Journal of Digital Technology and Computer Science
E-ISSN: 3030-8127; P-ISSN: 3031-0318
https://journal.lontaradigitech.com/DIGITECH

Classify: Classroom Reservation Mobile Application for Students Based on Agile Development

Joy Febrianto Hale*[1], Muhammad Taufiq Alhidayah Syah [2], Surya Radi Ramdhani. S [3]
[1,2,3,4,5] *Universitas Negeri Makassar, Indonesia*

Corresponding e-mail: joyfebrian17@gmail.com

ARTICLE INFO

Keywords:

Agile Method
Android
Classroom reservation
Mobile Application

Article History

Received: August 15, 2025
Revised : September 25, 2025
Accepted : October 05, 2025

ABSTRACT

Classroom management in higher education often suffers from inefficiencies and scheduling conflicts as the reservation process is still carried out manually. To address these issues, this research developed Classify, an Android-based mobile application designed to help students reserve classrooms in a structured, efficient and digital way. The development followed the agile method using the Scrum framework, which was divided into three iterative sprints, including system planning, interface design with Figma, backend integration through Firebase and implementation of the main functions such as login, class list per floor, booking system and user profile management. System modeling is done with the Unified Modeling Language (UML), including Use Case Diagram and Activity Diagram. The tests are carried out using black-box and white-box methods to ensure the reliability of the system and the correctness of the program logic. The test results show that all functions work as expected and meet user requirements. Classify is thus able to improve the efficiency of the classroom reservation process, reduce the administrative burden and contribute to the digitalization of academic services in higher education.

To cite this article : Author. (20xx). Title. Information Technology Education Journal, X(X), XX-XX. Doi. xxxx

INTRODUCTION

Digital transformation in the world of higher education is experiencing rapid growth, especially after the COVID-19 pandemic forced educational institutions to drastically change the pattern of academic services. Universities in Indonesia and the world are required to provide services that are more flexible, efficient, and based on digital technology. Student learning styles have also changed, from conventional to more active, collaborative, and relying on access to services quickly through digital devices [1]. One of the academic services that are increasingly needed is booking classrooms or discussion rooms for academic purposes such as group studies, final project consultations, and practicum activities. However, the reality on the ground shows that this service is often not integrated in the existing academic information system. The reservation process is still done manually-through letters of request, verbal communication, or intermediaries of department coordinators-which results in overlapping schedules, non-optimal

use of space, and triggers complaints from students and education staff [2]. This shows that an efficient and easily accessible reservation system is still a real unmet need.

In answering this problem, the utilization of mobile applications is a strategic choice. Mobile platforms have advantages in terms of accessibility, flexibility, and proximity to students' daily lives. Applications built specifically for room reservations allow the process to take place independently, in real-time, and without administrative barriers. However, developing a system like this requires a software approach that is not only fast, but also adaptive to the dynamics of user needs. Agile methodology emerges as a very suitable approach in this context. With its iterative and user engagement-oriented nature, Agile allows development teams to respond quickly to changing needs and conduct continuous evaluation at every stage of development. Scrum as one of the Agile frameworks offers the division of development into short sprints that include structured planning, implementation, and evaluation processes [3]. Research by Pakpahan and Simatupang (2022) shows that the application of Agile in the development of campus reservation systems can increase user participation and accelerate the system validation process [4].

The Agile approach also encourages the adoption of user-centered design and minimum viable product (MVP) principles, which provide room for teams to build systems based on real user behavior and preferences. A study by Fatman and Erlianti (2023) confirms that the success of a space reservation system is strongly influenced by the convenience of the user interface, clarity of the booking flow, and integration with campus services such as academic calendars and single authentication (SSO) [5]. This means that it is not only the technical aspects that are key, but also the extent to which the system integrates with the campus ecosystem and the daily needs of students. However, most of the existing research is still focused on the technical side of information systems without addressing the synergy between the Agile approach and participatory design. Iterative approaches that involve students as end-users directly are still rarely found, even though such involvement is very important to build a truly relevant and useful system [6]. In addition, the utilization of mobile platforms in the context of classroom reservation in the campus environment is still very limited, even though the potential for use is very large.

Departing from this research gap, this study focuses on the development of the Classify application, a mobile-based classroom reservation system developed using the Agile method. The main goal is to create a system that is adaptive to user needs, can adjust to dynamic campus policies, and provide an intuitive usage experience for students. The development process was conducted through several iterative sprints by involving students directly in the evaluation and feedback stages [7]. Scientifically, this research offers two main contributions. First, the system was developed with full integration of the academic calendar and campus authentication, thus reducing schedule clashes between units and ensuring the validity of user data. Second, the Agile method applied serves not only as a technical approach, but also as a participatory design framework that puts students at the center of the development process. With this combination, Classify is expected to be an effective, adaptive, and digital solution capable of driving the transformation of academic services in higher education.

METHOD

Subjects and Objects of Research

The subjects in this research are students of the Department of Informatics and Computer Engineering (JTIK), Makassar State University, who are the main users of the Classify application. This application is designed to facilitate them in booking classrooms independently, anytime and anywhere. Meanwhile, the object of this research is the Classify application system itself, which is

an Android-based mobile platform developed using the Agile method. The system includes features such as login, class selection by floor, time booking system, and user profile management. The focus of the development is aimed at creating an efficient, secure, and appropriate classroom reservation service that meets the academic needs of students in the JTIK environment.

Development Method

The Agile method is an iterative and incremental software development approach, where the development process is carried out collaboratively between the development team and stakeholders. This method can be applied in the development of goods inventory information systems to increase flexibility, responsiveness, and product quality [9]. Agile technique is a project management technique that uses short development cycles or commonly called sprints. This technique focuses on continuous improvement in product or service development. Agile uses a phased development approach, where software is released in stages, so as to reduce the burden of the process, produce high-quality code, and involve customers directly in the development process [10]. The design of the ordering and production system in this study was carried out using observation, interview, and literature study methods as data collection techniques. These techniques are used to understand existing problems and obtain solutions to these problems [11].

According to Kevin Oloan Simatupang, the Agile method is a technique that uses a short development cycle (sprint) that focuses on continuous improvement in system development [10]. Meanwhile, Billyanto Hendrik explains that the Agile method is a methodology in software development that uses small, iterative processes [12]. The advantages of the Agile Development method are that it is able to adapt to changes, can improve software quality through repeated testing, and also improve team collaboration. This method also encourages transparency, effective communication, and rapid feedback from users.

Figure 1. Research Stages Using the Agile Method

The software development cycle using Agile methods is depicted below. Agile is an iterative and incremental approach that emphasizes team collaboration, rapid and continuous product delivery, and flexibility to change. The process starts with the Plan stage, where the team defines the goals, needs, and scope of the project, including compiling the backlog and user stories. Next comes the Design stage, where the team concentrates on the design of the system, user interface, and other technical structures. In the Develop stage, the team starts writing code to build features based on the design that has been compiled. After that, the Test stage is conducted to ensure that the feature runs as expected and has no bugs. If it meets the standards, the feature goes to the Deploy stage to be released to the production or staging environment. Review, or review, is the next stage. Here, user feedback is collected for improvements in the next iteration. Finally, the

final version of the product is rolled out to users at the launch stage. This entire process is done repeatedly, so that each cycle results in a product that is better in terms of quality and better suited to customer needs.

1. Planning
   In this stage, design is carried out by collecting data and making a system plan to be developed for users in the form of distributing questionnaires to get the desired needs, followed by using system tools and application tools to create designs [12].
2. Design
   At this stage, the system is designed in depth to ensure that the functional requirements that have been collected can be implemented technically. The design starts with creating a system architecture. The purpose of this design is to ensure the system runs well and make it easier for users [13].
3. Develop
   At this stage, the system begins to be developed based on the results of the design. The development process is carried out by a team of developers using appropriate programming languages and frameworks. System modules are built and tested locally before entering the formal testing stage.
4. Testing
   At this stage, testing whether the program is running properly can be tested manually using blackbox, if an error is found, it can easily find the cause on the production server. The stage of testing the usability of the application before the implementation stage [24].
5. Deployment
   In this phase, the system is deployed in a production environment. Servers are set up to support the application, and integration with geolocation is done to ensure the system runs in real-time. After implementation, the system begins to be used by users on a limited scale to observe its initial stability and performance [16]. After the system passes the testing stage, the deployment process is carried out to the production environment. The system begins to be used for real by users, and is monitored to ensure that no major problems occur in the initial use.
6. Review
   Upon completion of the first sprint, hold a retrospective meeting to evaluate the process and results. b. Get feedback from stakeholders to ensure their needs are met and to refine the plan for the next sprint [21]. Evaluation is done after the system is running to identify weaknesses, shortcomings, and feedback from users. This information will be the basis for future system improvements and enhancements.
7. Launch
   The last stage is the official launch of the system to the public or all users. The system is considered ready and stable, and publication is carried out to expand the reach of users. At this stage, regular system maintenance is carried out to ensure the security of the system [15].

RESULTS AND DISCUSSION

The result of this development is a mobile application called Classify, which is designed to help the classroom reservation process efficiently and can only be used by students as the main user. The development of the Classify application was carried out using the Agile method, which consists of four main stages: system planning, needs analysis, design, and implementation which

includes coding and testing processes. The following is a description of the results of each stage in the Agile development method that has been applied to the Classify application.

1. System Planning Results

The planning stage began with observations and in-depth interviews with relevant stakeholders (in this case, the class representatives and other potential users) to gather system requirements, which were then documented in the form of a system request. The results of this system request indicated the need for a platform capable of efficiently managing classroom reservation processes, particularly to support well-structured academic activities. Subsequently, the identified requirements were analyzed through a feasibility study to assess whether the proposed system could be realistically developed based on the available resources.

Table 1. Technical Feasibility

System Feasibility Study
Technical Feasibility • Study In the Classify application development project, technical aspects were assessed from the availability of development tools, technological infrastructure, and the suitability of the system to user needs. • Platform and Technology Used The Classify application was built using Android Studio as the main Integrated Development Environment (IDE), with the Java programming language which has matured and is widely used in the development of Android-based mobile applications. For data storage and synchronization, Firebase Realtime Database is used which supports real-time two-way communication, suitable for the needs of a class booking system that relies on fast and accurate data updates. • System Compatibility This application is designed to run on Android devices version 10 and above, which covers the majority of smartphones used by students today. The use of Firebase also extends compatibility as the cloud service is easy to integrate with mobile applications and does not rely on local backend systems. This makes the Classify app compatible with a wide range of devices and network environments. • Compatibility with External Systems Dependence on a third-party service (Firebase), if there is a service disruption on Firebase, the system will be disrupted.
Economic Feasibility Study The development of this application does not require additional license or hardware costs because it uses open-source tools such as Android Studio and Firebase (free package). Thus, from an economic perspective, this project is very feasible because it does not require large costs and has high potential benefits in overcoming the problem of classroom availability.
Operational Feasibility Study This system is designed to be used by students in the Department of Informatics and Computer Engineering (JTIK), who generally have a basic understanding of the use of mobile applications. Therefore, the operational use of the system does not pose any significant obstacles. In addition, Classify offers a simple interface and intuitive navigation to make it easier for users to make classroom reservations.

2. Needs Analysis

At this stage, an analysis of system requirements is carried out based on the problems faced by users, namely students in the Department of Informatics and Computer Engineering (JTIK),

Makassar State University. This analysis is divided into four parts, namely functional needs, non-functional needs, use case diagrams, and activity diagrams.

Table 2. Functional Requirements

Functional Requirement		
ID	**Fitur**	**Explanation**
FR001	Main page	Display a list of classes that can be booked by users
FR002	Login	Allows users to log in to the system using the username and password provided by the developer.
FR003	change password	Allows users to change passwords according to their needs
FR004	Class order	Feature for users to book the class they want to use by entering the date, time, and origin of the user's class.

Table 3. Non-Functional Requirements

Non-Functional Requirement		
ID	**Parameter**	**Explanation**
NFR-001	Availability	The application must be available for access 24 hours a day, 7 days a week, with a minimum uptime of 99.9%.
NFR-002	Reliability	The application must have a very low failure rate, no more than 0.1% of the operational time.
NFR-003	Ergonomy	The app should be designed with a user interface that is intuitive and easy to use for a wide range of users.
NFR-004	Portability	The app must be compatible and interoperable across multiple platforms including Windows, macOS, iOS, and Android.
NFR-005	Memory	The application should be efficient in memory usage, should not require more than 100 MB of memory on the user's device.
NFR-006	Response time	The application must be able to display the results or feedback from user actions in no more than 3 seconds.
NFR-007	Security	Applications should use end-to-end data encryption and two-factor authentication to secure user and transaction data.
NFR-008	Others 1: Communication	All Q&As in the app must be presented in Bahasa Indonesia.

The system analysis stage not only includes the identification of functional and non-functional requirements, but also emphasizes the importance of systematically documenting and visualizing the system structure using the Unified Modeling Language (UML). In this stage, two main types of UML diagrams are used: Use Case Diagrams, which show how actors (in this case students as the main users) interact with the Classify system, as well as Activity Diagrams, which illustrate the process flow or sequence of activities in the classroom reservation system. The use of these two diagrams helps ensure that the integration between functional and non-functional requirements is carried out effectively to support the achievement of system development goals. Further visualization of the diagrams is shown in Figure 2.

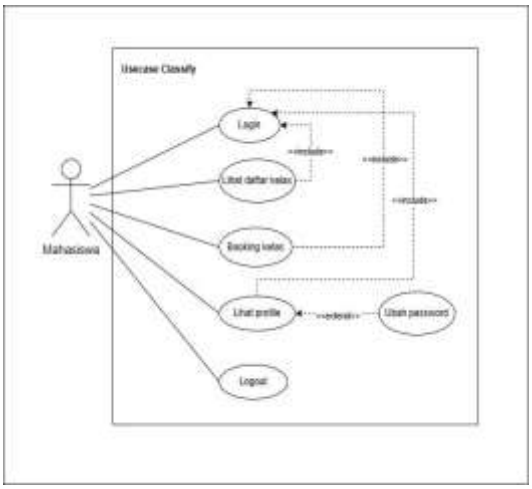

Figure 2. Anwendungsfall-Diagramm

a. Login: The user (student) authenticates himself in the system with NIM and password.
b. View class list: After successfully logging in, the user can view the list of available classes, divided by floor.
c. Book class: The user can select the room, date and time to book a class.
d. View profile: Users can view profile details such as name, NIM, class and email registered in the system.
e. Change Password: Users can change their password to protect their account.
f. Logout: The user logs out of the account to prevent others from accessing the account.

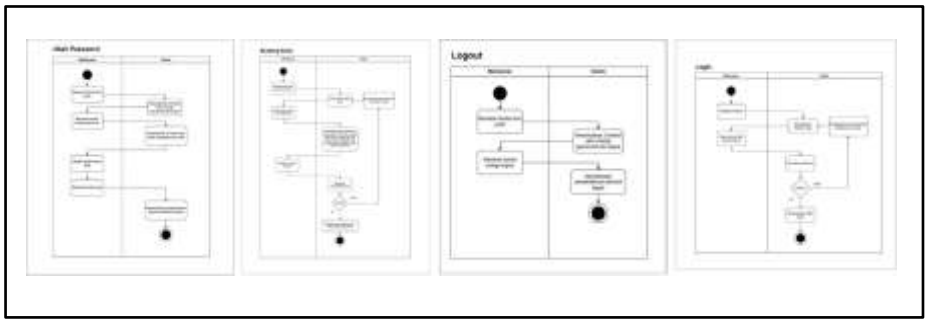

Figure 3. Activity diagram

Thus, this diagram illustrates the various interactions performed by one main actor, namely the Level Leader (User), in using the core features of the Classify application. These interactions include the login, logout process, make class room reservations (booking), and change passwords. This diagram provides a clear visual representation of the flow of application usage by users directly and independently without the involvement of the admin role.

3. Design

At the design stage in this research, several important components have been developed, namely the design of Class Diagrams, Sequence Diagrams, and User Interface (UI) which have been adjusted to the needs of the system. The adjustment process is carried out to ensure that each design element is able to optimally support the functionality of the Classify application.
The user interface is designed with the principle of ease and comfort for users, namely students.

Some UI elements, such as the space booking layout, the list of available classes, and the space detail page, refer to similar application design references but have been modified according to the needs of the Classify application.

1. Desain class and sequence diagram

   To design the visual design of the program, the Unified Modeling Language (UML) is used as an object-oriented modeling language. UML helps simplify complex problems to make them easier to understand [22]. At this stage, modeling is done in the form of Class Diagrams to describe the class structure along with attributes, methods, and relationships between classes as shown in Figure 4.

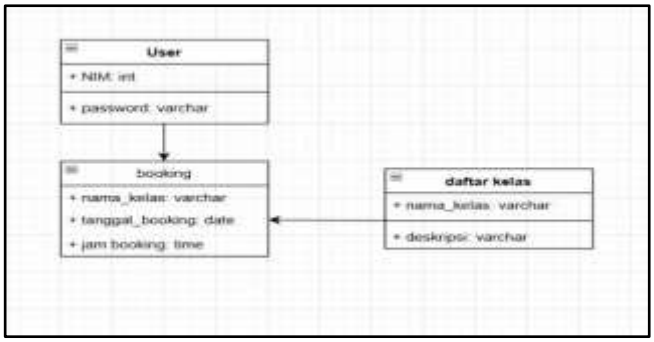

Figure 4. Class diagram

In addition, Sequence Diagrams are used to help understand system requirements, document processes, and visualize technical scenarios when the system is running [19], which can be seen in Figure 5.

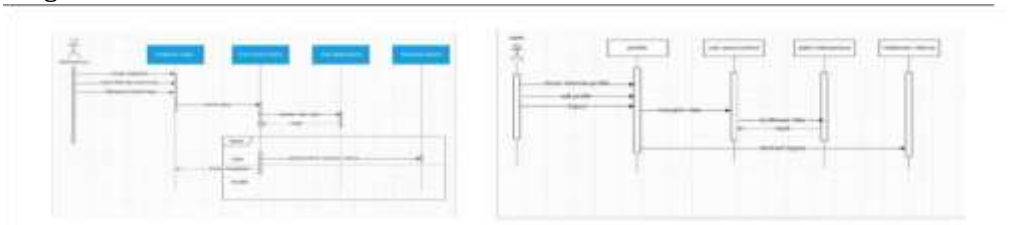

Figure 5. Sequence Diagram

2. User Interface (UI) Design

   The user interface (UI) of the Classify application was developed using Figma, guided by a comprehensive style guide that outlines design principles, visual rules, and interface elements to ensure consistency and user comfort throughout the application.

   The application development followed the Scrum methodology and was divided into three sprints, each representing an iterative cycle encompassing planning, implementation, testing, and evaluation of development outcomes. The first sprint focused on creating the Home and Login pages. The second sprint concentrated on developing fragments that display classroom lists by floor and the room booking system. Finally, the third sprint involved the development of the Profile, Change Password, and Logout features.

   a. First Sprint

      In the first sprint, the main focus of the Classify application development was to build

two essential initial components: the Home page and the Login page. The Home page was designed as the application's introductory interface, providing users with a first impression while serving as the starting point for navigating to other key features. The design emphasized accessibility and visual simplicity, enabling users to easily understand the core functions of the application.

Next, the Login page was developed as an authentication feature to restrict system access. Since the application is specifically intended for students, the login system was designed with special validation using a combination of student identification number (NIM) and password. This validation ensures that only authorized users can access the system and make classroom reservations.

During this sprint, the development team also ensured that the authentication process functioned properly, both from the user interface and programming logic perspectives. Testing was conducted to verify that the login page responded appropriately to both valid and invalid inputs. The outcomes of this first sprint provided a crucial foundation for the subsequent development of more advanced features in the following sprints.

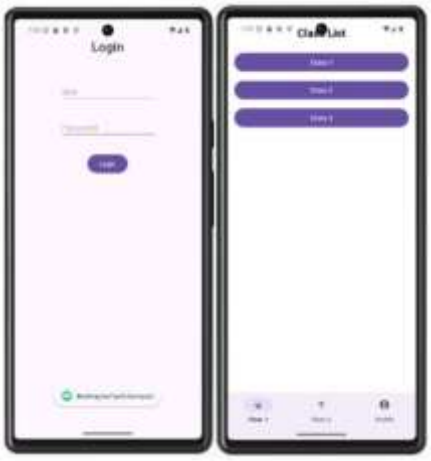

Figure 6. Design of the Homepage and Login Features

b. Second Sprint

In the second sprint, the development focus shifted to implementing two main fragments Floor 1 and Floor 2, each displaying a list of classrooms corresponding to their respective building levels. Each fragment presents the names of available classrooms, allowing users to select a room for the booking process.

The primary feature developed in this sprint was the classroom reservation system, which includes the selection of date and time. The booking process was designed to be time- and location-based to prevent scheduling conflicts among users. When a class representative makes a reservation for a specific classroom at a given time, the system stores the data in the Firebase Realtime Database, which immediately updates the room's availability status for other users.

Validation is performed before data submission to ensure that if a time slot is already taken, the system rejects the request and displays a notification to the user. This sprint also involved implementing real-time conflict-checking logic, which serves as the core functionality of the Classify system.

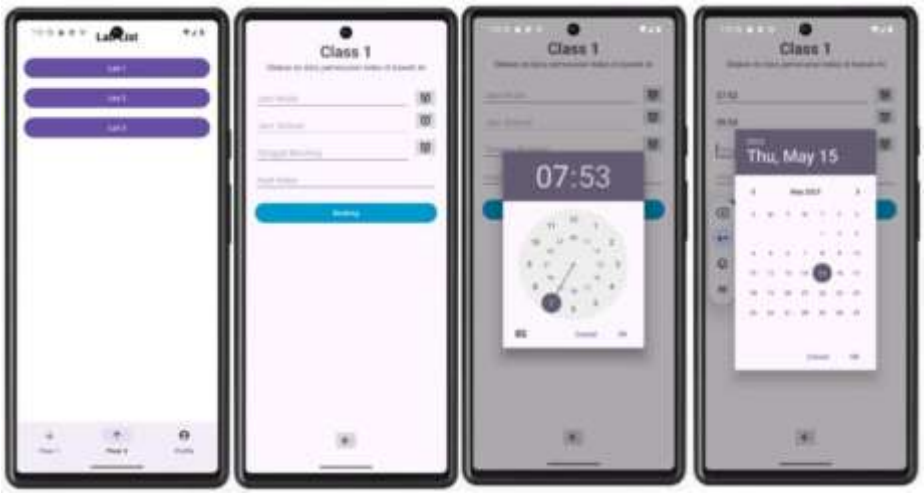

Figure 7. Booking System in the Application

c. Third Sprint

The final sprint focused on developing features related to personal data management and account security, namely the Profile, Change Password, and Logout pages. The Profile page displays the logged-in class representative's personal information, including name, student identification number (NIM), class, and email. This information is retrieved directly from the Firebase Database and presented in a concise and readable format.

In addition to viewing personal data, users are provided with the option to change their password through the Change Password page. In this feature, users must enter both their current and new passwords, after which the system verifies the credentials before updating the data in Firebase Authentication. This step is crucial for maintaining account security and ensuring that only authorized users can modify their credentials.

Finally, the Logout feature was developed to terminate the user session, redirecting the application back to the login page to prevent unauthorized access. These three features complete the functional scope of the application, ensuring that users can securely and independently manage their accounts.

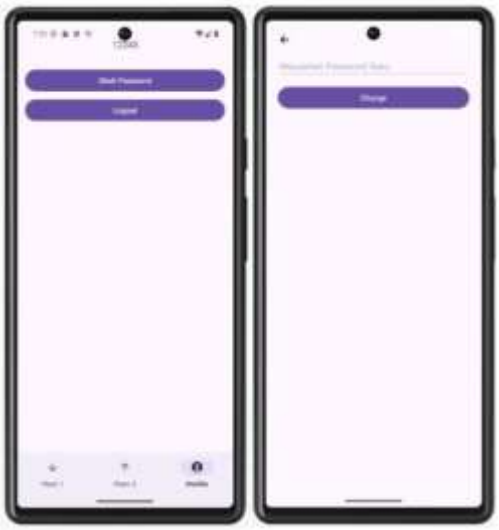

Figure 8. Change Password and Logout Features

4. Implementasi

a. Coding Results

The coding stage resulted in the full implementation of Classify, a mobile application designed for classroom reservations. The coding process was carried out using Android Studio as the main development environment, as the platform provides comprehensive support for the Java programming language used in the development of the Classify application. Android Studio also offers an efficient interface and a wide range of tools that facilitate the coding and debugging process [23].

Table 4. Coding Results

Coding	Output Display
First Sprint	
Login and Homepage 	
Second Sprint	
Fragments and Booking System 	
Third Sprint	
Profile, Change Password, and Logout	

b. Testing Results

System testing was conducted to evaluate the developed application and to provide users with an understanding of how the system operates [17]. The Classify application was tested using two approaches: Black Box Testing, which examines the application's functionality from the user's perspective without considering the underlying code structure, and White Box Testing, which inspects the internal logic flow of the program. Both methods were employed to ensure that the application performs according to requirements and is free from logical errors.

Black Box Testing

Black Box Testing was carried out by verifying the application's functionality based on input and output without examining the source code [25]. The objective of this testing was to ensure that the application responds correctly to various user scenarios. The results indicate that features such as login, room booking, profile management, and password change function as expected. Detailed results for each test case are presented in Table 5.

Tabel 5. Black Box Testing Results

Test Code	Test Case	Expected Result	Actual Result	Status
A-1	Login using unregistered NIM and password	Error message "Data belum terdaftar" is displayed	Error message "Data belum terdaftar" displayed	Successful
A-2	Login using correct NIM and password	Message "Login Berhasil" is displayed and redirected to Home page	Message "Login Berhasil" displayed and redirected to Home page	Successful
A-3	Click "Kelas" button	Displays input fields to be filled	Input fields displayed	Successful
A-4	Select booking time	Able to select booking time	Booking time selection successful	Successful
A-5	Select booking date	Able to select booking date	Booking date selection successful	Successful
A-6	Fill in class origin	Able to fill in class origin	Class origin filled successfully	Successful
A-7	Book the same class using different accounts	Message "Jadwal bentrok! Sudah ada booking pada waktu itu" is displayed	Message "Jadwal bentrok! Sudah ada booking pada waktu itu" displayed	Successful
A-8	Click "Floor 2" navigation	Displays list of classrooms on floor 2	List of classrooms on floor 2 displayed	Successful

A-9	Click "Profile" navigation	Displays Change Password and Logout features	Change Password and Logout features displayed	Successful
A-10	Click "Change Password" button	Redirected to Change Password page	Change Password page displayed	Successful
A-11	Click "Logout" button	Logs out and returns to Login page	Logged out and returned to Login page	Successful

White Box Testing

In this testing phase, the tests were designed from the developer's perspective since the internal structure of the system was known. The purpose was to examine all testable sections of the source code to identify any logical errors within the software [18]. The White Box Testing focused on analyzing the internal logic of the program, particularly the login feature. This testing involved creating a flowchart and a flowgraph, as well as calculating cyclomatic complexity to ensure that all execution paths were tested according to the predefined scenarios. The visualization of the testing process is presented in Figure 9.

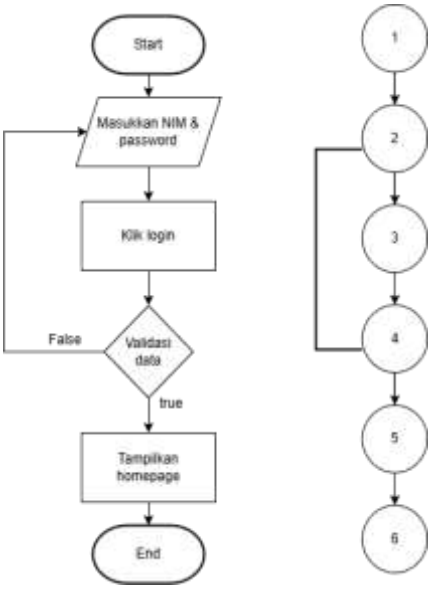

Figure 9. Flowchart and Flowgraph of the Login Process

The classroom booking feature serves as the core component that enables users to select rooms by floor, check time availability, and make direct reservations. The workflow was designed to be simple and efficient. White Box Testing was applied to trace each logical branch, including schedule conflict detection, input data validation, and the process of storing booking data into the system.

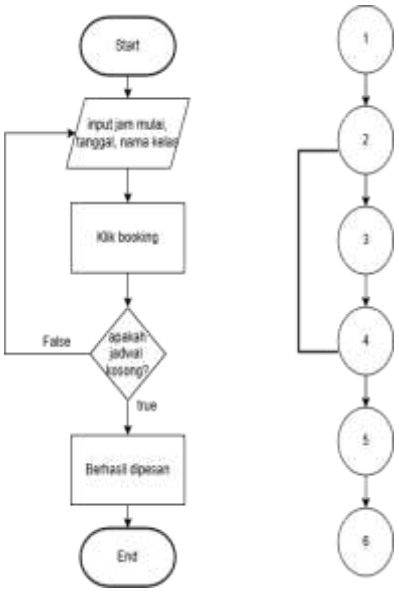

Figure 10. Flowchart and Flowgraph of the Classroom Booking Process

5. Discussion

The development results of the Classify application demonstrate that all main features login, classroom booking by floor, and account management, function properly and align with the designed testing scenarios. The time-based reservation feature with real-time validation successfully prevents scheduling conflicts and accelerates the classroom booking process for students.

Compared to the study by Pakpahan and Simatupang (2022), who developed a facility reservation system using the Agile approach but still based on a web platform, Classify offers greater mobility advantages as it is built on the Android platform [4]. Furthermore, Fatman and Erlianti (2023) emphasized the importance of a clear and user-friendly reservation flow, which has been implemented through Classify's consistent UI design [5].

The study by Rizky and Munawaroh (2024) also developed a mobile academic system using Jetpack Compose but did not include real-time schedule conflict validation as implemented in Classify [8]. Meanwhile, the system developed by Al Faruk et al. (2022), which was based on Laravel and web technology, did not integrate an academic calendar as the basis for time-based reservations [6]. Similarly, Manguling and Tambotoh (2023) applied the Agile method in developing a vehicle rental application but did not integrate authentication and campus service features key advantages of Classify [7].

In terms of testing, Fedianto et al. (2024) demonstrated the effectiveness of combining Black Box and White Box testing in verifying system logic, a method also applied in Classify's testing process [25]. Based on these comparisons, it can be concluded that Classify excels not only in technical performance but also in its participatory design approach, which involved users from the early development stages. This strengthens both the validity of the results and the system's relevance to real user needs.

CONCLUSIONS

This study successfully designed and implemented Comerch, a web based sales information system for cashier and product management, using the Waterfall approach through clearly defined stages of requirements analysis, system design, coding, integration, and testing, resulting in core capabilities for secure authentication and role based access, comprehensive product and stock management, reliable transaction processing with receipt printing, and downloadable reports that support managerial decisions; feasibility evaluations classify the project as technically and organizationally feasible, with the economic analysis yielding a positive cumulative net present value from the second year and a break even point of approximately two point zero five years; quality assurance is strong with forty six black box scenarios passing and high coverage white box tests confirming correct branching for user existence, password verification, and role redirection, indicating readiness for controlled deployment; the work contributes a reproducible blueprint that includes complete UML artifacts, a transparent feasibility framework, and a comprehensive test suite for practitioners; future improvements should address validation at larger scale and under higher loads, deeper security hardening following established secure coding practices, broader usability evaluation, tighter integration with production payment and logistics services, automated backup and monitoring, enhanced audit trails, and richer analytics to support data driven decisions.

REFERENCES

[1] R. Anggraeni dan I. E. Maulani, "Pengaruh Teknologi Informasi Terhadap Perkembangan Bisnis Modern," *Jurnal Sosial Teknologi*, vol. 3, no. 2, hlm. 94–98, Feb 2023, doi: 10.59188/jurnalsostech.v3i2.635.

[2] Y. S. Mulyani, "E-Commerce Solusi Pemasaran UMKM Dalam Mengembangkan Industri Pariwisata Di Tengah Pademi Covid-19 (Studi Kasus UMKM di Kota Tasikmalaya)," *Khasanah Ilmu - Jurnal Pariwisata Dan Budaya*, vol. 12, no. 2, Art. no. 2, Sep 2021, doi: 10.31294/khi.v12i2.11293.

[3] D. Riswan, H. E. R. Putra, dan R. N. Saputra, "Pengembangan Sistem Rekomendasi Berbasis Kecerdasan Buatan Untuk Meningkatkan Pengalaman Pengguna Di Platform E-Commerce," *Jurnal Komputer Teknologi Informasi dan Sistem Informasi (JUKTISI)*, vol. 2, no. 3, Art. no. 3, Feb 2024, doi: 10.62712/juktisi.v2i3.145.

[4] A. Adenowo dan B. Adenowo, "Software Engineering Methodologies: A Review of the Waterfall Model and Object- Oriented Approach," *International Journal of Scientific and Engineering Research*, vol. 4, hlm. 427–434, Sep 2020.

[5] S. Yulina, S. P. Arifin, dan Y. D. L. Widyasari, "PELATIHAN PENGEMBANGAN KOMPETENSI SISWA DALAM MERANCANG UI/UX MENGGUNAKAN FIGMA PADA SMK MUHAMMADIYAH 2 PEKANBARU," *Jurnal Pengabdian Masyarakat Multidisiplin*, vol. 8, no. 1, Art. no. 1, Okt 2024, doi: 10.36341/jpm.v8i1.4982.

[6] C. Ningki dan N. P, "Implementasi Aplikasi Penjualan Produk Tradisional Berbasis Website Menggunakan Metode Waterfall," *Informatik : Jurnal Ilmu Komputer*, vol. 19, no. 2, Art. no. 2, Sep 2023, doi: 10.52958/iftk.v19i2.6149.

[7] R. Risald, "IMPLEMENTASI SISTEM PENJUALAN ONLINE BERBASIS E-COMMERCE PADA USAHA UKM IKE SUTI MENGGUNAKAN METODE WATERFALL," *Journal of Information and Technology*, vol. 1, hlm. 37–42, Agu 2021, doi: 10.32938/jitu.v1i1.1393.

[8] M. Hendriawan, T. Budiman, V. Yasin, dan A. S. Rini, "PENGEMBANGAN APLIKASI E-COMMERCE DI PT. PUTRA SUMBER ABADI MENGGUNAKAN FLUTTER," *Journal of Information System, Informatics and Computing*, vol. 5, no. 1, Art. no. 1, Jun 2021, doi: 10.52362/jisicom.v5i1.371.

[9] S. Aisyah, R. Amelia, I. Sabrina, dan T. Syahrifah, "PERANCANGAN APLIKASI E-COMMERCE PADA TOKO LILI DENGAN PENERAPAN METODE WATERFALL," *Jurnal Sistem Informasi dan Ilmu Komputer*, vol. 5, no. 1, Art. no. 1, Agu 2021, doi: 10.34012/jurnalsisteminformasidanilmukomputer.v5i1.1917.

[10] T. Haryati, D. H. Kusuma, dan H. Ferliyanti, "Penerapan Metode Waterfall Sebagai Pengembangan Perangkat Lunak Sistem Informasi Penjualan PT. Arta Putra Nugraha Karawang," *Simpatik: Jurnal Sistem Informasi dan Informatika*, vol. 1, no. 2, Art. no. 2, Des 2021, doi: 10.31294/simpatik.v1i2.955.

[11] R. A. Ghani, N. W. Azani, S. N. Auliani, S. Maharani, M. D. Gustinov, dan M. L. Hamzah, "Perancangan Sistem Informasi e- Commerce Berbasis Website Menggunakan Metode Waterfall," *Prosiding Seminar Nasional Teknologi Informasi dan Bisnis*, hlm. 99–106, Jul 2022.

[12] A. Hidayat, F. Faisal, K. P. Harum, dan M. M. Fakhri, "Sistem Informasi Permintaan Barang Atk (Alat Tulis Kantor) Pada Pengadilan Negeri Makassar Kelas Ia Khusus," *Jurnal Media Elektrik*, vol. 20, no. 2, hlm. 77–85, 2023.

[13] B. Yuanita, M. M. Fakhri, dan W. Hidayat, "Sistem Informasi Medical Record Pasien Pada Puskesmas Sudu Kec. Alla Kab. Enrekang Berbasis Website," *Journal of Embedded Systems, Security and Intelligent Systems*, hlm. 93–100, 2021.

[14] Fariz Rahmat Mulyadi dan Yuda Syahidin, "Rancang Bangun Sistem Informasi Kepegawaian dengan Metode Waterfall," *Explore: Jurnal Sistem Informasi dan Telematika*, vol. 12, no. 2, hlm. 186–196, Des 2021, doi: 10.36448/jsit.v12i2.2056.

[15] "A Wearable Sonification System to Improve Movement Awareness: A Feasibility Study." Diakses: 24 April 2025. [Daring]. Tersedia pada: https://www.mdpi.com/2076-3417/14/2/816

[16] M. Prihandoyo, "Unified Modeling Language (UML) Model Untuk Pengembangan Sistem Informasi Akademik Berbasis Web," *Jurnal Informatika: Jurnal Pengembangan IT*, vol. 3, hlm. 126–129, Jan 2018, doi: 10.30591/jpit.v3i1.765.

[17] S. Syauqi dan S. Suendri, "Information System Design of Web-Based Document Archives Management In The Office Bappeda of North Sumatra Province," *Journal of Information Systems and Technology Research*, vol. 1, no. 1, Art. no. 1, Jan 2022, doi: 10.55537/jistr.v1i1.66.

[18] M. M. Fakhri, M. S. J. Irmawan, A. S. Alwi, I. F. Asril, N. Q. Ridhaihi, dan D. Fadhilatunisa, "Perancangan Sistem Informasi Manajemen Karyawan Berbasis Website dengan Metode Waterfall," *Jurnal Mediatik*, hlm. 35–44, 2023.

[19] Z. A. Hamza dan M. Hammad, "Analyzing UML use cases to generate test sequences," Jan 2021, Diakses: 24 April 2025. [Daring]. Tersedia pada: https://journal.uob.edu.bh:443/handle/123456789/3923

[20] R. Fauzan, D. Siahaan, S. Rochimah, dan E. Triandini, "Automated Class Diagram Assessment using Semantic and Structural Similarities," *International Journal of Intelligent Engineering and Systems*, vol. 14, no. 2, hlm. 52–66, 2021, doi: 10.22266/ijies2021.0430.06.

[21] M. R. Wayahdi dan F. Ruziq, "Pemodelan Sistem Penerimaan Anggota Baru dengan Unified Modeling Language (UML) (Studi Kasus: Programmer Association of Battuta)," *Jurnal Minfo Polgan*, vol. 12, no. 1, Art. no. 1, Agu 2023, doi: 10.33395/jmp.v12i1.12870.

[22] Y. S. Purbo, F. S. Utomo, dan Y. Purwati, "Analisis dan Perancangan Antarmuka Aplikasi Wisata Menggunakan Metode User Centered Design (UCD)," *Jurnal Teknologi Terpadu*, vol. 9, no. 2, Art. no. 2, Des 2023, doi: 10.54914/jtt.v9i2.977.

[23] M. Muauwanah, H. D. Saputra, M. Afifudin, dan M. Y. P. Chusnani, "Implementasi Pengujian Black Box pada Form Login E-Learning di Universitas Airlangga," *Jurnal Sistem Informasi, Manajemen dan Teknologi Informasi*, vol. 2, no. 2, Art. no. 2, Jul 2024, doi: 10.33020/jsimtek.v2i2.727.

[24] "Keamanan Sebuah Sistem Point of Sale: Sampai Sejauh Mana?" Diakses: 24 April 2025. [Daring]. Tersedia pada: https://www.opaper.app/blog/keamanan-point-of-sale

[25] C. A. Pamungkas dan A. Lutfiyani, "Sistem Informasi Kasir pada Usaha Serba Rasa Food Program Wirausaha Merdeka IPB University Berbasis Website Menggunakan Metode

Waterfall," *Jurnal Penelitian Inovatif*, vol. 4, no. 2, Art. no. 2, Apr 2024, doi: 10.54082/jupin.314.

[26] Lesmono, I. D. (2018). RANCANG BANGUN SISTEM INFORMASI PENJUALAN SEPATU BERBASIS WEBSITE DENGAN METODE WATERFALL. *Swabumi*, *6*(1). https://doi.org/10.31294/swabumi.v6i1.3316