Revamp of Grade and Attendance Management Features in the Academic Information System

Y K W Utama [1], G L Pritalia*[2], H Gunawan[3], R Perwiratama[4]

[1-4]Program Studi Sistem Informasi, Universitas Atma Jaya Yogyakarta, DI Yogyakarta

E-mail: yohkevinwu@gmail.com[1], generosa.pritalia@uajy.ac.id*[2], hendro.gunawan@uajy.ac.id[3], rangga.perwiratama@uajy.ac.id[4]

Abstrak. Penelitian ini menyajikan perancangan ulang fitur pengelolaan nilai dan kehadiran dalam Sistem Informasi Akademik (SIAKAD) Universitas Atma Jaya Yogyakarta (UAJY) menggunakan pendekatan Desain Berbasis Domain yang didukung metode prototyping. Tujuan utama penelitian adalah memigrasikan fungsi kehadiran dan penilaian dari Sistem Penilaian Kinerja Pegawai (SPKP) ke SIAKAD untuk meningkatkan kualitas layanan akademik dan efisiensi operasional. Pengumpulan data dilakukan melalui observasi dan wawancara dengan *domain expert*, dilanjutkan dengan pemodelan domain untuk mengidentifikasi kebutuhan sistem yang spesifik. Arsitektur sistem disusun dalam tiga *bounded context* utama: Pengelolaan Nilai, Pertemuan Perkuliahan, dan Pertemuan Pengganti, yang masing-masing dirancang mengikuti prinsip Desain berbasis domain agar fungsionalitas selaras dengan kebutuhan pengguna. Tahap perancangan mengintegrasikan wireframe dan mockup prototype untuk memastikan visualisasi dan interaksi sistem memenuhi persyaratan akademik. Implementasi diikuti dengan Black-Box Testing komprehensif, termasuk validasi pihak ketiga, untuk memverifikasi bahwa fitur baru berfungsi sesuai spesifikasi. Hasil penelitian menunjukkan bahwa sistem yang telah dirancang ulang berhasil memenuhi kebutuhan operasional dan siap diimplementasikan, memberikan peningkatan integrasi dan efisiensi dalam proses administrasi akademik.

Kata kunci: Sistem Informasi Akademik; Manajemen Nilai; Manajemen Kehadiran; Perombakan; Desain Berbasis Domain

Abstract. *This study presents the redesign of grade and attendance management features in the Academic Information System (SIAKAD) at Universitas Atma Jaya Yogyakarta (UAJY) using Domain-Driven Design (DDD) supported by the prototyping method. The primary objective is to migrate attendance and grading functionalities from the Employee Performance Assessment System (SPKP) to SIAKAD to enhance academic service quality and operational efficiency. Data collection was conducted through observations and interviews with domain experts, followed by domain modeling to identify specific system requirements. The system architecture was structured into three main bounded contexts: Grade Management, Course Meetings, and Substitute Meetings, each designed following DDD principles to align functionalities with user needs. The design phase incorporated wireframe and prototype mockups to ensure proper system visualization and interaction that meet academic requirements. Implementation was followed by comprehensive Black-Box Testing, including third-party validation, to verify that the new features function according to specifications. The results demonstrate that the revamped system successfully fulfills*

operational requirements and is ready for deployment, providing improved integration and efficiency in academic administration processes.

Keywords: Academic Information System; Grade Management; Attendance Management; Revamp; Domain-Driven Design.

1. Introduction

Education in Indonesia continues to undergo significant transformation, particularly in adapting to workforce demands and technological advancements. Higher education institutions must update their systems to ensure operational efficiency and relevance to user needs. One common strategy implemented is the revamp or renewal of academic systems. Santoso (2020) stated that updating academic systems can enhance the efficiency and relevance of higher education institutions in addressing the ever-evolving challenges of the industry [1].

Universitas Atma Jaya Yogyakarta (UAJY) is one of the higher education institutions actively leveraging technology to support academic activity management. At UAJY, a centralized academic information system known as the Academic Information System (SIAKAD) has been used to manage various functions, including course registration (KRS), class management, and support for the implementation of the Merdeka Belajar – Kampus Merdeka program. However, SIAKAD does not yet include grade and attendance management, which is still handled through a separate system called the Employee Performance Assessment System (SPKP). Initially, SPKP was designed for employee performance management, but in practice, it is used for student grade and attendance management, which clearly does not align with its primary purpose.

The use of SPKP for academic functions creates inconsistencies because, although it shares the same database with SIAKAD, SPKP is more suited for employee performance management than academic purposes. This SPKP feature confuses users who are more familiar with the academic system in SIAKAD. Furthermore, the technology used in SPKP is outdated, making it difficult to maintain or update. This results in limitations in system development and complicates maintenance and future integration with modern systems.

The migration of grade and attendance management from SPKP to SIAKAD is essential. SIAKAD is the primary system frequently used by administrative staff and supports various academic activities. Integrating this feature into SIAKAD will make academic management more centralized and organized. Additionally, the grade and attendance management feature in SPKP is the last remaining component, as other features have already been migrated to newer, more modern, and relevant systems.

This update is also necessary to improve the quality of academic services. Research by Pratama indicates that periodically updated systems can enhance user satisfaction and operational efficiency in higher education institutions [2]. Thus, developing SIAKAD to include more relevant features facilitates operational smoothness and meets users' expectations of a modern information system.

Moreover, this system revamp plays a crucial role in increasing operational efficiency. A study conducted by Lestari found that strategically updated information systems can reduce administrative processing time by up to 30% [3]. This highlights the importance of a system continuously adapting to evolving academic needs. As student enrolment grows, the complexity of academic management increases. A study by Putri shows that inefficient systems can lead to errors and slow down administrative processes [4]. Therefore, UAJY's Information Systems Office (KSI) must ensure that SIAKAD can address these challenges by providing more suitable and efficient features. In the digital era, updating academic information systems is an essential requirement. Research by Subagyo states that educational institutions that actively update their information systems are better positioned to compete and provide superior services to students and academic staff [5]. Therefore, UAJY must prioritize the development of grade and attendance management features to support the sustainability of academic services.

2. Method

This research uses the Software Development Life Cycle (SDLC) framework. SDLC is a systematic approach that guides developers in handling the complexities of software projects, ensuring that all aspects are well-documented and reducing the risk of errors throughout the development cycle [6]. Additionally, SDLC is highly relevant in ensuring software alignment with user needs due to its systematic and iterative stages, which can enhance efficiency and system development quality [7] [8].

This research uses the Domain-Driven Design (DDD) methodology as the primary approach within the SDLC analysis framework (Tabel 1). DDD is a software development approach emphasizing the importance of understanding and modelling the domain in designing complex systems. This method is chosen because it fosters collaboration between software developers and domain experts (individuals who understand the details of a business domain or existing system), ensuring that the resulting model accurately reflects business needs and provides flexibility in adapting to business changes [9] [10] [11].

The prototyping method is also applied as the primary approach in the SDLC design phase. This method is selected because it allows developers and users to collaborate in understanding system requirements through an initial prototype that can be used to validate key workflows before full implementation [12].

Tabel 1. Mapping DDD activities to SDLC phase

SDLC phase	DDD activities	Output	Tujuan
Analysis	Data Collection Domain Modeling Defining Ubiquitous Language	Interview results Sitemap Use case diagrams	Understanding business domain and building shared vocabulary
Design	Defining Bounded Context Defining Context Map Subdomain Modeling Wireframe Creation	Bounded context diagrams Context map Detailed sitemap Wireframes	Designing modular architecture with clear context boundaries
Design	Mockup Prototype Development	Prototypes	Validating interface design with stakeholders
Implementation	Code Implementation	Functional system	Implementing bounded contexts into code
Testing	Black-Box Testing	Test result tables	Verifying functionality according to specifications

2.1. Research Stages

The following are the stages of this research:

2.1.1. Data Collection

In this stage, data collection is conducted by analyzing the grading and attendance management features in SPKP, which will be revamped into SIAKAD. This analysis aims to identify the key components that require updates. This process involves understanding the system's workflow and identifying relevant core business functions. Direct observation of the existing system and interviews with domain experts are the primary methods to obtain more accurate and in-depth information. The output of this stage is interviewing results, which provide initial data on key features and issues that need to be addressed to support the revamp of the grading and attendance management features in SIAKAD.

2.1.2. Domain Modeling

In this stage, high-level domain modeling is conducted to revamp the grading and attendance management features. Each domain includes core business functions previously identified during the data collection phase. The modeling process is detailed to align with SIAKAD requirements, focusing on relevant elements based on interviews and system analysis. This process utilizes visual representations in the form of site maps to manage system complexity more effectively and systematically.

2.1.3. Defining Ubiquitous Language

In this stage, a common language is defined for each domain previously modeled during the domain modeling phase. This process involves conducting interviews with domain experts to collect key terms and specific terminology for each domain, particularly those related to the grading and attendance management features to be modified. These interviews aim to ensure that all stakeholders consistently understand the meanings and usage of these terms across the domain. The output of this stage includes use case diagrams and use case specification documents, which reflect a shared understanding of terminology and workflow within the system.

2.1.4. Defining Bounded Context

In this stage, each modeled domain is then separated into bounded contexts to ensure modularity and independence between system components. Using Context Mapper, the boundaries between contexts are defined to minimize direct dependencies between them. The output of this stage is a bounded context diagram that outlines the identified entities and services.

2.1.5. Defining Context Map

This stage is used to map the relationships between bounded contexts. The mapping uses tools such as Context Mapper to illustrate data flow and dependencies between system components. The output of this stage is a context map diagram that provides a visual guide for interactions between different parts of the system.

2.1.6. Subdomain Modeling

This stage involves dividing the domain into smaller subdomains to facilitate the update process. Using a site map, each subdomain is modeled and grouped based on functional relevance to ease the transition to SIAKAD. The output of this stage is a more detailed site map.

2.1.7. Wireframe Creation

In this stage, wireframes are used to design the flow and structure of the proposed grading and attendance management features, focusing on element placement and relationships without emphasizing visual details. Wireframe creation follows a prototyping approach based on the earlier DDD analysis, ensuring the initial design aligns with system requirements. The output of this stage is a wireframe design that provides a clear overview of the system structure and interface flow for further development.

2.1.8. Mockup Prototype Development

After completing the wireframe, the next step is creating mockups and prototypes. The mockup refines the wireframe by incorporating visual elements such as colors, fonts, and icons, making it more realistic. The prototype is developed to test the design further and ensure that user interactions with the system interface function smoothly. The output of this stage is a prototype design that is ready for further testing.

2.1.9. Implementation

In this stage, the implementation process begins with refining the grading and attendance management features in SIAKAD. This process is carried out in stages based on the prototype designed in the previous stage, referring to requirements analysis and modeling. Each subdomain defined in the design phase is updated and adjusted until all features in SIAKAD are fully optimized. The output of this stage is an updated SIAKAD system with optimized grading and attendance management features.

2.1.10. Testing

The testing stage uses the Black-Box Testing method to ensure that the revamped grading and attendance management features function according to specifications. This test is performed by validating inputs, outputs, and system workflows without examining the code implementation in detail. The results of this testing are documented in a test result table, which includes pass/fail outcomes for each test scenario designed based on user requirements and system specifications.

3. Results and Discussion

3.1. Data Collection

Data was collected through observations of the old Grade and Attendance Management feature's code in SPKP and interviews with domain experts to understand the business processes and key components that need to be updated in the revamped feature within SIAKAD. The primary goal of this process was to analyze

the functionality of the existing feature and identify aspects requiring updates or adjustments for implementation in SIAKAD.

Since direct access to the SPKP application was impossible, the observation focused on analyzing the SPKP code flow. To complement the information that could not be obtained from code analysis alone, interviews were conducted with the Deputy Head of System Development and the system development staff at the UAJY Information Systems Office (KSI). These interviews provided more profound insights into system requirements and offered critical data regarding the core business functions and components that need to be updated. Key Findings from the Interviews:

- **Core Function:** The revamped feature will include Grade & Attendance Management, which consists of three main menus: Class Management, Faculty Class Management, and Replacement Meeting Management
- **System Improvements and Adjustments:** Simplification of the grade upload and storage process and adjustments to grade templates according to study programs and courses
- **Primary Users:** The feature will be accessible to lecturers, program heads, department heads, and administrative staff, with role-based access restrictions.
- **System Standards:** The system must comply with technology standards such as .NET 5 MVC, Rotativa for PDF generation, and OfficeOpenXml for handling Excel or CSV files.
- **SPKP Integration with Other Systems:** The old Grade and Attendance Management feature in SPKP was not directly integrated with other systems. However, the database used in SPKP is shared with other systems such as SIAKAD, SIATMA, SIATMA Mobile, and others.

3.2. Domains Modelling

At this stage, a high-level domain modeling process is conducted for the Grade & Attendance Management feature that will be revamped. The purpose of domain modeling is to provide a detailed representation of the structure and workflow of the feature, ensuring it aligns with SIAKAD's requirements.

This process utilizes visual representations in sitemaps to handle system complexity in a more structured and manageable way. Doing so provides a clear overview of the division and relationships between different system parts.

The outcome of this stage is creating a sitemap that illustrates the main structure of the revamped Grade & Attendance Management feature in SIAKAD. Figure 1 presents an overview of the core features and their functionalities within the system.

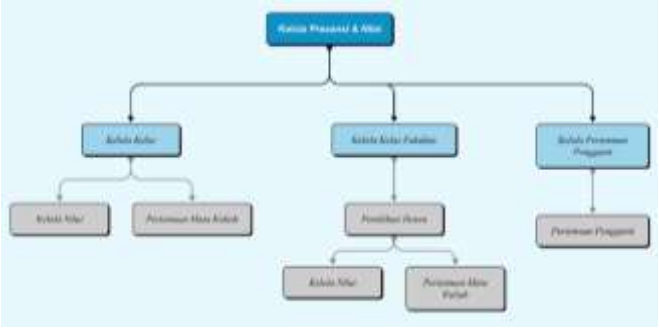

Figure 1. Sitemap of Grade & Attendance Management Features

In this structure, the "Grade & Attendance Management" feature is divided into several menus, each focusing on different user needs and responsibilities, including:

- **Manage Classes:** Used by lecturers to manage student grades and attendance for each class they teach. This menu includes both the grade management function and the course meeting management function.
- **Manage Faculty Classes:** Accessed by Heads of Study Programs (Kaprodi), Department Heads (Kadep), and Administrative Staff (TU) to oversee classes at the faculty level, including lecturer

administration and final grades. This menu also includes grade management and course meeting management functions. However, Kaprodi, Kadep, and TU have specific restrictions for course meeting management, particularly regarding certain sub-functions within this feature.

- **Manage Substitute Meetings:** Used by Administrative Staff (TU) to handle and approve class meeting reschedules requested by lecturers due to specific reasons, such as leave or scheduling conflicts. This menu has only one main function: substitute meeting management.

3.3. Defining Ubiquitous Language

This process was done through interviews with domain experts, specifically the Deputy Head of System Development and the system development staff from KSI UAJY. These interviews aimed to gather key terms and specialized terminology in the Grade & Attendance Management feature, which will be revamped for integration into SIAKAD.

The primary goal of defining this ubiquitous language is to align the understanding among all stakeholders involved in the development process, ensuring effective communication and minimizing potential misunderstandings. This consistency allows all parties to use uniform terminology when developing and comprehending the system. Additionally, it facilitates code implementation, as the developed code can be aligned with the agreed-upon terminology. Below are some of the terms included in the ubiquitous language, tailored to match the features and functions within this domain:

1. **Involved Actors:** Actors with access include Lecturers, Heads of Study Programs (Kaprodi), Department Heads (Kadep), and Administrative Staff (TU), each with different roles according to their responsibilities. Lecturers and TU are responsible for inputting and uploading grades, while Kaprodi and Kadep have supervisory roles to monitor grade data management.
2. **Grade Types and Criteria:** Lecturers act as primary managers, while Kaprodi, Kadep, and TU serve as observers. Their access is limited to monitoring student attendance, recording lecture materials, and viewing meeting details without making modifications.
3. **Grade Management Functions:**
    - **Download:** A function that provides a pre-formatted grade template in Excel, aligned with each course's Course Learning Outcomes (CPMK). Lecturers download this template and fill it according to the grading regulations.
    - **Upload:** Lecturers upload the file into the system once student grades are entered into the template. When the "Submit Grades" button is pressed, the system permanently saves the data, preventing further modifications.
    - **Print:** This feature allows lecturers to print or download the grade list in a well-structured PDF format, which is helpful for administrative purposes such as reports or archiving.
4. **Course Meeting Management Functions:**
    - **Lecturer Attendance**: Lecturer attendance is recorded in two stages:
        - **"Presensi In"** at the beginning of the session to mark attendance.
        - **"Presensi Out"** at the end of the session to finalize attendance recording.
        - Attendance must be completed within 3 days after the meeting.
    - **Student Attendance:** Students can mark their attendance via Quick Response Code (QR Code), or lecturers can record attendance manually or upload an attendance list from platforms such as Microsoft Teams for online meetings.
    - **Lecture Material Recording:** Lecturers can document the topics covered in each class meeting.
5. **Substitute Meeting Management Functions**: Once the substitute meeting request is approved by TU, the details, such as date, room, and session, are arranged to ensure the lecture proceeds smoothly without schedule conflicts.

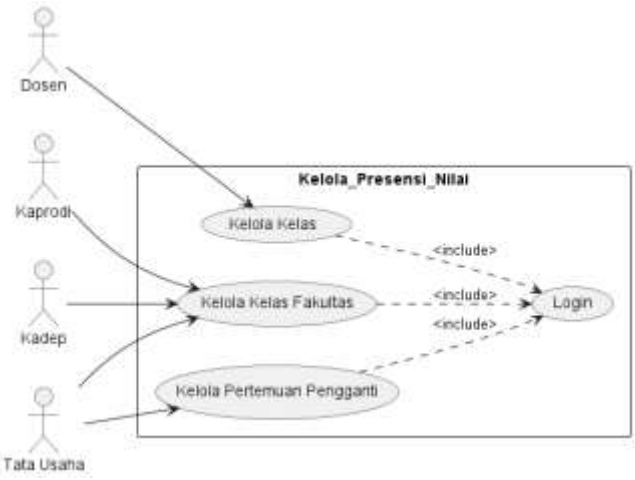

Figure 2. Use Case Diagram for Grade and Attendance Management

3.4. Defining Bounded Context

1. Grade Management Context

The Grade Management Context is a bounded context designed to handle the academic grading process within the educational information system domain. This bounded context implements the Grade Management domain, aiming to simplify the management of student grades in various aspects, such as Mid-Semester Exams (UTS), Final Exams (UAS), Final Grades, and remedial assessments.

Figure 3. Bounded Context of Grade Management Context

2. Course Meeting Context

The Course Meeting Context is a bounded context designed to manage various lecture meeting processes within the academic system. This context aims to record all activities related to lecture meetings, such as documenting discussed materials, submitting requests for substitute meetings, and recording the attendance of lecturers and students. The diagram consists of two main aggregates, Lecture Attendance, and Student Attendance, and several services supporting business processes.

Figure 4. Bounded Context of Course Meeting Context

3. Substitute Meeting Management Context

The Substitute Meeting Management Context is a bounded context responsible for handling the process of substitute meetings within the academic environment. This includes approval, rejection, and recording of schedule changes for lectures. This bounded context consists of a single main aggregate, a Substitute Meeting, and several services supporting the related business processes.

Figure 5. Bounded Context of Substitute Meeting Management Context

3.5. Defining Context Map

Figure 6 illustrates the relationships between the existing bounded contexts: *Grade Management Context*, *Course Meeting Context*, and *Substitute Meeting Management Context*. This Context Map represents the interconnections between these contexts based on two primary patterns: *Shared Kernel* and *Partnership*

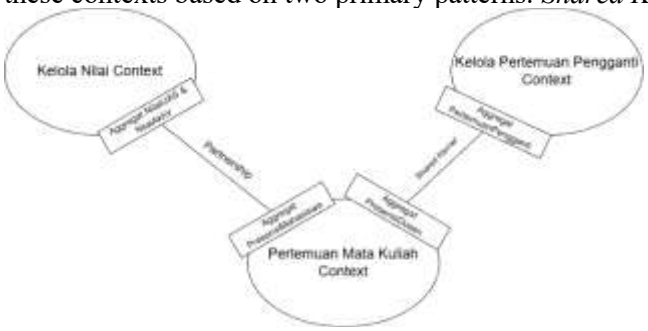

Figure 6. Context Map

The *Shared Kernel* relationship occurs between the *Substitute Meeting Management Context* and the *Course Meeting Context*. In this relationship, both contexts share a subset of the same domain model, specifically the *PresensiDosen* entity or aggregate. This allows both contexts to maintain a consistent domain model, ensuring that data related to lecturer meetings, including substitute meetings, are managed with proper synchronization.

The relationship between *the Grade Management Context* and *the Course Meeting Context* exists. In this relationship, the *Course Meeting Context* contains the *PresensiMahasiswa* aggregate, a critical data source for the *Grade Management Context* to calculate student attendance percentages needed in the *NilaiUAS* and *NilaiAkhir* aggregates.

3.6. Subdomain Modelling

1. Subdomain in the *Grade Management* Domain

This subdomain encompasses all functions and processes related to managing grades, such as *Download*, *Upload*, and *Print*. The structure of this subdomain is illustrated in *Figure 7*.

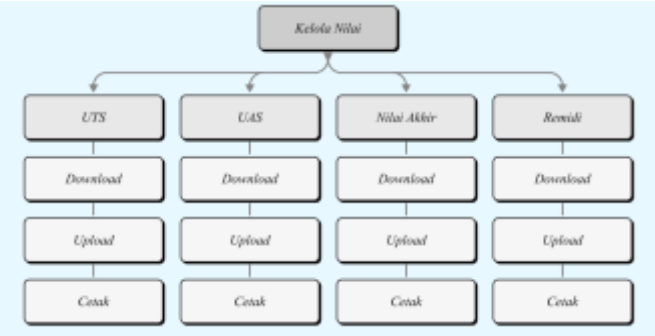

Figure 7. Subdomain of Grade Management

2. Subdomain in the *Grade Management* Domain

This subdomain is responsible for managing processes related to course meetings. These processes include recording lecture materials, submitting requests for leave or substitute meetings, lecturer attendance, student attendance, and attendance recapitulation for both lecturers and students. The structure of this subdomain is illustrated in *Figure 8*.

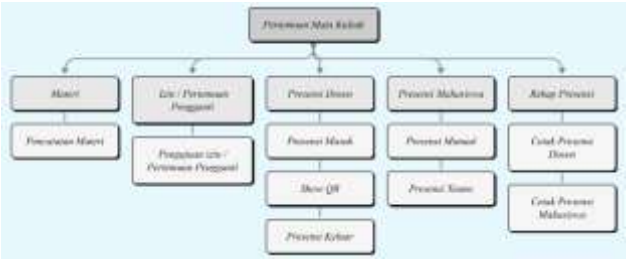

Figure 8. Subdomain of Course Meeting

3. Subdomain in the *Substitute Meeting Management* Domain

This subdomain is designed to handle activities related to managing substitute course meetings. The processes managed include rejecting or approving substitute meeting requests and recording schedule changes. The structure of this subdomain is illustrated in *Figure 9*.

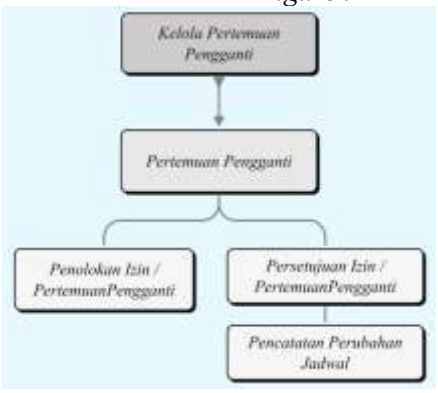

Figure 9. Subdomain of Substitute Meeting Management

3.7. Creating Wireframes

Wireframes illustrate the basic structure and workflow of the *Kelola Nilai* and *Presensi* features. The wireframe's primary focus is placing key elements within the application without considering visual details such as colors or icons. This ensures that the relationships between elements and the system workflow are well-designed. The wireframe was created using a prototyping method based on the previous *Domain-Driven Design (DDD)* analysis.

Figure 10 depicts the wireframe for *Course Meetings*. This wireframe offers an initial representation of the *Course Meetings* page structure, covering both the class and faculty-level class management menus. The primary elements displayed include the meeting list, check-in and check-out attendance buttons, material submission modal, permission request modal for rescheduling meetings, and other relevant components. This wireframe serves as a foundational structure for the interface and workflow development of the *Course Meetings* page.

Figure 10. Wireframe for Course Meetings

3.8. Creating Mockups and Prototypes

Following the completion of wireframes, the design phase progresses to the development of mockups and prototypes. A mockup is an advanced wireframe version incorporating visual elements such as colors, icons, and fonts, resulting in a more realistic design representation. The prototype is designed to validate the system's interface, test user interactions, and ensure that the visual design and system functionality align with the requirements identified in the previous stages.

Figure 11. Mockup of Class Management Index

3.9. Implementation

The final phase of system design using the Domain-Driven Design (DDD) concept is implementing the system. The previously designed system is transformed into a functional application at this stage. The implementation process ensures that each defined domain is properly integrated, allowing the system to meet business requirements while effectively aligning with the predetermined objectives.

1. Class Management

Figure presents the Class Management Index page, which displays a list of classes taught by lecturers, along with relevant information for each class. Additionally, this page provides various services within the Grade Management domain, including features for downloading, uploading, and printing data related to Mid-Semester Exams (UTS), Final Exams (UAS), Final Grades, and Remedial Scores. A Meeting button is also available, enabling lecturers to access the Class Meeting Management domain. Furthermore, a year and semester filter facilitate easy access to classes from previous academic years or semesters.

Figure 12. Class Management Index

Figure 13 illustrates the Upload Grades page, which is one of the key features within the Grade Management domain. This page is designed to assist users in uploading files and efficiently storing student grades within the system.

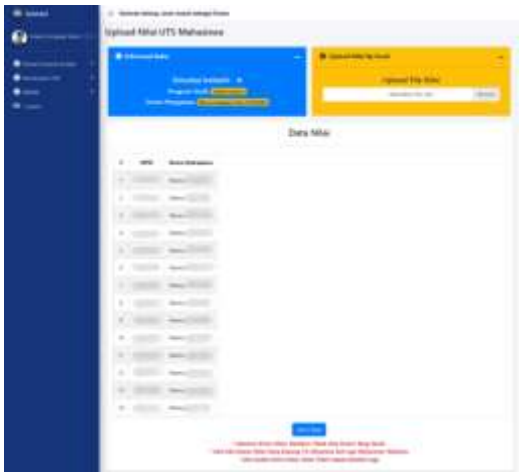

Figure 13. Upload Grades in Class Management Menu

Figure 14 shows the Course Meeting page within the Class Management menu. This page provides various services related to the Course Meeting domain, including lecturer check-in and check-out attendance, lecture material documentation, leave requests, and substitute class management. Additionally, it features a Student Attendance button, leading to the student attendance page, a Student Attendance Recap button directing users to the attendance summary page, and a Lecturer Attendance Print button, enabling users to download lecturer attendance reports based on completed meetings. This page is designed to streamline the management of course meetings for lecturers.

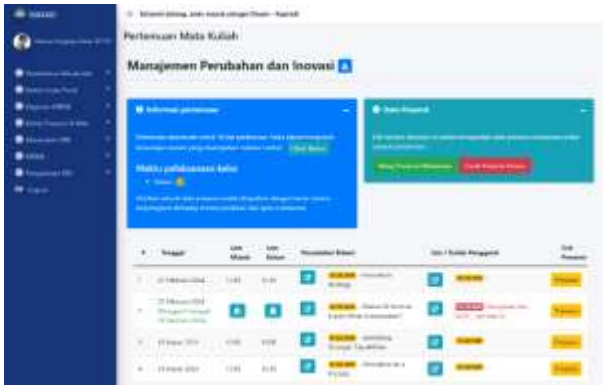

Figure 14. Course Meeting in Class Management Menu

2. Faculty-Wide Class Management

Figure 15 presents the Lecturer Selection page, displaying a complete list of lecturers under the supervision of authorized users such as Head of Study Program (Kaprodi), Academic Administration (TU), and Department Head (Kadep). This page is the first step in accessing and managing the faculty-wide Class Management menu.

Figure 15. Lecturer Selection

Figure 16. Faculty-Wide Class Management Index

Figure 16 displays the faculty-wide Class Management Index page, which presents a list of classes taught by the selected lecturer, along with relevant class information. This page also provides various services within the Grade Management domain, including features for downloading, uploading, and printing UTS, UAS, Final Grades, and Remedial Scores data. Additionally, the Meeting button allows users to access the Class Meeting Management domain. To enhance accessibility, a year and semester filter is included, enabling users to retrieve information on past courses. Moreover, the page features a bar chart that visualizes the number of courses assigned to the selected lecturer.

3. Substitute Class Management

Figure 17 presents the Substitute Class Management Index page, which is accessible exclusively to Academic Administration (TU) and provides various services within the Substitute Class Management domain. The primary functionalities available on this page include Approval of Leave/Substitute Classes, represented by an Approve button, and Rejection of Leave/Substitute Classes, represented by a Disapprove button. This page is designed to help TU staff manage leave requests and substitute class applications from lecturers.

Figure 17. Substitute Class Management Index

3.10. Testing

Testing was conducted using the Black Box Testing method, focusing on the system's functionality. Each subdomain was tested based on the expected outcomes. The results indicate that all functions and services within the subdomains operate correctly and as expected. Therefore, the revamped functions or services were tested successfully. An example of the test results using the Black Box Testing method is shown in Figure 18.

Actor	Service	Scenario	Expected Results	Actual Results	Status
Tata Usaha	Persetujuan Izin / Pertemuan Pengganti	Aktor menekan tombol "Approve" pada data pengajuan yang akan disetujui	Sistem menampilkan formulir pencatatan perubahan jadwal	Sistem menampilkan formulir pencatatan perubahan jadwal dalam bentuk modal	Passed
	Pencatatan Pertemuan Pengganti	Aktor mengisi formulir dan mengirim data pertemuan pengganti	Sistem menyimpan data perubahan pertemuan pengganti	Sistem menyimpan data izin dan mengubah jadwal pertemuan	Passed
	Penolakan Izin / Pertemuan Pengganti	Aktor menekan tombol "Disapprove" pada data pengajuan yang akan disetujui	Sistem membatalkan izin	Sistem membatalkan izin dan mengembalikan data pertemuan seperti semula	Passed

Figure 18. Testing on Manage Substitute Meetings

3.11. Discussion

In terms of system integration, SPKP operates as a separate application from SIAKAD despite sharing the same database. Users must open and switch between two different applications to complete academic tasks—for example, accessing SIAKAD to view class data, then separately opening SPKP to manage grades and attendance. This fragmented workflow causes inefficiency and disrupts user concentration. The new system fully integrates grade and attendance features directly into the SIAKAD interface. The SPKP technology stack has been upgraded to .NET 5 MVC with modern libraries such as Rotativa and OfficeOpenXml, facilitating maintenance and future development with improved maintainability. Functional improvements are clear in grade and attendance management. Grade templates are now dynamic and automatically adjusted to align with the Course Learning Outcomes (CPMK) of each study program, replacing the rigid static templates in SPKP. Based on functionality validation checklists with domain experts, the new system successfully covers 90% of existing SPKP functionalities with significant improvements, plus 10% new features that were not previously available (QR Code attendance, Microsoft Teams integration, workflow approval, dynamic CPMK templates, and dashboard analytics).

4. Conclusion

This study successfully redesigned (revamped) the Grade and Attendance Management features in the Academic Information System (SIAKAD) of Atma Jaya University Yogyakarta (UAJY) using a Domain-Driven Design (DDD) approach, integrated within the Software Development Life Cycle (SDLC) framework. The revamped system demonstrates significant improvements over the legacy SPKP system, achieving 100% test success rate across 112 test scenarios and covering 90% of existing functionalities while introducing 10% additional features not available in the original system. The resulting system encompasses three main domains: Grade Management, Course Meetings, and Substitute Meeting Management. In the Grade Management domain, the system facilitates the administration of student

academic records, including the downloading, uploading, and printing of midterm (UTS), final exam (UAS), final grades, and remedial scores. The Course Meetings domain enables the management of lecturer and student attendance, lecture material documentation, and requests for leave or rescheduled meetings. Meanwhile, the Substitute Meeting Management domain allows administrative staff to approve or reject leave requests and manage substitute meeting schedules. Applying the DDD approach, the system is structured modularly, utilizing key concepts such as bounded context, context mapping, and ubiquitous language to ensure clear domain separation and maintainability. The implementation followed a structured process, beginning with domain modelling, followed by the development of wireframes and mock-ups, and culminating in a functional system that meets user requirements. The system underwent rigorous testing, ensuring all features functioned correctly according to specifications. The revamp process, which included development, testing, and deployment preparations, successfully delivered an enhanced Grade and Attendance Management system. Comprehensive Blackbox Testing, including evaluations by third-party testers, confirmed that the system is fully operational and ready for implementation.

References

[1] D. Santoso, "Pentingnya Revitalisasi Sistem Informasi Akademik di Era Digital," Jurnal Teknologi dan Pendidikan, vol. 3, no. 12, pp. 45-56, 2021.

[2] R. Pratama, "Analisis Risiko Sistem Informasi Akademik," Jurnal Sistem dan Teknologi Informasi, vol. 15, no. 1, pp. 12-15, 2022.

[3] N. Lestari, "Dampak Sistem Informasi Terbaru terhadap Pengelolaan Akademik," Jurnal Riset Teknologi Informasi,, vol. 9, no. 3, pp. 56-70, 2021.

[4] M. Putri, "Peningkatan Kepuasan Pengguna Melalui Pembaruan Sistem Informasi Akademik," Jurnal Teknologi Pendidikan, vol. 11, no. 2, pp. 90-102, 2020.

[5] Subagyo, "Pembaruan Sistem Informasi Akademik di Perguruan Tinggi," Jurnal Sistem Informasi, vol. 7, no. 4, pp. 45-62, 2019.

[6] S. S, "A Study of Software Development Life Cycle Process Models," 10 June 2017. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2988291. [Accessed 8 March 2025].

[7] F. Anisa, F. S. Harahap, H. A. Khosyi, I. P. Sari and Yahfizham, "Pengembangan Software Menggunakan Model SDLC Guna Mencapai Keselarasan dengan Kebutuhan Pengguna," Journal Of Informatics And Busines, vol. 1, no. 4, pp. 229-232, 2024.

[8] M. Safi, A. Samad and I. Djufri, "IMPLEMENTASI MODEL SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC) PADA SISTEM INFORMASI ADMINISTRASI BERBASIS WEB KELURAHAN TOMALOU KOTA TIDORE KEPULAUAN," Indonesian Journal on Information System, vol. 9, no. 2, pp. 242-254, 2024.

[9] S. Millett and N. Tune, Patterns, Principles, and Practices of Domain-Driven Design, Indianapolis: John Wiley & Sonc, Inc, 2015.

[10] O. Özkan, Ö. Babur and M. V. D. Brand, "Domain-Driven Design in Software Development A Systematic Literature Review on Implementation, Challenges, and Effectiveness," ACM Computing Surveys, vol. 55, no. 4, pp. 1-36, 2023.

[11] E. Evans, "Domain-Driven Design: Tackling Complexity in the Heart of Software," 15 April 2003. [Online]. Available: https://msstest.sankuai.com/v1/mss_156341e12eb248878e532dd820706c40/mall-data-init/ddd-en.compressed.pdf. [Accessed 1 Oktober 2024].

[12] R. S. Pressman, Software Engineering: A Practitioner's Approach, New York: McGraw-Hill Companies, Inc., 2015.

[13] A. Rachmadani, "Pemrograman Web Menggunakan HTML dan CSS," Jurnal Teknologi Informasi, vol. 15, no. 3, pp. 87-95, 2021.

[14] A. Alharbi and M. Zain, "The Role of Cascading Style Sheets in Web Development," International Journal of Computer Applications, vol. 175, no. 7, pp. 12-17, 2021.

[15] A. Yani and B. Saputra, "RANCANG BANGUN SISTEM INFORMASI EVALUASI SISWA DAN," JURNAL PETIR, vol. 11, no. 2, September 2018.

[16] F. A. Juyuspan and P. Oktivasari, "PENGEMBANGAN WEBSITE DINAMIS MENGGUNAKAN ASP.NET MVC DAN SQL SERVER DENGAN METODE RAD," Jurnal Nasional Informatika dan Teknologi Jaringan, vol. 2, pp. 16-21, 2017.

[17] R. Wibowo and A. Santoso, "Pengaruh Penggunaan Framework Bootstrap terhadap Efisiensi Pengembangan Web Responsif.," Jurnal Teknologi Informasi, vol. 12, no. 2, pp. 34-41, 2021.

[18] R. M. Yanwar, A. B. Putra and R. Hadiwiyanti, "RANCANG BANGUN SISTEM INFORMASI MANAJEMEN SUMBER DAYA MANUSIA BERBASIS DESKTOP DI CV.DIRACO," Jurnal Mahasiswa Teknik Informatika, vol. 8, no. 4, pp. 5592 - 5599, 2024.

[19] Context Mapper, "What is Context Mapper?," Context Mapper, 2024. [Online]. Available: https://contextmapper.org/. [Accessed 28 Oktober 2024].

[20] PlantUML, "PlantUML at a Glance," PlantUML, [Online]. Available: https://plantuml.com/en/. [Accessed 30 Oktober 2024].

[21] S. Sudrajat, "Mengenal Software Figma: Fungsi, Fitur, dan Keunggulannya," IDN TIMES, 21 Agustus 2023. [Online]. Available: https://www.idntimes.com/tech/trend/sony-sudrajat/mengenal-software-figma-c1c2. [Accessed 30 Oktober 2024].