

The Development of UMS Building Catalogue Information System

Ganno Tribuana Kurniaji¹, Mohammad Faqih Eza ‘Ammar², Yusuf Sulisty Nugroho³

^{1,2,3}Teknik Informatika, Universitas Muhammadiyah Surakarta, Surakarta, Indonesia

¹l200184092@student.ums.ac.id, ²l200183178@student.ums.ac.id, ³yusuf.nugroho@ums.ac.id

Accepted 23 March 2022

Approved 10 May 2022

Abstract— Universitas Muhammadiyah Surakarta (UMS) has a large number of new students per year. However, most new students often get confused to identify and even do not know the location of the buildings that handle various activities in the early period of lectures, such as student orientation, English pre-tests, etc. This is due to lack of information provided by the university to the new enrolled students. To overcome this problem, this work aims to develop a web-based building catalogue by implementing a framework, namely Ionic. This system is built to help the new students getting the building-related information in UMS and find out the location of the building. By following the waterfall model to build the system, the results show that we have successfully developed the building catalogue information system. The acceptance testing result shows that the system is acceptable to implement, supported by the SUS score of 76.17. The system is helpful for new students to get information of all buildings in UMS.

Index Terms— building; catalogue; information system; UMS.

I. INTRODUCTION

Universitas Muhammadiyah Surakarta (UMS) is one of the largest private university in Indonesia. It has 12 faculties of Undergraduate Programs, Graduate Schools, Professional Programs, and Vocational Schools consisting of various disciplines, such as advancements in science, technology and social relations, and arts and culture. To support the business processes, UMS has 5 campuses that are integrated with IT systems, it has a land area of more than 40 hectares, and supported by an Edutorium that covers an area of 6.5 hectares which is considered as the largest convention center in Central Java. The implementation of the lectures in UMS is supported by very adequate facilities such as representative study rooms, complete laboratories, IT-based libraries, and other supporting facilities.

Each year, UMS receives around 8,000 new students that are distributed to 60 study programs. At the beginning of the lecture period, the new students are required to complete various activities such as student orientation activities, English pre-tests, etc. However, The students often get confused and even they only know the name of the buildings without knowing their

locations where the activities will be held [1]. This is due to lack of information provided by the university for new students regarding the list of buildings in UMS.

Many studies have been carried out to develop similar applications. A research [2] explains that the virtual tour application provides early information as an earlier experience for students in terms of introducing the area of the new environment. Other study [3] developed a virtual tour as an effective means of introducing new academic environment to the wider community and prospective students. A web-based virtual tour application has also been developed by prior study [4] to display the campus atmosphere and provide information on the location of the building.

Due to the benefit of the application to overcome buildings introduction problems faced by UMS, we developed a similar web-based catalogue building app to help the new enrolled students in identifying the buildings in the university [5]. Catalogue is a medium that is able to provide optimal and easy-to-understand information, contains more detailed and complete information than other media packaged in the form of photographs and attractive layouts [6]. To build the catalogue information system, we implement the Ionic framework [7]. Ionic is a platform to design and build specific applications. It has its own development tools but is very time consuming and costly [8]. On the other hand, Ionic is an independent platform that uses HTML5, CSS and Ionic's JavaScript. The framework is built with AngularJS, which is arguably the most tested and widely used JavaScript framework out there [9].

We developed a web-based building catalogue to help the students in getting the information of UMS buildings [10]. With this building catalogue, the university can easily provide information and locations of the buildings in UMS to the new enrolled students. The design of this application makes it easier for new students to recognize the buildings in UMS.

II. METHODOLOGY

In designing the UMS building catalogue information system, we follow the Software Development Life Cycle (SDLC) approach, specifically the waterfall model [11]. Waterfall is the

oldest and most well-known model in SDLC. A special feature of this model is its sequential steps. It goes down through the phases of requirements analysis, design, coding, testing, and maintenance [12]. This model works well for projects where quality control is a major

concern due to its intensive documentation and planning [13]. Each stage in this model is not overlapping the other stages, that means the process in the waterfall model starts and ends in one stage before starting the next stage [14].

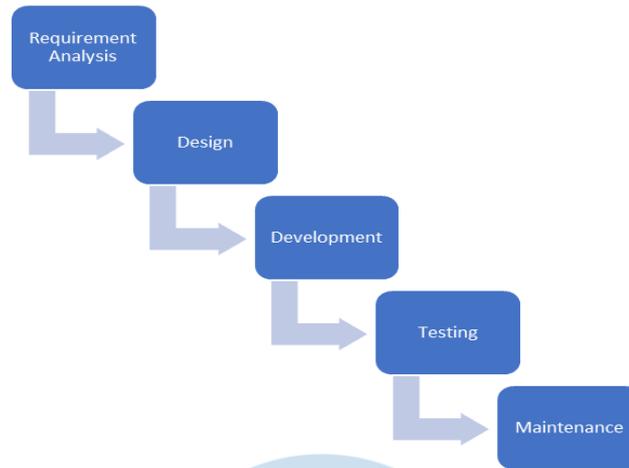


Fig. 1. The process flow of waterfall model in SDLC

Figure 1 shows that there are five stages in the waterfall model that we follow to develop the UMS building catalogue information system [15]. The details of each stage are as follows:

A. Requirement Analysis

Requirement analysis is a description of the behavior of the system to be developed [16]. Usually, it provides the information of requirements from clients. Hence, it establishes an agreement between the client and the developer for the specifications and features of the software. In short, requirements are collected, analyzed, and the proper documentation is prepared to help the development process.

The initial discussion was carried out to extract the main problems faced by the clients. The results of the discussion indicate that a web-based building catalogue information system is necessary. In the discussion, we also observe the requirements analysis, which is categorized into two types, namely (1) functional, and (2) non-functional requirements [17].

The functional requirement analysis describes that the building catalogue system is developed online. The system is able to provide detailed and complete information of the building. The catalogue system does not require a “user sign in” to access the information. The users of the system are classified into two types, that is administrator and common users [18]. An administrator is authenticated to have the whole access to the system, so that it can perform the CRUD (Create, Read, Update, and Delete) transaction in the system [19]. Common users do not have rights to update the content in the catalogue. They can only search and read the provided information.

For the non-functional requirements, we analyze the performance and scalability, portability and compatibility, reliability, availability, maintainability, and the usability, as described as follows:

1. Performance and Scalability

One of the steps of the text pre-processing stage which serves to convert all letters in the document to lowercase. This step is done to make the search easier.

2. Portability and Compatibility

The system should be able to run on a computer device that runs on Windows 7 version or newer, 1.8 GHz or faster processor, 2 GB of ram or higher, and video card that supports a minimum display resolution of 720p.

3. Reliability, Availability, Maintainability

There are 75% chances that the component can be fixed in 6 hours for the maintainability of this system. The web dashboard must be available to users 98% of the time every month during business hours.

4. Localization

The system is intended for the new enrolled students since they are not familiar with the buildings in UMS. The language of the system is provided in Bahasa Indonesia, so that it is easy to understand for the students.

5. Usability

The building catalogue system is simple and easy to use. So that the users can quickly access the catalogue to get such information. In this system, the users do not need to login to access the catalogue system. Although

the main target of the system users is the new enrolled students in UMS, however, all academic staff can also access the catalogue system.

B. Design

The information gathered from the previous phase is evaluated and the proper implementation is formulated. The design describes the key components and the interface of the components. It focuses on how to deliver the required functionality (must-haves and wish-list items) to the system [20].

In this step, we initially design the use case diagram of the system, as shown in Figure 2. Use case diagram is a methodology used in system analysis to identify, clarify, and organise system requirements. Use case diagrams are employed in UML (Unified Modelling Language), a standard notation for the modelling of real-world objects and systems [21]. It shows how a system interacts with external entities. So, it is relatively sparse about the details of how the system behaves internally and how the external environment is arranged.

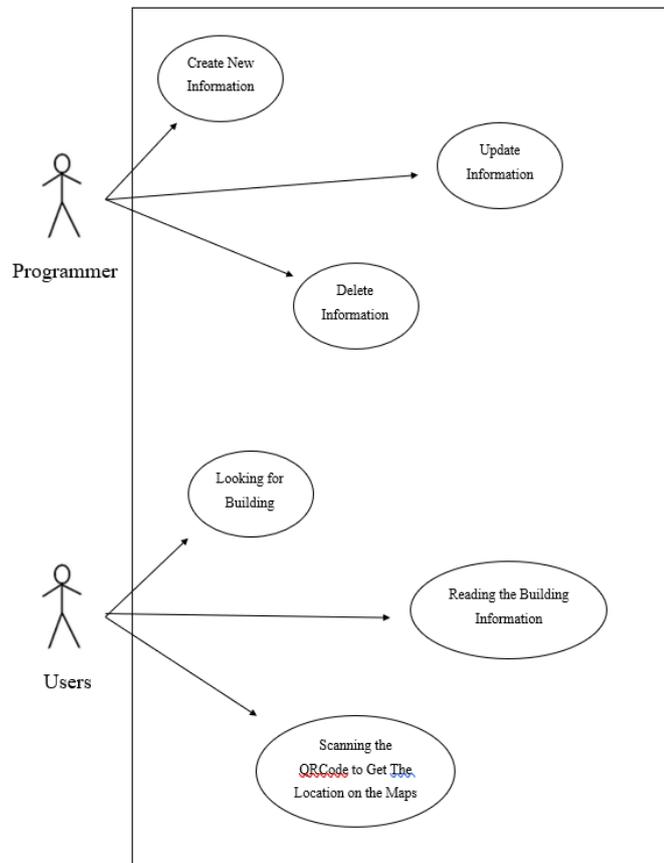


Fig. 2. Use case diagram of building catalogue information system

Figure 2 explains that there are 2 actors who can access the system, namely Administrator and Users. The administrator can change or add new information. On the users side, users can only look for buildings, reading the building information, and directed to the location of the building on the map.

The activity performed by the actors indicated in Figure 2, is then illustrated using an Activity Diagram, as shown in Figure 3. Activity Diagrams are behavioural diagrams which depict the internal behaviour of different operations of a program with the help of nodes and edges [22]. Activity Diagram is used to represent different activities, sub activities, transitions, decision points, guard conditions, concurrent activities, branch, merge, swim lanes, join

forks, and etc. An activity diagram starts with one start activity and ends at one final activity.

In addition, this building catalogue system was not developed by implementing a database. This is because the information needed to be displayed on the system is simple and does not require data storage. Therefore, we do not include sequence diagrams and class diagrams in this discussion.

Figure 3 shows that the administrator can create, add, or update the detailed information in the catalogue system which will be sent to the server. The users who access the system via a browser will see the content of the system and be able to select the building based on the provided list. Once the users select one of the building lists, the server will send the detailed information and location of the building to the browser.

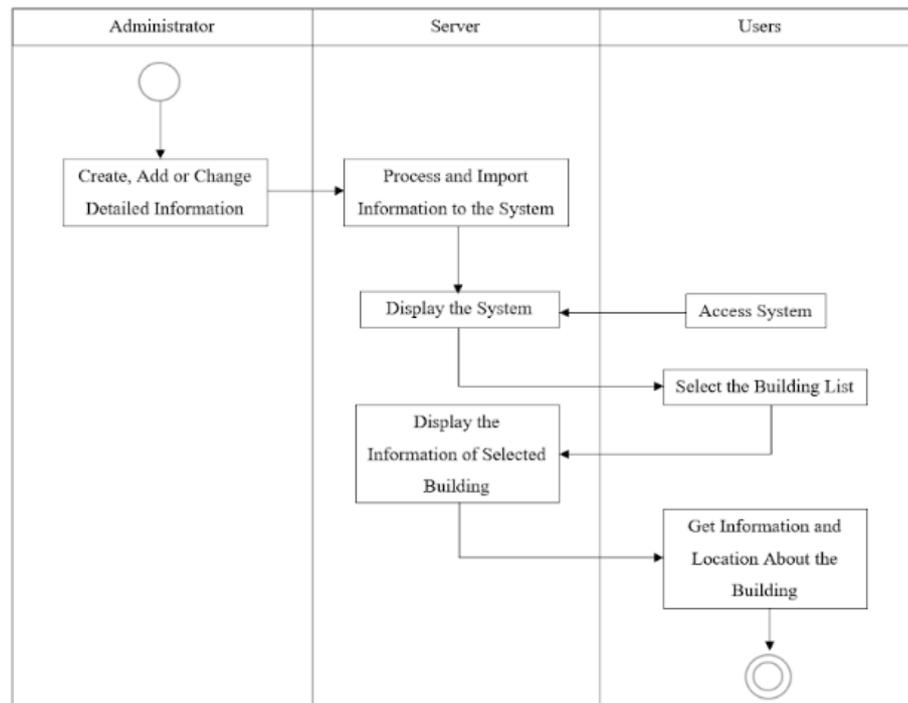


Fig. 3. Activity diagram of building catalogue information system

C. Development

In this phase, the whole requirements will be converted to the production environment. We designed the UMS building catalogue system by applying the Ionic framework. The Ionic framework can be run by installing the Node.js. Node.js is a platform built on top of the Chrome JavaScript runtime to easily build fast and scalable network applications. Node.js uses a non-blocking I/O model that makes it lightweight and efficient, perfect for real-time data-intensive applications running across distributed devices [23]. We are also required to install Ionic CLI using "npm". The Ionic CLI is used to scaffold, develop, and run Ionic applications [24].

D. Testing

This phase deals with the actual testing and inspection of the software solution that has been developed to meet the original requirements. Also, this is the phase where system bugs and glitches are discovered, fixed and refined [25]. In this step, we perform a black-box testing to validate the system and ensure that the system is running as expected. Black-box testing is a functional testing technique that designs test cases based on the information from the specification. Black box testing is not concerned with the internal mechanisms of a system. These are focused solely on the outputs generated in response to selected inputs and execution conditions [26].

Through the black-box testing, we test the building catalogue system by involving one of the representatives from the UMS Information Technology

Bureau to ensure that there are no errors in the system and satisfy all the requirements needed [27]. Various checks on the main menu page are performed, starting from checking the image of the building, sorting the name of the building, and checking all features in the systems. On the building menu page, we validate all information listed, starting from the name of the faculty that occupies the building, the list of study programs in the building, information of the student organisation units in the building, and all related information. In addition, we check the general information of the building, such as address, telephone number, and email, as well as the barcode of the building's location.

To evaluate the system acceptance, we apply the testing using System Usability Scale (SUS). SUS is a Likert scale that is simply one based on forced choice questions, where a statement is made and the respondent then indicates a degree of agreement or disagreement with the statement on a 5 (or 7) point scale. The statements with which the respondent expresses agreement and disagreement should be chosen carefully [29].

As shown in Figure 4, it can be seen that the 10 statements in SUS testing actually covers various aspects of system usability, such as the need for support, training, and complexity, and thus has a high degree of facial validity for measuring the usability of a system.

	Strongly disagree						Strongly agree
1. I think that I would like to use this system frequently	1	2	3	4	5		
2. I found the system unnecessarily complex	1	2	3	4	5		
3. I thought the system was easy to use	1	2	3	4	5		
4. I think that I would need the support of a technical person to be able to use this system	1	2	3	4	5		
5. I found the various functions in this system were well integrated	1	2	3	4	5		
6. I thought there was too much inconsistency in this system	1	2	3	4	5		
7. I would imagine that most people would learn to use this system very quickly	1	2	3	4	5		
8. I found the system very cumbersome to use	1	2	3	4	5		
9. I felt very confident using the system	1	2	3	4	5		
10. I needed to learn a lot of things before I could get going with this system	1	2	3	4	5		

Fig. 4. The System Usability Scale chosen statement

To calculate the SUS score, we first add up the contribution of each item's score. The contribution score of each item will range from 0 to 4. For items 1,3,5,7, and 9 the contribution score is the scale position minus 1. For items 2,4,6,8 and 10 the contribution is 5 minus the scale position in this case we get a total score

of 27,420 from 360 respondents. We take data as many as 360 respondents based on the calculation of a representative sample from around 8,000 new students using a survey calculator with the confidence level of 95% and the interval of 5. The calculation of the sample size produces 360 respondents.

E. Maintenance

Maintenance is the process of taking care of such concerns [24]. In this step, we release the building catalogue system. Although it has been implemented, it is possible that the system needs updates, improvements, errors fixing, and refinement depending on the user's requirements in the future.

III. RESULT AND DISCUSSION

The web-based building catalogue system has been successfully implemented. It contains 2 menus, namely Main Menu Page and Building Menu Page. The main menu page functions as the starting page, which contains a list of the names of the buildings accompanied by pictures. On this main menu page, the user first needs to select which building he wants to look for to view detailed information in it. On the building menu page, the function is to display all available information regarding the selected building on the main menu page. On the building menu page there are various kinds of complete information and also accompanied by complementary images and barcodes indicating the location of the building.

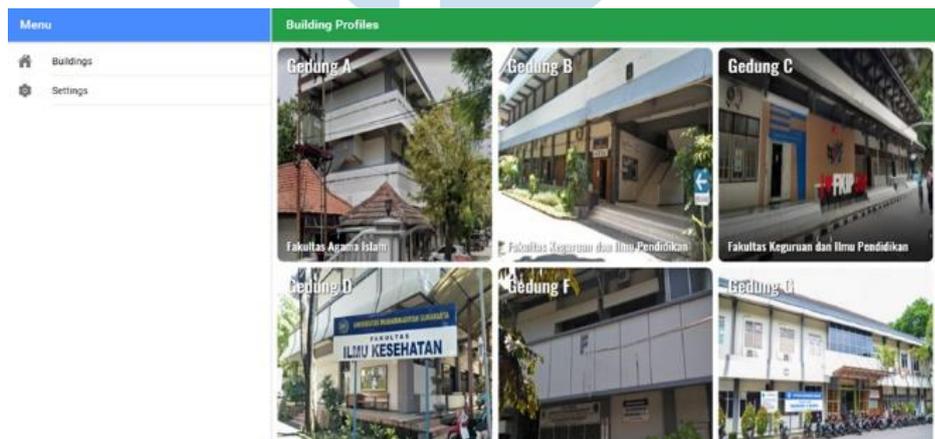


Fig. 5. The main page of the building catalogue information system

In Figure 5, The main page menu displays a list of buildings that are within the scope of UMS, starting from building A to building L. On the main page, there are also pictures of each building. In the bottom left corner of the image, the name of the faculty where the

building is located is listed to clarify the description of the building. Each column in the list of buildings on the main page can be clicked to see the more detailed information.

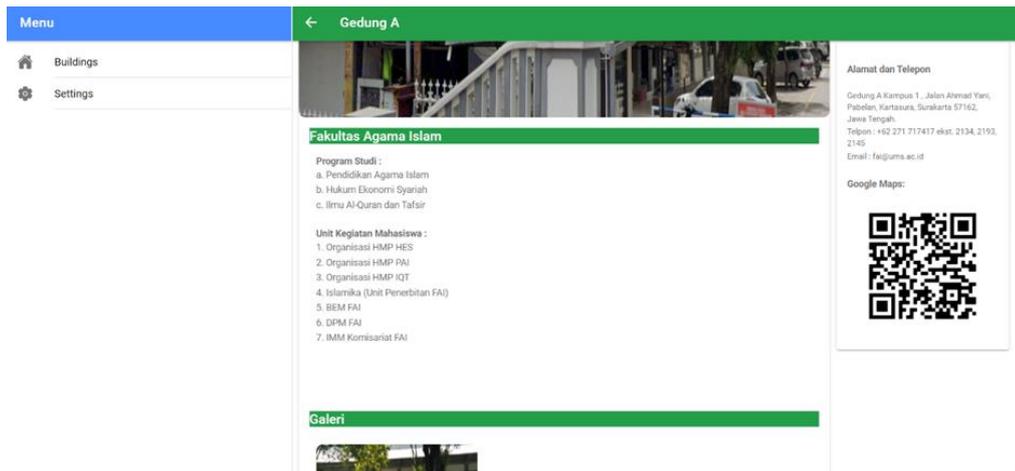


Fig. 6. Detail information of the building menu page

On the building menu, as shown in Figure 6, there is a lot of detailed information about the building such as the photos of the buildings, the study programs located in the buildings, the student organization units located in the buildings, the building addresses, telephone numbers, emails, and barcodes. In general, each building has more than one various study programs. Thus, it is necessary to mention the study programs in the building on the page. It also provides the information of student organization units that occupy the buildings to help the new students be more familiar with the description of each organization.

On the same page, the system displays a gallery of the building photos to show the actual image of the building. We also include detailed information of the building, such as the address where the building is located, the phone number of the faculty, and the email of the faculty. This information can be used for students to contact the associated staff to get further information that is not available in the system. A QR code is added to the page in the system to store additional information of the buildings, such as the location guidance that leads to the building. If the QR code is scanned, it will automatically be redirected to the Google Maps application and will show the exact location.

The building catalogue system is tested using a Blackbox technique [28] to ensure that there are no errors in the system and the System Usability Scale (SUS) to ensure that the system is well developed and easy to use. For the Blackbox technique, we test the system through 8 scenarios, as described in Table 1.

TABLE I. BLACKBOX TEST SCENARIOS

No	Test Scenario	Expected results	Test result
1	Check whether the image of the building matches the name of the building	The image of the building matches the name of the building	Valid
2	Check whether the name	The name of the	Valid

	of the building is in order	building is in order	
3	Check if there are a typo in the building name	There are no typo in the building name	Valid
4	Check the name of the Faculty that occupies the building whether is correct or not	The name of the faculty is correct	Valid
5	Check the list of Study Programs in the building whether it is complete and correct	The list of Study Programs in the building is complete and correct	Valid
6	Check the list of Student Organisation Units in the building whether it is complete and correct	The list of Student Organisation Units in the building is complete and correct	Valid
7	Check whether address, telephone number, and email are correct	The address, telephone number, and email are correct	Valid
8	Check the barcode whether it gives correct location to the building or not	The barcode gives correct location to the building	Valid

Table 1 describes that all scenarios in the Blackbox testing are running well. We perform various checks on the main menu page, including the image of the building that matches the name of the building, sorting the name of the building, and checking the spelling of the building names. On the building menu page, we check all the information listed, including the name of the faculty that is occupied in the building, the list of the study programs in the building, information of the student organization units in the building, and the related information.

Furthermore, the implementation of SUS testing produced a total score of 27,420 from all respondents. We then divided the total score by 360 of total respondents to get the overall system usability score. The calculation yields a SUS score of 76.17 which means that the application is acceptable and usable for the new students to get familiar with the new environment in UMS.

IV. CONCLUSION

Based on the results of the study, it can be concluded that we have succeeded in developing a web-based building catalog to assist students in obtaining information about UMS buildings. With this building catalog system, the University can easily provide information and locations about buildings at UMS to new students. Based on the SUS score, the system can be implemented to introduce the buildings in UMS to the new enrolled students in the future. This study also may open an opportunity to develop a mobile-based application in the future.

ACKNOWLEDGMENT

This work has been successfully conducted with the help of the Information Technology Bureau of Universitas Muhammadiyah Surakarta.

REFERENCES

- [1] R. B. N. Budi and M. B. Wenas, "Perancangan Virtual Tour Kampus IUKSW sebagai Media Informasi Mahasiswa Baru," *ANDHARUPA J. Desain Komun. Vis. Multimed.*, vol. 4, no. 01, pp. 51–65, 2018.
- [2] F. Annafi, "Pengembangan Aplikasi Virtual Tour Berbantuan Video sebagai Media Informasi Wilayah Fakultas Teknik Universitas Negeri Yogyakarta," *Yogyakarta Univ. Negeri Yogyakarta*, 2014.
- [3] A. Z. Z. Falani, H. H. A. S. Ramadan, and E. S. Setiawan, "Implementasi Sistem Virtual Tour Berbasis E-Panorama Untuk Pengenalan Kampus Universitas Narotama Surabaya," *Insa. Comtech Inf. Sci. Comput. Technol. J.*, vol. 1, no. 1, 2016.
- [4] A. Choiron, "Aplikasi Virtual Tour Dinamis Pada Universitas Dr. Soetomo Surabaya Berbasis Web," *Inf. J. Ilm. Bid. Teknol. Inf. dan Komun.*, vol. 2, no. 1, 2017.
- [5] T. Bayer, D. P. Ames, and T. G. Cleveland, "Design and development of a web-based EPANET model catalogue and execution environment," *Ann. GIS*, vol. 27, no. 3, pp. 247–260, 2021.
- [6] H. A. REVIL HUDRIANI and M. S. D. Ariusmedi, "Perancangan Katalog Panduan Wisata Seribu Rumah Gadang di Kabupaten Solok Selatan," *DEKAVE J. Desain Komun. Vis.*, vol. 8, no. 3, 2019.
- [7] S. Alian, J. Li, and V. Pandey, "A personalized recommendation system to support diabetes self-management for American Indians," *IEEE Access*, vol. 6, pp. 73041–73051, 2018.
- [8] S. Yusuf, *Ionic Framework By Example*. Packt Publishing Ltd, 2016.
- [9] J. Waranashiwar and M. Ukey, "Ionic Framework with Angular for Hybrid App Development," *Int. J. New Technol. Res.*, vol. 4, no. 5, p. 263068, 2018.
- [10] B. Safuan, R. K. Arief, and E. Y. T. Adesta, "Web based e-catalog implementation at TPK-KOJA: Case study of stock inventory division," *Int. J. Adv. Sci. Technol.*, vol. 29, no. 7, pp. 481–488, 2020.
- [11] W. W. Royce, "Managing the development of large software systems: concepts and techniques," in *Proceedings of the 9th international conference on Software Engineering*, 1987, pp. 328–338.
- [12] S. Balaji and M. S. Murugaiyan, "Waterfall vs. V-Model vs. Agile: A comparative study on SDLC," *Int. J. Inf. Technol. Bus. Manag.*, vol. 2, no. 1, pp. 26–30, 2012.
- [13] Y. Bassil, "A simulation model for the waterfall software development life cycle," *arXiv Prepr. arXiv1205.6904*, 2012.
- [14] A. Alshamrani and A. Bahattab, "A comparison between three SDLC models waterfall model, spiral model, and Incremental/Iterative model," *Int. J. Comput. Sci. Issues*, vol. 12, no. 1, p. 106, 2015.
- [15] W. Hardyanto, A. Purwinarko, F. Sujito, and D. Alighiri, "Applying an MVC framework for the system development life cycle with waterfall model extended," in *Journal of Physics: Conference Series*, 2017, vol. 824, no. 1, p. 12007.
- [16] U. Udi, "Penerapan Metode SDLC Waterfall Dalam Pembuatan Sistem Informasi Akademik Berbasis Web Studi Kasus Pondok Pesantren Al-Habib Sholeh Kabupaten Kubu Raya, Kalimantan Barat," *J. Teknol. Dan Manaj. Inform.*, vol. 4, no. 1, 2018.
- [17] P. Becker, G. Tebes, D. Peppino, and L. Olsina, "Applying an Improving Strategy that embeds Functional and Non-Functional Requirements Concepts," *J. Comput. Sci. Technol.*, vol. 19, no. 2, pp. e15–e15, 2019.
- [18] N. Farooqi, S. Alshehri, S. Nollily, L. Najmi, G. Alqurashi, and A. Alrashedi, "UParking: Developing a smart parking management system using the internet of things," in *2019 Sixth HCT Information Technology Trends (ITT)*, 2019, pp. 214–218.
- [19] K. Košťál, P. Helebrandt, M. Belluš, M. Ries, and I. Kotuliak, "Management and monitoring of IoT devices using blockchain," *Sensors*, vol. 19, no. 4, p. 856, 2019.
- [20] M. Kramer, "Best practices in systems development lifecycle: An analyses based on the waterfall model," *Rev. Bus. Financ. Stud.*, vol. 9, no. 1, pp. 77–84, 2018.
- [21] A. Y. Aleryani, "Comparative study between data flow diagram and use case diagram," *Int. J. Sci. Res. Publ.*, vol. 6, no. 3, pp. 124–126, 2016.
- [22] M. Touseef, N. A. Butt, A. Hussain, and A. Nadeem, "Testing from UML design using activity diagram: a comparison of techniques," *Int. J. Comput. Appl.*, vol. 975, p. 8887, 2015.
- [23] M. Cantelon, M. Harter, T. J. Holowaychuk, and N. Rajlich, *Node.js in Action*. Manning Greenwich, 2014.
- [24] A. Ravulavaru, *Learning ionic*. Packt Publishing Ltd, 2015.
- [25] H. Nur, "Penggunaan Metode Waterfall Dalam Rancang Bangun Sistem Informasi Penjualan," *Gener. J.*, vol. 3, no. 1, pp. 1–10, 2019.
- [26] S. Nidhra and J. Dondeti, "Black box and white box testing techniques-a literature review," *Int. J. Embed. Syst. Appl.*, vol. 2, no. 2, pp. 29–50, 2012.
- [27] I. R. Munthe, B. H. Rambe, R. Pane, D. Irmayani, and M. Nasution, "UML Modeling and Black Box Testing Methods in the School Payment Information System," *J. Mantik*, vol. 4, no. 3, pp. 1634–1640, 2020.
- [28] B. Beizer, *Black-box testing: techniques for functional testing of software and systems*. John Wiley & Sons, Inc., 1995.
- [29] J. Brooke, "Sus: a 'quick and dirty' usability," *Usability Eval. Ind.*, vol. 189, no. 3, 1996.