ISSN 2320 - 2602

Volume 12 No. 3, March 2023

International Journal of Advances in Computer Science and Technology

Available Online at http://www.warse.org/IJACST/static/pdf/file/ijacst011232023.pdf

https://doi.org/10.30534/ijacst/2023/011232023

Developing an E-report System for Complaint Management at Catholic University of St. Thomas using Prototyping Method

Andy P Harianja[1], Aldo A Moloan[2]

[1]Information Engineering Study Program, Universitas Katolik Santo Thomas, Indonesia, apharianja@gmail.com

[2]Information System Study Program, Universitas Katolik Santo Thomas, Indonesia, aldosirait8@gmail.com

Received Date : January 15, 2023 Accepted Date : February 22, 2023 Published Date : March 07, 2023

ABSTRACT

This study aimed to develop an E-report System for Complaint Management at Catholic University of St. Thomas using the Prototyping Method. The system includes a mobile Android app, enabling users to submit and track complaints on-the-go. The system's four phases are requirements gathering, design, implementation, and testing. The system provides a user-friendly interface for submitting and tracking complaints, features a dashboard for administrators, and offers push notifications. The system's four phases are requirements gathering, design, implementation, and testing. The system offers a user-friendly interface for submitting and tracking complaints, a dashboard for administrators, and push notifications. The study used a structured and iterative approach, resulting in a cost-effective solution that meets stakeholder needs. The E-report System is an effective and efficient solution for managing complaints in the education sector.

Key words: E-report System, Prototyping Method, Complaint Management, Mobile App.

1. INTRODUCTION

Over the past few decades, technological advancements have transformed the way organizations operate and interact with their stakeholders [1]. The education sector is no exception, with universities and schools using technology to enhance their administrative processes and services. One of the critical areas of focus in the education sector is managing complaints raised by students, staff, and faculty members [2].

Effective complaint management is essential to maintain a positive reputation, foster a healthy work and study environment, and address grievances in a timely and efficient manner [3]. This study aims to develop an E-report System for Complaint Management at Catholic University of St. Thomas using the Prototyping Method. The system includes a mobile Android app that enables users to submit and track complaints on-the-go, which is especially useful for students and faculty members who may not have access to a desktop computer.

The study's primary objective is to provide a cost-effective solution that meets the needs of stakeholders and offers a user-friendly interface for submitting and tracking complaints. The Prototyping Method provides a structured and iterative approach to system development, allowing for timely feedback and a more efficient solution. The four phases of the system's development include requirements gathering, design, implementation, and testing [4].

The E-report System for Complaint Management offers a dashboard for administrators to manage and address complaints, generate reports, and perform data analysis. The mobile Android app provides users with the flexibility to submit and track complaints on-the-go, view updates on the status of their complaints, and receive push notifications.

This study's contributions lie in providing an effective and efficient solution for managing complaints in the education sector while providing a user-friendly and cost-effective system that meets stakeholder needs. The study used a structured and iterative approach, resulting in a system that meets the required specifications and offers a modern and reliable solution for managing complaints.

2. METHODS

The research and development of the E-report application at Catholic University of St. Thomas utilized a combination of methodologies including literature review, observation, Rapid Application Development (RAD), and Unified Modeling Language (UML). The research stages in writing this research:

A. *Data Collection*

Data collection techniques consisted of:

1. Literature Review

   A thorough literature review of relevant materials such as books, journals, articles, and online sources.

2. Observation

Conducted directly at the Catholic University of St. Thomas location to identify the needs for the application design and validate data accuracy.

3. Requirement Design

The requirement design phase involved defining the necessary requirements for the application design, including devices used, the database, and the team involved in the design process.

4. System Analysis and Design

Evaluating the requirements gathered from the previous phase and transforming them into a prototype. This phase included designing the interaction of system functions and objects, creating schemas and databases, and designing the user interface.

5. Implementation.

In the implementation phase, the design was transformed into code using multiple programming languages to create an application system that could be operated.

6. Report Writing

The final report was written detailing the methodology, theory, and findings of the research.

B. System Development Methodology

The prototyping methodology is a software development approach used to efficiently address user needs. The customers often find it challenging to identify detailed requirements or needs for the functions and features of the software they want to develop [5]. Consequently, software developers may not have a high degree of confidence in the efficiency of algorithms or their ability to adapt to the operating system. To overcome these uncertainties, prototyping is used to develop a working model of the software, which helps customers and stakeholders to better understand the final product [6]. The stages of the prototyping methodology, as illustrated in Figure 1.

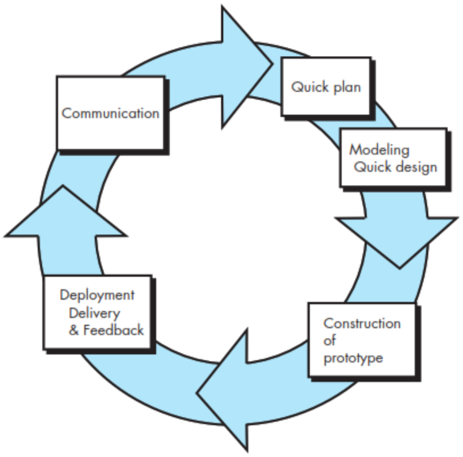


Figure 1: The Stages of Prototyping Method (source: Software Engineering: A Practitioner’s Approach, 2015)

C. Application Modeling Tools

In order to provide a clear understanding of the application system design using UML (Unified Modeling Language) modeling, which facilitates the visualization of application design in a software [7].

By using UML modeling, the application design can be represented in a graphical form that includes different views, such as use case diagrams, class diagrams, sequence diagrams, activity diagrams, and others. Each view provides a different perspective of the application design, which allows developers to identify and address potential issues and improve the design [8].

3. RESULT AND DISCUSSION

A. Communication

This stage starts with the collection of data related to the needs for application development. This is supported by the dissemination of questionnaires to several academic communities in the form of surveys of application needs and requirements. The questionnaire distributed is loaded into Google Form which is then accessed by several samples within the campus environment.

B. Quick Plan

In this stage, based on the system analysis conducted to build an understanding of the problems faced, the system to be developed, and the system requirements analysis that have been included in the Communication stage. The system requirements are designed using a use case diagram that can provide a complete picture of the interactions that occur between actors and the developed system. Two actors are involved in the system development: the administrator and the regular user. Next, the proposed system details are completed, including data definition descriptions and interaction analysis with the process. The results will be created using Figma as the interface design tool in the prototype. The research findings that will be developed into the system include the database design, activity design, and interface design. The Quick Plan is classified into:

1. Administrator

The admin is responsible for managing reports and user data such as handling report progress, deleting reports, adding users, and so on. This page can only be accessed when the actor has logged in as an admin. The admin use case diagram can be seen in Figure 2 below.

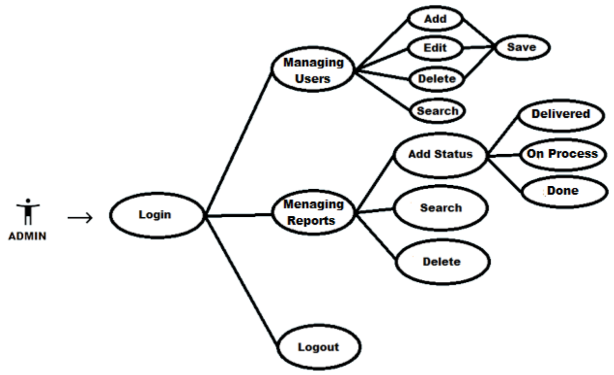


Figure 2: Use Case Admin

2. Regular User

The user is a member of the academic community of Saint Thomas Catholic University, including students, lecturers, and staff. They have access to the main page of the application, which can be seen in Figure 3.

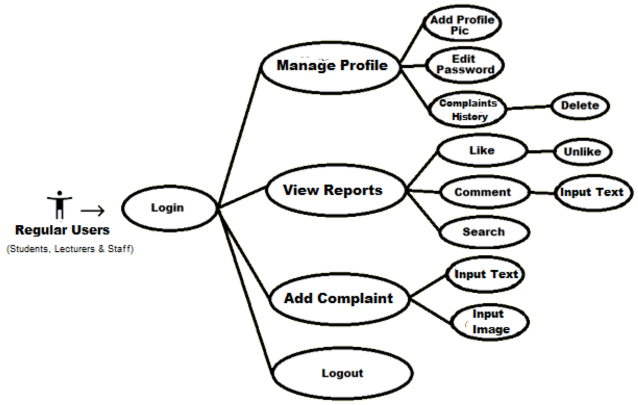


Figure 3: Use Case Regular Users

C. *Modeling Quick Design*

Modeling Quick Design focuses on representing the software aspects that will be visible to users. The prototype design is classified into:

1. Login Page

This page is the first display that users can see when opening the application, which shows a form consisting of a username and password. The design of the login page can be seen in Figure 4.

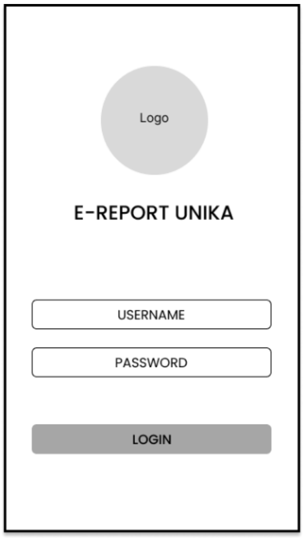


Figure 4: Login Page Application

2. Admin Home Page

This page is divided into two parts, namely a display for managing existing reports and managing users. The admin home page can be seen as shown in Figure 5.

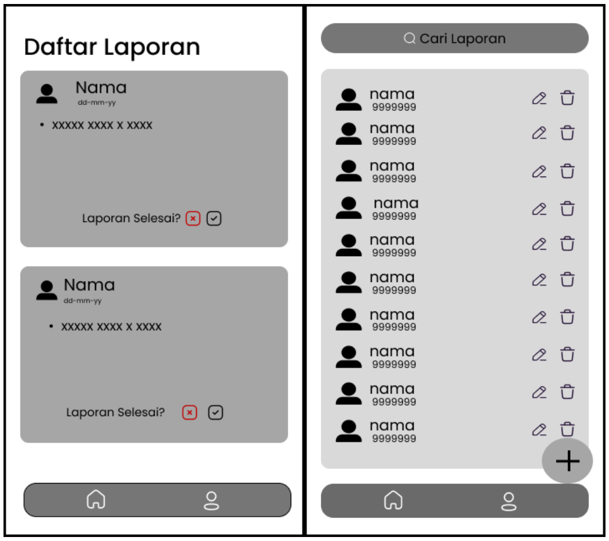


Figure 5: Admin Home Page Application

3. Regular User Home Page

This page is also divided into two parts, namely a display for viewing existing reports and manage profile page. There are various features such as searching for reports, liking, commenting, viewing report progress, creating complaint, changing profile photos, and changing passwords to maintain account security. The Regular User Home Page can be seen as shown in Figure 6.

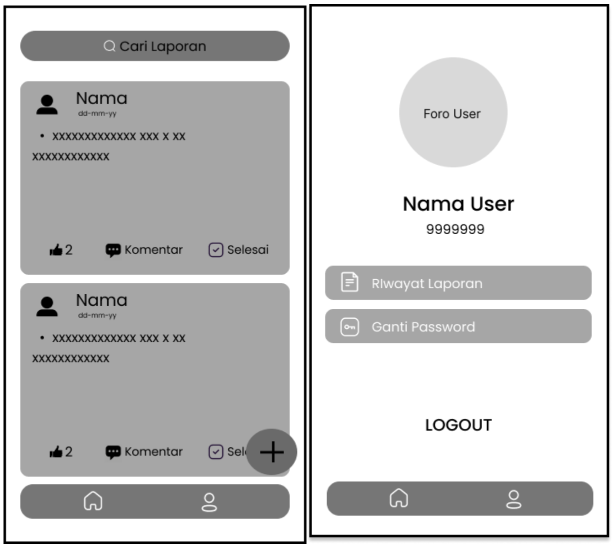


Figure 6: Regular User Home Page Application

D. Construction of Prototype

After the prototype is created and has been evaluated based on user needs, the developer can use the prototype to design a new system. The system developed as a medium for the stakeholder to make complaints, complaints or suggestions for any problems that occur in the campus environment. This application was developed based on android mobile using the Flutter framework. The following are the results of the user interface design:

1. Login Page

The initial display in the application, useful for verifying whether the user is registered or not to access the application. there are two access rights, namely login as user and login as admin. The login interface display is contained in Figure 7 below:

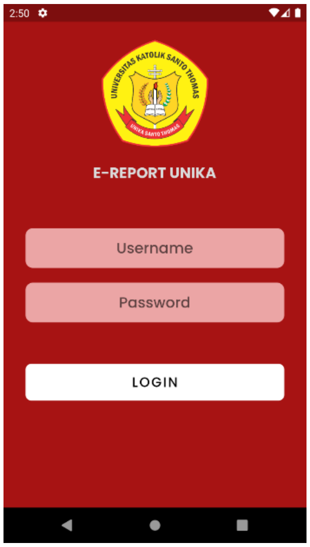
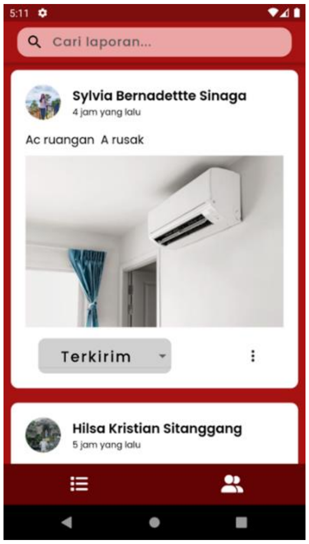


Figure 7: Login Page

2. Admin Home Page

This page will appear for the first time after logging in as an admin, there are various features such as viewing reports, searching for reports, changing report status and deleting reports. The login page image can be seen in Figure 8:



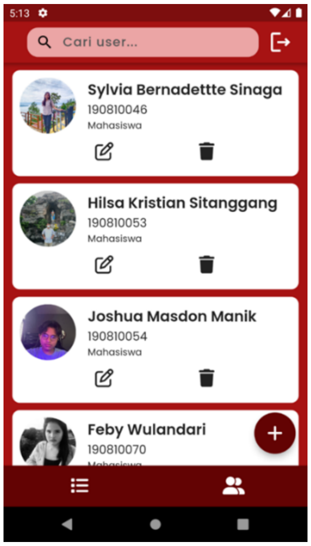
Sylvia Bernadettte Sinaga
4 jam yang lalu
Ac ruangan A rusak

Hilsa Kristian Sitanggang
5 jam yang lalu

Figure 8: Admin Home Page

3. User Data Manager Page

This page is also dedicated to admins who function to view users, search for users, add users, edit user data, delete users and log out. The visitor report interface can be seen in Figure 9.



Sylvia Bernadettte Singga
190810046
Mahasiswa

Hilsa Kristian Sitanggang
190810053
Mahasiswa

Joshua Masdon Manik
190810054
Mahasiswa

Feby Wulandari
190810070

Figure 9: User Data Manager Page

4. Add and Edit User Data Page

This page contains forms for adding user and editing user data by the admin. Figure 10 displays the add user page while Figure 11 displays the edit user data page.

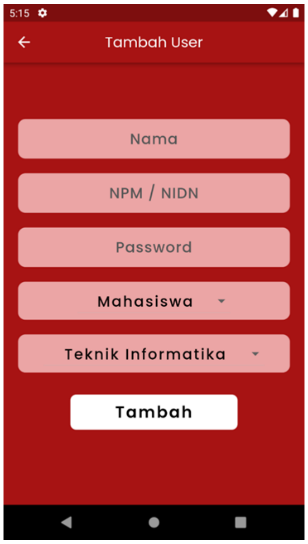


Figure 10: Add User Page

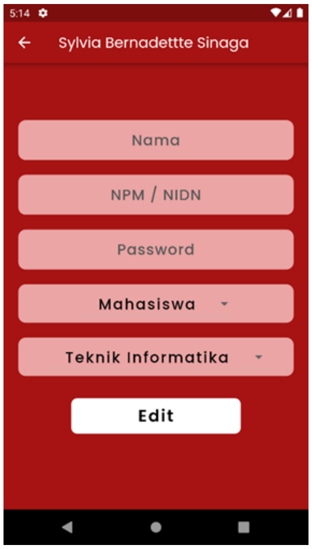
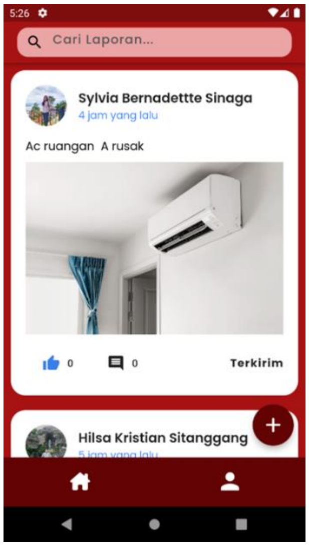


Figure 11: Edit Data User Page

5. User Home Page

This page is the initial page if logged in as a regular user. There are several features on this page, namely viewing all complaints or reports, searching for reports, searching for reports, liking, commenting and making complaints. Can be seen in Figure 12.



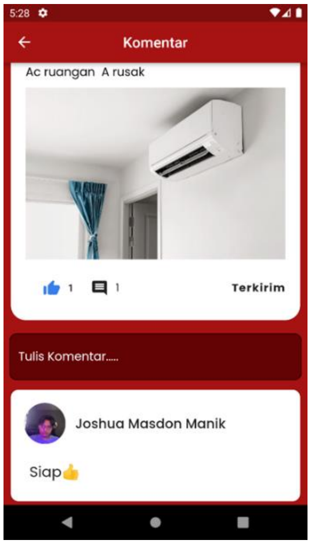
Sylvia Bernadettte Sinaga
4 jam yang lalu
Ac ruangan A rusak

Hilsa Kristian Sitanggang
5 jam yang lalu

Figure 12: Regular User Home Page

6. Comment Page

This page is dedicated to users who want to make comments on one of the complaints. The display of this page can be seen in Figure 13.



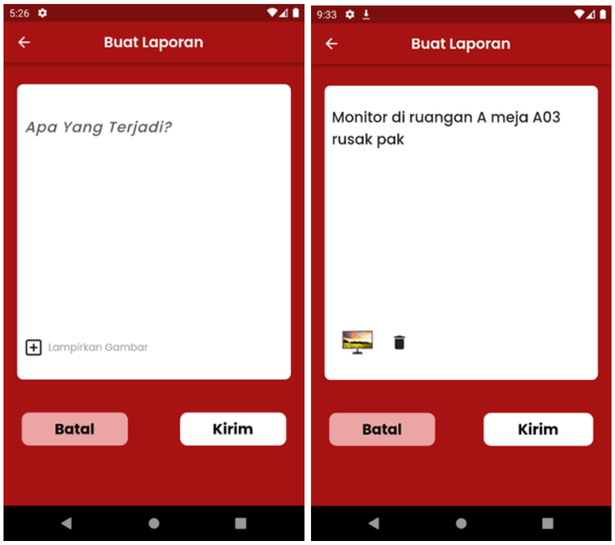
Ac ruangan A rusak

Joshua Masdon Manik
Siap👍

Figure 13: Comment Page

7. Page Creating a Complaint

The display of this page contains text and image input forms for users to create complaint reports. can be seen in Figure 14.

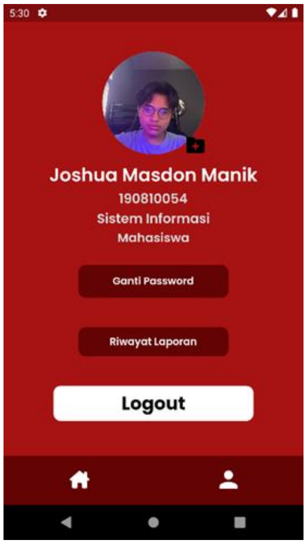


Monitor di ruangan A meja A03 rusak pak

Figure 13: Create Complaint Page

8. Profile Page

This page contains user profile data. On this page there are features such as changing profile photos, changing passwords, viewing history reports that have been created and logging out. Can be seen in Figure 15.



Joshua Masdon Manik
190810054
Sistem Informasi
Mahasiswa

Figure 15: Profile Page

9. Change Profile Photo Page

This page has a feature to change the profile photo from the user's smartphone gallery. Can be seen in Figure 16.

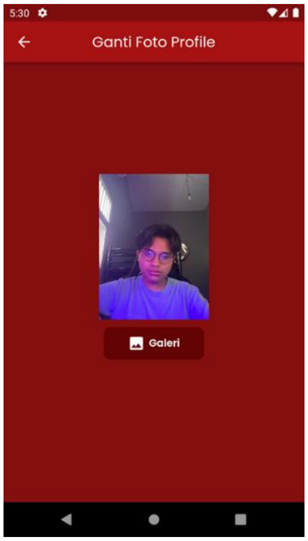


Figure 16: Change Profile Photo Page

10. Change Password Page

This page contains a form if the user wants to change their password. The old password required to change the password can be seen in Figure 17.

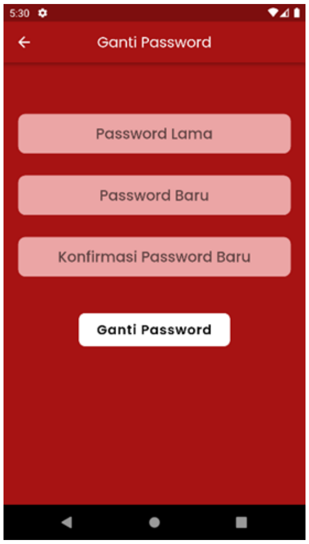
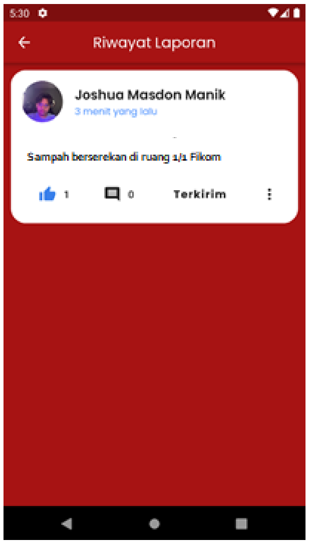


Figure 17: Change Password Page

11. Complaints History Page

This page contains a history of complaints that have been made by users and can also delete reports that have been made. Can be seen in Figure 18.


5:30

Riwayat Laporan

Joshua Masdon Manik
3 menit yang lalu

Sampah berserekan di ruang 1/1 Fikom

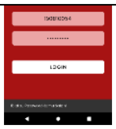
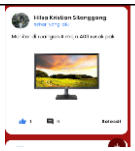
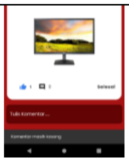
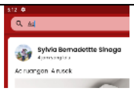
1 0 Terkirim

Figure 18: Complaints History Page

E. Development Delivery & Feedback

After the system building stage is carried out, the prototype is deployed and evaluated by stakeholders to provide feedback that is used for further system improvement. System evaluation is carried out with the support of a system trial process using the Blackbox testing method. Testing with Blackbox testing has been widely used by professional developers in the process of testing a software. This is due to the ease of the method offered which focuses on the results of programmer execution by creating steps and expected results. If the output results match the expected functional requirements, it means that a functional has been fulfilled and declared appropriate or valid. The opposite applies if the test steps have been carried out but the expected results do not match the expected output, it can be stated that there are still errors during coding. In essence, the purpose of Blackbox testing is to observe the possibility of missing functions or errors during coding. Table 1 shows some of the results of testing with Blackbox.

Table 1 : Blackbox Testing Results

Featured	Test Step	Expected Result	Display	Status
Login	Input an invalid username and password	Error massage		Success
Create a complaint	Send a blank complaint	Alert massage display		Success
Like and unlike	Pressing the like button	Like increases and decreases when pressed again		Success
Comment	Send an empty comment	Alert massage display		Success
Change password	Correctly input both the old password and the new password	Success massages		Success
Search complaints	Search same text complaints	Display various reports that match to input words		Success

4. CONCLUSION

The development of an E-report system for complaint management at Catholic University of St. Thomas using the prototyping method was a significant step forward for the university. The new system is a vast improvement over the previous complaint management system, as it is more user-friendly and efficient.

The success of the project can be attributed to the collaborative efforts of all stakeholders involved, including the developers, end-users, and university management. The iterative approach allowed for continuous feedback and refinement, resulting in a system that accurately met the needs of its users.

This project serves as an excellent example of the importance of involving end-users in the development process, as well as the value of using an iterative approach to software development. The success of the E-report system has the potential to inspire other organizations to improve their complaint management processes.

Overall, the development of the E-report system for complaint management at Catholic University of St. Thomas using the prototyping method is a significant achievement that will benefit the university and its stakeholders for years to come.

REFERENCES

[1] N. Kshetri, “Artificial intelligence in business and society: Opportunities and challenges,” Routledge, 2022.

[2] W. Lin, and X. Zhang, “Online complaints management in higher education institutions: An exploratory study of three Australian universities,” *Journal of Higher Education Policy and Management*, 39(1), 80-96, 2017.

[3] J. Bughin, T. Catlin, M. Hirt, and P. Willmott, “Why digital strategies fail,” McKinsey Quarterly, 1-12, 2018.

[4] R. S. Pressman, “Software engineering: a practitioner's approach,” Palgrave Macmillan, 2014.

[5] N. Niu, and X. Chen, “Identifying and validating software requirements: A systematic literature review,” *Information and Software Technology*, 131, 106418, 2021. doi: 10.1016/j.infsof.2021.106418.

[6] R. S. Pressman, and B. R. Maxim, “Software engineering: A practitioner's approach,” McGraw-Hill Education, 2015.

[7] C. Ghezzi, and D. Mandrioli, “Fundamentals of Software Engineering,” Springer, 2022.

[8] P. G. Larsen, and B. Fitzgerald, “Collaborative Software Engineering: A Comprehensive Overview,” Springer, 2021.