. The following is a mockup of the proposed application for an OBE-based curriculum management information system.

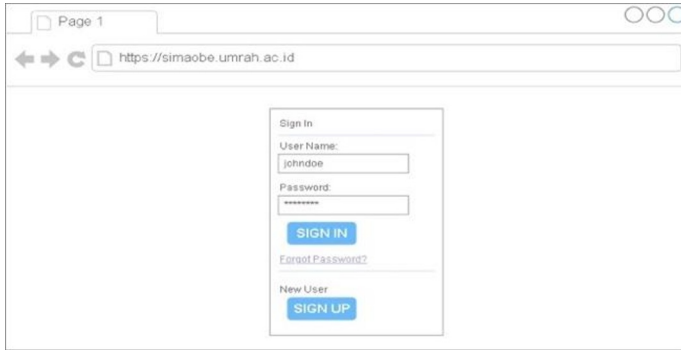


Fig 8. Mockup of user login page on OBE curriculum management information system.

The mockup in Figure 8. is designed to provide a simple and intuitive user experience, namely lecturer, curriculum admin, and head of school, making it easier for users to log in to the system. With clear and easily accessible elements, it is expected that users can quickly log in and start using the application.

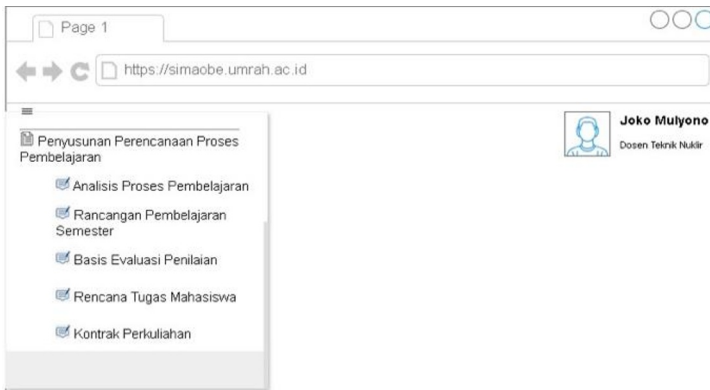


Fig 9. Mockup of the main page of the OBE curriculum management information system.

The mockup in Figure 9 is designed for users to use the system that has been created and is accompanied by the available menus.

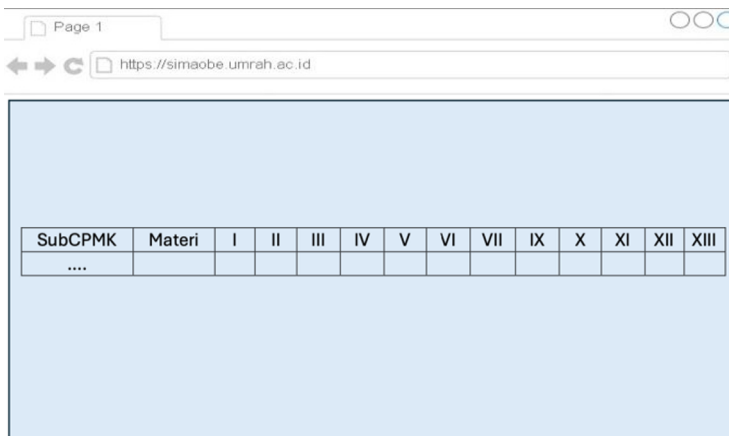


Fig 10. Mockup for learning analysis input form.

Mockup in Figure 10. Designed for lecturers to input learning analysis (a systematic evaluation process to understand how students learn and the factors that influence learning outcomes) for the courses they teach.

CAPAIAN PEMBELAJARAN	CPL-PRODI yang dibebankan pada Mata Kuliah	
	Kode CPL	Deskripsi CPL
	CPLxx
	CPLxx
	CPLxx
	Capaian Pembelajaran Mata Kuliah (CPMK)	
	CPMKxyx
	CPMKxyx
	CPMKxyx
	Sub Capaian Pembelajaran Mata Kuliah (Sub CPMK)	
	SubCPMKxyx
	SubCPMKxyx
	SubCPMKxyx
DESKRIPSI SINGKAT MATA KULIAH	

Fig 11. Mockup for Semester Learning Plan input form.

BAHAN KAJIAN DAN MATERI PEMBELAJARAN	Bahan Kajian:	
	1. BKxx
	2. BKxx
PUSTAKA	Materi:	
	1.
	2.
MATA KULIAH SYARAT	Utama:	
	• Author, title, publisher, year, ISBN	
	•	
MEDIA PEMBELAJARAN	Pendukung:	
	• Author, title, publisher, year, ISBN	
	•	
MATA KULIAH SYARAT	-	
	-	
MEDIA PEMBELAJARAN	Perangkat Lunak	Perangkat Keras
	•	•

Fig 12. Mockup for Semester Learning Plan input form (cont).

In the mockup Figure 11 and Figure 12 are designed for RPS input and the following points will be input, CPL code (drop down combo), CPMK Code (drop down combo), SubCPMK is filled in manually with both code and description, course description, study materials (BoK) from drop down combo, learning materials, main and supporting bibliography, required courses (list of courses from drop down combo), learning media is filled in by checking the options that have been set.

The screenshot shows a web browser window with the URL <https://simaobe.umrah.ac.id>. The page displays a form titled "A. RANCANGAN PEMBELAJARAN" (Learning Design). The form is a table with the following columns: No, Kemampuan Akhir yang di Sampaikan (Sub-CPMK), Indikator Pencapaian, Materi Pembelajaran/Praktik, Model Pembelajaran/Teori, Kegiatan Pembelajaran (Luring, Daring, Blended), Kriteria Penilaian, and Waktu. The table is mostly empty, with a few rows containing the text "Ujian Tengah Semester" and "Ujian Akhir Semester". At the bottom right, there is a field for "Total Jam pembelajaran per semester".

Fig 13. Mockup of learning design page in OBE curriculum management information system.

In the mockup Figure 13, the learning plan form will be inputted with the following points, Where inputted manually, achievement indicators (the week and SubCPMK columns are automatically filled in from the learning analysis), lecture material / learning main points, learning model and time allocation, offline, online and blended learning activities, assessment criteria, and length of time allocation in minutes (total amount of time in hours is automatically summarized at the bottom).

The screenshot shows a web browser window with the URL <https://simaobe.umrah.ac.id>. The page displays a form titled "RENCANA PEMBELAJARAN SEMESTER" (Semester Learning Plan). The form features the logo of Universitas Maritim Raja Ali Haji in the center. Below the logo, there are three fields for input: "MATA KULIAH : (huruf kapital)", "KODE MATAKULIAH : (huruf kapital)", and "SEMESTER : ... (angka romawi)".

Fig 14. Mockup for printing semester learning plan forms.

The screenshot shows a web browser window with the URL <https://simaobe.umrah.ac.id>. The page displays a detailed form for "RENCANA PEMBELAJARAN SEMESTER" (Semester Learning Plan) for Universitas Maritim Raja Ali Haji, Fakultas Teknik dan Teknologi Kemaritiman, Program Studi Teknik Informatika. The form includes the university logo, the name of the institution, and the program name. It also includes a field for "RIF-kodemk". The main part of the form is a table with the following columns: MATA KULIAH, KODE, RUMPUN MATA KULIAH, BOBOT (SKS), SEMESTER, and TANGGAL PENYUSUNAN. Below this table, there are fields for "DOSEN PENGEMBANG RPS", "DOSEN PENGAMPU MATA KULIAH" (with a list of three names), and "KETUA PRODI".

Fig 15. Mockup for printing semester learning plan forms (cont).

Table 2. Table of OBE academic management information system design test.

1.	Preparation	Objective	Testing the user interface of the OBE-based curriculum management information system design mock up to ensure the design is intuitive and easy to use.
		Participants	7 users who are also members of the development team and have contributed to the design of the system.
2.	Test design	Scenario	"Please use this app mock up and perform the following tasks. Create a new project. Add a curriculum matrix to the project. Create and mark completed tasks."
3.	Implementation of Testing	Time	1 Hour
		Step	Introduction (5 minutes): The facilitator explains the purpose of the test and provides context about the mock up.
			Mock-up Exploration (30 minutes): Participants explore the mock up, where the facilitator observes interactions and notes difficulties.
			The facilitator asks participants to explain their thinking, for example: "What do you do now?" or "What is confusing?"
			Feedback Discussion (15 minutes): The facilitator invites participants to share their experiences. Example question: "Is there anything you feel is unclear?"
Q&A (10 minutes): Participants can ask questions about the features they see.			
4.	Data Collection	Problem Notes	There were no records of problems from respondents.
		Positive Feedback	Clean interface and logical workflow, Participants felt comfortable because they were involved in the design process. Participants reported feeling comfortable using the mock-up because they were involved in the design process, which gave them a better understanding of the purpose and function of the system.
5.	Conclusion	Test accuracy	This test is considered accurate because all respondents are team members who actively contribute to the design of the system. They have a deeper understanding of the features and functions of the application, which allows them to provide more relevant and specific feedback. With in-depth knowledge of the design, respondents are better able to identify problems and provide useful recommendations for improvement.

4 Conclusion

The implementation of Model View Controller (MVC) architecture in the design of Curriculum Management Information System based on Outcome-Based Education (OBE) shows significant potential to improve the efficiency and effectiveness of curriculum management. Usability testing conducted ensures that the system meets user needs and provides a satisfying experience. Therefore, this system not only supports better implementation of OBE curriculum but also contributes to improving the quality of education at the university. In addition, this system can be adapted for other educational institutions, such as high schools and training institutions, with additional features such as learning outcome data analysis and e-learning platform integration. By establishing a clear framework using MVC architecture, this design not only serves as a technical blueprint but also as a foundation for achieving better educational goals through a robust and responsive information system, making it an effective tool for improving the overall quality of education.

References

1. A. Alip, S. Kosasi, I. D. A. E. Yuliani, G. Syarifudin, and D. David, *Jurnal Bumigora Information Technology* **3**, 135 (2022)
2. A. F. S. Wahyudi and D. Heksaputra, *INSERT* **4**, 86 (2023)
3. A. H. A. Rasyid, B. Yunitasari, I. W. Susila, D. Dewanto, Y. Yunus, and D. I. Santoso, *Jurnal Pendidikan (e-Journal)* **7**, 8 (2022)
4. A. Hitchcock, K. Sung, and N. Lee, Springer International Publishing (2024)
5. A. Subari, S. Manan, and E. Ariyanto, 1918, 042145 (2021)
6. D. I. Ekpo, O. E. Taylor, and D. Matthias, *International Journal of Computer Techniques* **10**, (2023)
7. E. D. C. Sihombing and S. R. Wahab, **5**, 152 (2021)

8. F. Magfira, Jurnal Elektronika, Listrik Dan Teknologi Informasi Terapan (ELTI) **4**, 44 (2022)
9. H. Asgar and B. Hartono, Jurnal Informatika Teknologi Dan Sains (Jinteks) **4**, 8 (2022)
10. M. G. Efgivia and A. Albahra, OUTCOME-BASED EDUCATION: TEORI, PRAKTIK, DAN APLIKASI DALAM PENDIDIKAN MODERN (2024)
11. M. Kurniawan, P. E. Gunawan, S. Kushanifah, S. D. Artanti, and R. Rusdita, JATI (Jurnal Mahasiswa Teknik Informatika) **8**, 3603 (2024)
12. N. P. E. Sulistyadewi, G. Sastrawangsa, and R. Wulandari, Technomedia Journal **8**, 34 (2023)
13. N. T. Saptadi, M. Alwi, G. Maulani, W. Novianti, M. Muhammadiyah, Y. Agustina, E. Susilawati, F. Sampe, T. H. Wardoyo, T. Riyadi, R. A. Hadikusumo, L. Nurlily, S. S. Evenddy, V. K. Ananingsih, and A. Holid, Revolusi Pendidikan: Merdeka Belajar Kampus Merdeka (MBKM) (2024)
14. R. Amhar and N. W. A. Majid, (2024)
15. S. Munir, W. Adidaya, E. Riansyah, and H. Sasmita, Jurnal Informatika Terpadu **2**, (2016)
16. S. Siswidiyanto, A. Munif, D. Wijayanti, and E. Haryadi, **15**, 18 (2020)
17. W. S. Ningsih and H. Nurfauziah, Jurnal Ilmiah Metadata (2023)
18. Y. Firmansyah, R. Maulana, and M. S. Maulana, (2021)