EVENTSWIFT: AGILE EVENT PLANNING AND COORDINATION SOLUTION FOR COLLEGE OF ENGINEERING AND COMPUTER TECHNOLOGIES

WENCY N. CASTRO, WESLEYAN UNIVERSITY - PHILIPPINES
CARL JESUS B. LOMAAD, WESLEYAN UNIVERSITY - PHILIPPINES
VONN HOWELL M. VILLAROMAN, WESLEYAN UNIVERSITY - PHILIPPINES
JHON JESTER P. CARLOS, WESLEYAN UNIVERSITY - PHILIPPINES

ABSTRACT

Planning events at a university involves several steps, usually requiring approval from multiple offices. At Wesleyan University-Philippines, this process was mostly done using printed forms passed from one office to another. While it worked in theory, it often caused delays, confusion, and lost paperwork. To address these issues, we developed EventSwift, a web application built with ASP.NET that moves the entire proposal and approval process online. The system allows users to fill out a digital form, upload required files, and send everything through a set approval flow. Each office sees only the proposals it needs to review, and users can track the status of their request through a built-in document tracker. Offices can approve or return proposals for changes, and the final step routes the proposal to the university president for sign-off. Our goal was to make the process more manageable for everyone involved. Launching this system, we will see faster approvals and fewer lost forms. The system will keep all involved offices in the loop and makes the process less stressful for both organizers and administrators. The system will show that moving from paper to digital is not just about convenience. It simplifies event management within a department and makes it easier to coordinate with others.

Keywords: event approval workflow, digital document tracking, multi-office coordination, role-based access, paperless university administration, scrum development

INTRODUCTION

Planning an event at a university might sound simple, but anyone who has tried it knows how quickly things can get complicated. At Wesleyan University-Philippines, the process used to involve a lot of paperwork, walking from office to office, and waiting for signatures. Each event proposal had to be endorsed by a department, checked for academic alignment, approved for venue and budget, and finally signed off by the university president. With every step handled on paper, it was easy for documents to get lost, for people to miss updates, and for organizers to feel left in the dark about where their proposal stood.

This is not just a local problem. Many studies have pointed out that universities everywhere struggle with slow and confusing administrative processes. Smith et al. (2022) found that faculty and staff spend a surprising amount of time on paperwork, often at the expense of more meaningful work. Patel (2021) noted that without a central place to manage documents, it is tough to keep track of what has been approved and what still needs attention. As a result, proposals can get stuck, deadlines are missed, and people get frustrated.

In recent years, more universities have started using digital tools to help with these challenges. Researchers like Gonzalez et al. (2023) have shown that online systems can help reduce the time spent on approvals and make it easier to follow university policies. Lee and Kim (2021) also found that when requests go through a clear, step-by-step process online, it is easier to see who needs to do what and when. Good document management systems keep everything in one place, so nothing gets lost and all reviewing offices knows where to look.

Given these findings, we decided it was time for a change at WU-P. We built EventSwift, a web application that puts the whole event proposal process online. With this system, it makes event planning at WU-P easier and less stressful for everyone involved. In this paper, we explain why we built this, how it works, and what we have learned since putting it into use. We also look at how our experience fits with what other universities have found about digital workflow systems, and what this could mean for the future of event management in higher education.

Event planning in the College of Engineering and Computer Technologies at Wesleyan University-Philippines depended on manual paper processes that created real hurdles for students, faculty, and staff.

1. What delays happen when event proposals move physically from one office to another in the college?
2. How does lacking a central place to check proposal status cause constant follow-ups and poor communication for organizers?
3. How do scattered paper records across offices make policy compliance and scheduling difficult?
4. How may the system be assessed in terms of the ISO 25010 Software Quality Model criteria?

METHODS

Research Design

This study used a descriptive-developmental research design to guide the creation and evaluation of the EventSwift: Agile Event Planning and Coordination Solution. The approach focused on systematically designing, developing, and assessing the system to ensure it met the needs of faculty and staff in the College of Engineering and Computer

Technology at Wesleyan University-Philippines. The design allowed for iterative feedback and continuous improvement throughout the development and testing phases.

Research Locale

The research was conducted within the College of Engineering and Computer Technology at Wesleyan University-Philippines, located in Cabanatuan City, Nueva Ecija. This setting provided a relevant environment for system development, demonstration, and user evaluation, as the participants were directly involved in event management and approval processes within the department.

Research Instruments

In the development of EventSwift: Agile Event Planning and Coordination Solution, the following hardware and software resources were utilized to support the creation, testing, and deployment of the system.

Software:

a. Visual Studio 2019: Used as the main integrated development environment (IDE) for writing, editing, and debugging the application's code. This tool provided support for ASP.Net MVC 5 development and streamlined the coding workflow.
b. SQL Server Management Studio 2022: Managed the Microsoft SQL Server database, which stored user accounts, event proposals, approval records, and document files.
c. Bootstrap 5: Provided the framework for building responsive and user-friendly web interfaces, ensuring the application was accessible on various devices and screen sizes.
d. ASP.Net MVC 5: Served as the core framework for developing the web application, allowing for a clear separation of concerns and efficient management of user interactions, business logic, and data access.
e. GitHub: Used for version control and collaboration, enabling the development team to track changes, manage branches, and coordinate work efficiently.
f. Figma: Employed for designing wireframes and mockups of the user interface. Figma's collaborative features allowed for real-time feedback and adjustments during the design phase.
g. Draw.io: Used to create system flowcharts, helping visualize the application's logic flow and user journeys.
h. Lucid.app: Utilized for drawing entity-relationship diagrams (ERDs), which guided the structure and relationships within the database.

Hardware:

a. Computer systems: Desktop and laptop computers were used for all phases of development, from design and coding to testing and deployment.
b. Internet connectivity: Stable internet access was necessary for accessing cloud-based tools like GitHub, Figma, and Lucid.app, as well as for team collaboration and system deployment.

These materials ensured that the development process was organized, collaborative, and aligned with current web application development practices.

Statistical Tool

To evaluate the developed system, the researchers will utilize a structured survey questionnaire based on the ISO/IEC 25010 standards for software product quality. The instrument is designed to assess five key quality characteristics: Functionality, Performance Efficiency, Compatibility, Reliability, and Security. Each characteristic will include multiple specific items that aim to measure the system's performance across these dimensions.

Each item in the questionnaire will be rated using a 5-point Likert scale, where the numerical responses are interpreted as follows:

Weighted Mean	Likert Scale	Verbal Description
4.20 - 5.00	5	Strongly Agree
3.40 - 4.19	4	Strongly Agree
2.60 - 3.39	3	Neutral
1.80 - 2.59	2	Disagree
1.00 - 1.79	1	Strongly Disagree

Sampling

The evaluation focused on students from the College of Engineering and Computer Technology at Wesleyan University-Philippines. Since the college includes several different programs, stratified sampling was used to ensure proportional representation across all programs. This means the student population was divided into groups based on their program, and participants were selected from each group according to its size. This approach helped make sure that the survey results fairly reflected the views of students from every program.

To determine the minimum number of students to include, the researchers used Raosoft's online sample size calculator. They set a 5% margin of error and a 95% confidence level, using the total number of enrolled CECT students. The resulting figure was the target sample size for the survey.

In addition to students, purposive sampling was used to select a panel of IT experts. These experts, mainly instructors and system developers from the IT department,

were invited to review the system's content and provide expert evaluations on its quality.

By combining stratified sampling for students with purposive sampling for experts, the study aimed to gather broad user feedback as well as detailed technical insights, supporting a thorough evaluation of the system.

Statistical Tool

The following statistical tools will be used to evaluate the data collected from respondents:

Mean: To compute the average responses for each system quality criterion, such as Functionality, Performance Efficiency, Compatibility, Reliability, and Security.

Cronbach's Alpha: To measure the internal consistency and reliability of the questionnaire items across all dimensions.

Aiken's V: To assess the content validity of the survey instrument, based on evaluations from the panel of IT experts.

T-test (optional/comparative): If the study requires comparing the perspectives of experts and student respondents, the T-test may be used to determine if there are significant differences between the two groups in their assessment of the system.

Procedure

The development of the EventSwift web application followed the Scrum methodology, which is an agile approach that breaks work into short, focused intervals called sprints. This method was chosen because it encourages teamwork, transparency, and regular feedback, all of which are important when building a system that needs to fit real users' needs.

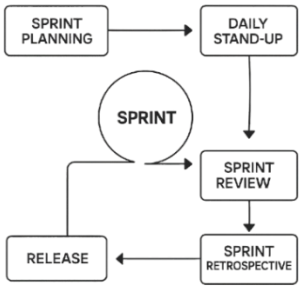

Figure 1. Scrum methodology process flow. The diagram illustrates key Scrum stages: Sprint Planning, Daily Stand-Up, Sprint execution, Sprint Review, Sprint Retrospective, and Release. Arrows indicate the iterative cycle used to continuously improve and deliver updates.

I. Requirement

The goal of developing EventSwift is to create a system that simplifies managing event proposals and approvals at Wesleyan University–Philippines, particularly within the College of Engineering and Computer Technology. We adopted an Agile approach using the Scrum method, which allowed us to remain flexible and responsive to user feedback throughout the development process.

During the planning phase, we identified several key features the system needed:

1. User Roles and Authentication: The system supports three main user types: event proposers, involved offices, and administrators. Each role has access to specific features based on their responsibilities, and users log in with their institutional credentials to ensure security.

2. Event Proposal Submission and Tracking: Users can submit event proposals with details like title, purpose, venue, date, and budget. They can upload necessary documents and track the status of their proposals without needing to follow up in person.

3. Multi-Level Approval Workflow: Proposals are routed through a series of approvals, starting with the department coordinator and moving through various offices before reaching the president. Users can monitor progress at each stage.

4. Document Management and Feedback: All documents and approval notes are stored in the system for easy access. If a proposal needs revisions, the system notifies the proposer and provides feedback on what needs to be fixed.

5. Calendar Integration: Once approved, events are automatically added to the university calendar, helping everyone stay informed about upcoming activities and avoid scheduling conflicts.

6. User -Friendly Design: Built with ASP.Net MVC, the system features a responsive design that ensures a consistent user experience across different devices.

7. Security and Data Integrity: Role-based access control limits what users can see and do based on their roles. The system also employs secure protocols to protect data and prevent unauthorized changes.

These requirements guided our development process, and we refined them based on user input during sprint reviews and testing. This iterative approach helped us create a system that enhances communication and simplifies event approval management.

II. Quick Design

The main flowchart diagrams used in EventSwift, which highlight its main functions and procedures, are shown in the sections that follow: The following figure displays the SYSTEM FLOWCHART:

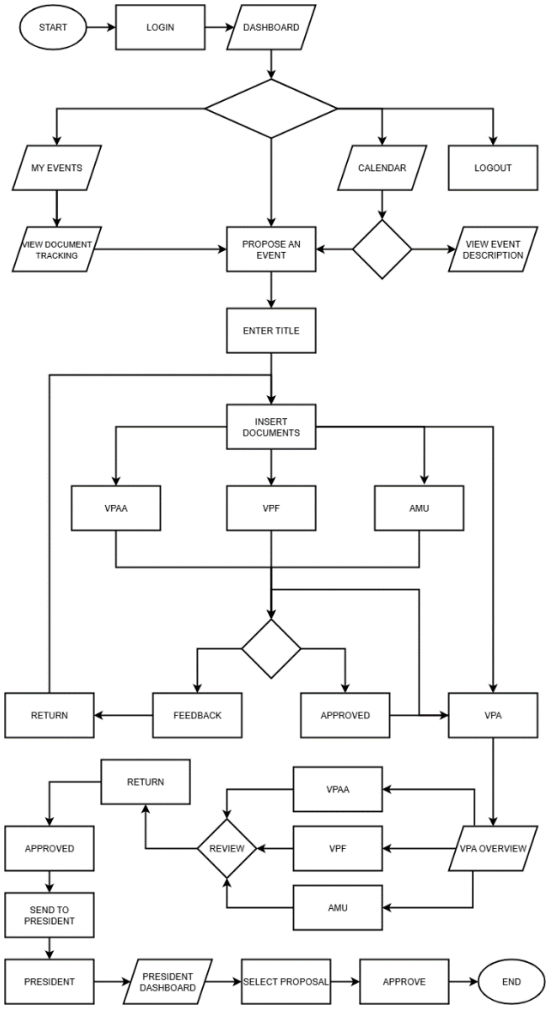

Figure 2. System Flowchart. This flowchart shows how users submit and track event proposals in EventSwift. After logging in, users can create or edit proposals, upload documents for each required office, and see real-time updates as offices review, approve, or return items for revision. Once all approvals are received, the event is finalized by the President and added to the user's calendar. The process keeps all steps clear and organized from start to finish.

The following figure displays the Entity-Relationship Diagram:

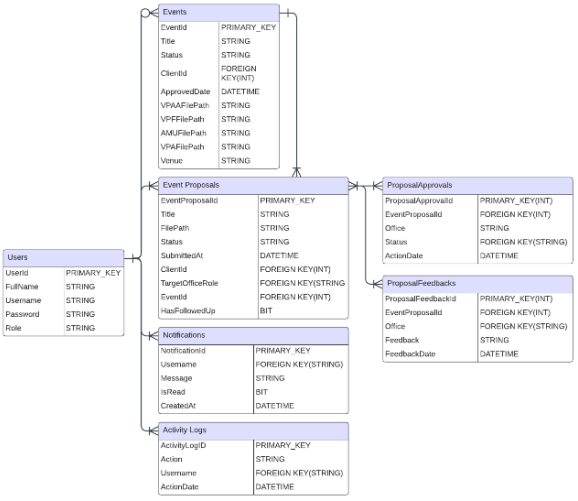

Figure 3. Entity-Relationship Diagram. This ERD shows the main structure of the EventSwift database for managing event proposals and approvals. It consists of seven tables: Users storing user details; Events holding approved event information; Event Proposals tracking submitted proposals and their documents; Proposal Approvals recording approval statuses; Proposal Feedback containing office comments; Notifications managing system messages; and Activity Logs tracking user actions. The tables are connected to support a clear, organized workflow, ensuring communication and accountability throughout the event approval process.

III. Prototype

The figures below show the user interface of EventSwift, designed to be simple and easy to use for students, faculty, and staff. The system allows users to submit event proposals, track approvals, and view feedback in a clear, organized way. It is responsive and works well on both desktop and mobile devices, making event management convenient across different platforms. The design focuses on accessibility and clear communication to support an efficient approval process.

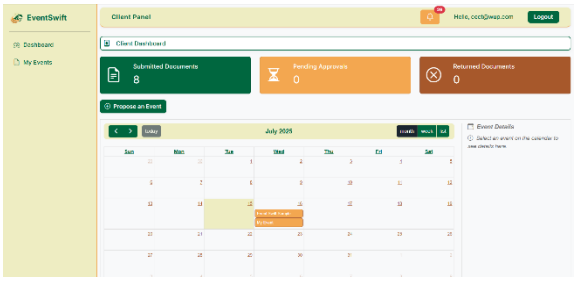

Figure 4. Client Dashboard

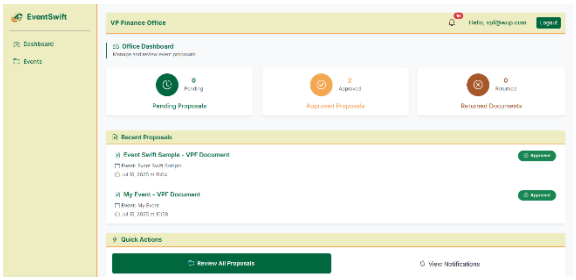

Figure 5. Office Dashboard

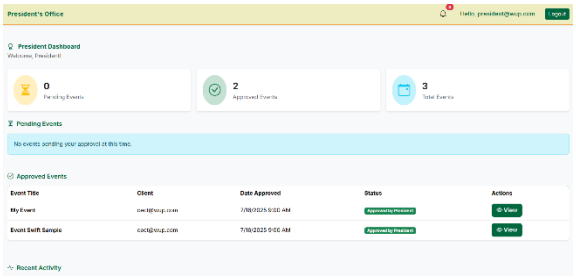

Figure 6. President Dashboard

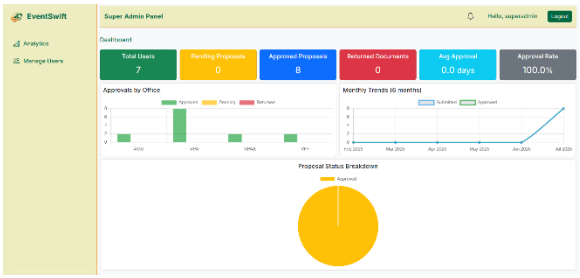

Figure 7. Admin Dashboard

Ethical Basis

This study followed ethical guidelines to protect the rights and privacy of all participants. Everyone who took part gave their informed consent and knew how their information would be used. All data collected were kept confidential and stored securely, with access limited to the research team. The system was designed with proper access controls to protect user information. Participants were not pressured to join and their participation was voluntary. Throughout the project, transparency was maintained by keeping stakeholders informed and involved. These steps helped ensure the study was conducted fairly and responsibly.

RESULTS AND DISCUSSIONS

A. **System Developmen**t

The development of EventSwift: Agile Event Planning and Coordination Solution produced several user interfaces designed to streamline event proposal and approval management. Each figure below presents a key part of the system and highlights how users interact with it, demonstrating how EventSwift supports efficiency, transparency, and secure processing across all roles

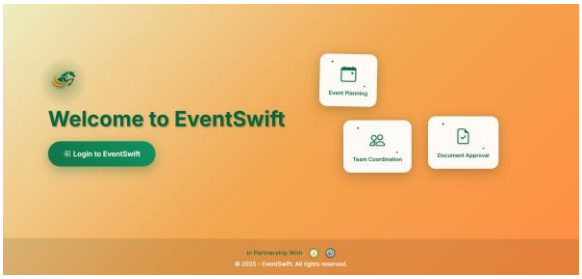

Figure 8. Splash Screen

Figure 8 shows the splash screen where users first arrive. It is simple and welcoming, displaying the EventSwift logo and a login button that allows users to begin using the system. This introductory screen establishes the system's identity and serves as a gateway for all users.

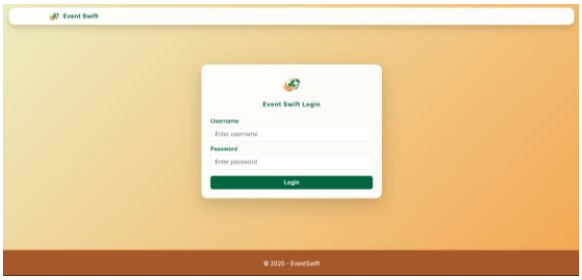

Figure 9. Login Screen

Figure 9 presents the login screen, where users enter their credentials to access the system. Once verified, they are directed to their designated dashboards based on their user roles. This ensures secure and role-based access to the system's features and functions.

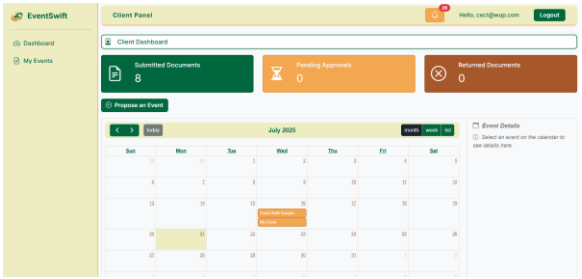

Figure 10. Client Dashboard

Figure 10 presents the client dashboard, which is the main interface for students or event proposers. It displays submitted proposals, pending approvals, and returned items. The dashboard also provides easy access to create new proposals and check scheduled events through the integrated calendar feature.

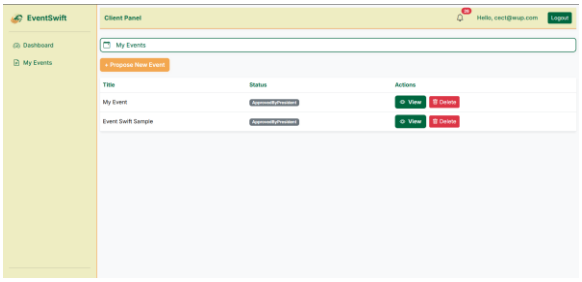

Figure 11. Propose New Event

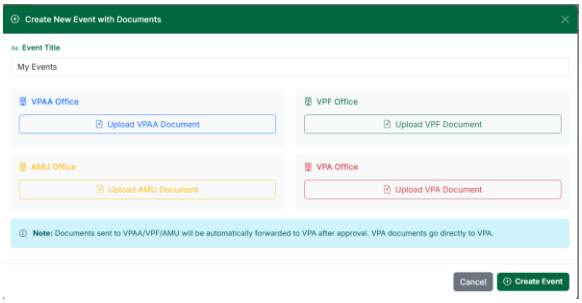

Figure 12. Create New Event with Documents

Figure 11 and 12 show the screen where clients create new event proposals. Users input the event title, details, and upload the required documents addressed to specific approving offices such as the VPAA, VPF, AMU, and VPA. This section guides users through a structured submission process, minimizing errors and incomplete entries.

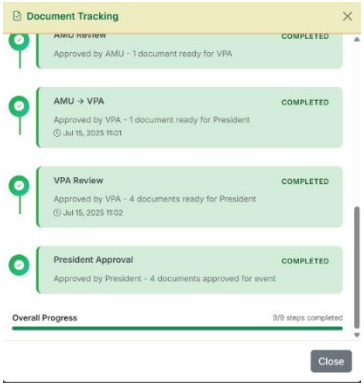

Figure 13. Document Tracking Panel (Client View)

Figure 13 presents the document tracking panel for clients. It clearly shows the progress of each document through the approval process, indicating which offices have approved or returned it for revision. This feature provides transparency and keeps proposers informed without needing manual follow-ups.

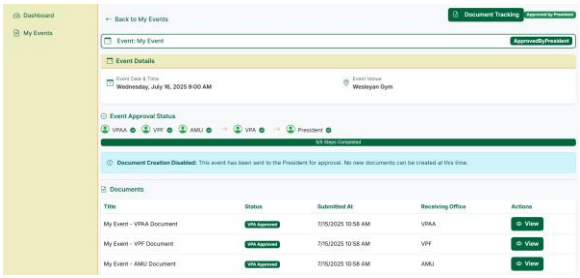

Figure 14. Proposal Details View

Figure 14 summarizes the details of an event proposal, including the event title, date, and venue, as well as the list of offices that have approved it. It also provides access to related documents, making it easier for users to review and manage their submissions.

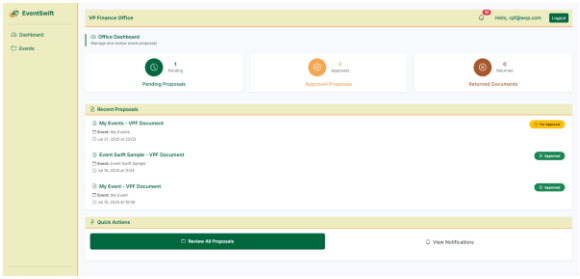

Figure 15. Office Staff Dashboard

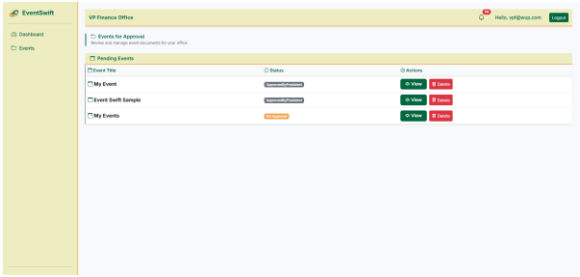

Figure 16. All Proposals

Figure 15 and 16 illustrate the dashboard interface for office personnel, using the VP Finance Office as an example. It displays a list of pending proposals and notifications, allowing staff to monitor their assigned reviews efficiently. This helps ensure that proposals move through the approval stages promptly.

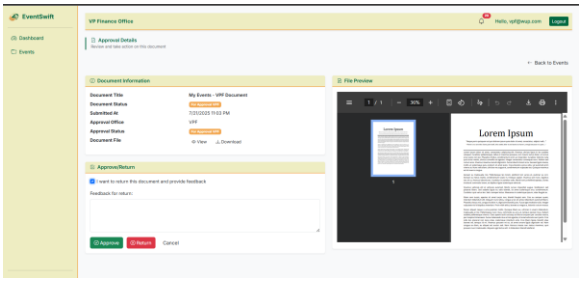

Figure 17. Document Review and Approval Workflow

Figure 17 demonstrates how office users review and approve event proposal documents. They can preview files, add comments, and either approve or return proposals with recommendations. Approval requires the user's password and electronic signature, which confirms the validity of the action and updates the document's status in real time.

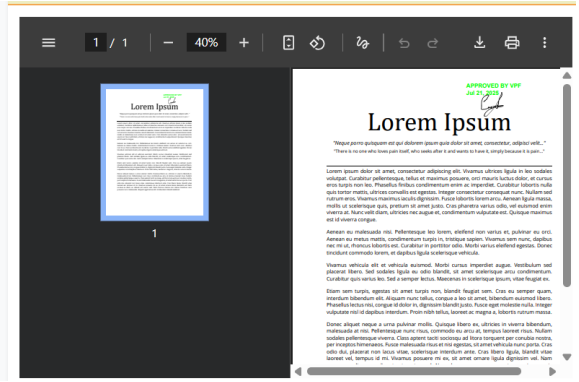

Figure 18. View Document

Figure 19. Resubmit Document

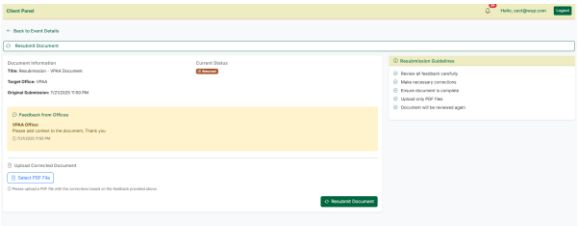

Figure 20. Returned Proposal with Feedback

Figure 18, 19, and 20 show what happens when a proposal is returned with feedback. The client receives the office's comments and can upload a revised document for re-evaluation. This feedback mechanism supports collaboration between offices and proposers while maintaining clear communication throughout the review process.

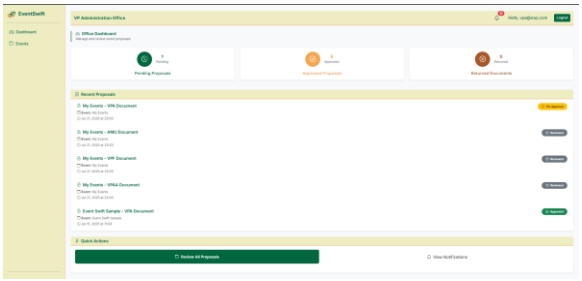

Figure 21. VPA Dashboard

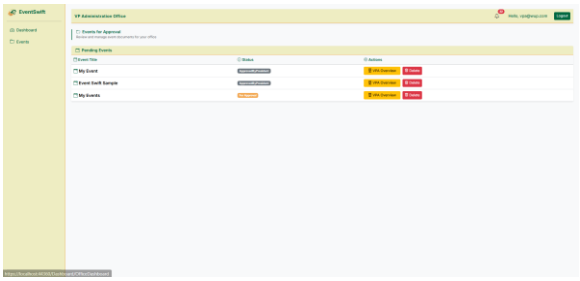

Figure 22. VPA Event Proposals

Figure 21 and 22 display the dashboard for the Vice President for Administration (VPA). It provides an overview of proposals and approvals, with the ability to set the official event date and venue once all necessary sign-offs are completed. This feature ensures proper scheduling and coordination of approved events.

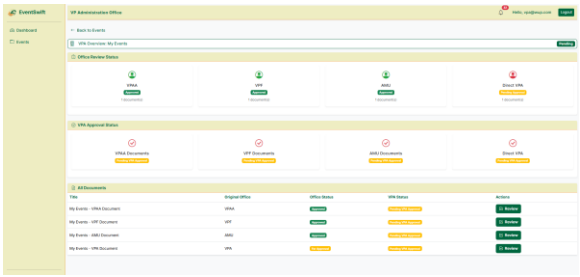

Figure 23. VPA Approval Interface

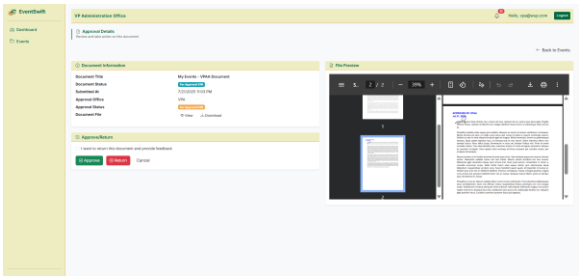

Figure 24. VPA Approval Details

Figure 23 and 24 present the VPA's approval interface, where the VPA can finalize event details, confirm approvals, and forward the completed proposal to the President for final review. This screen ensures that only fully endorsed proposals move to the final approval stage.

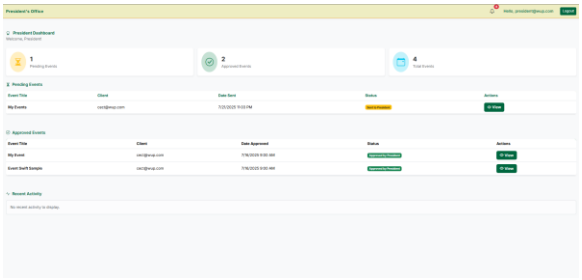

Figure 25. President Dashboard

Figure 25 shows the President's dashboard, which focuses on proposals awaiting final approval. The layout is simplified, displaying only essential details to assist in quick and informed decision-making.

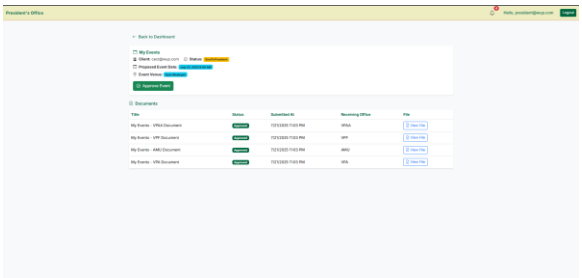

Figure 26. President Approval Screen

Figure 27 presents the President's approval screen. The President can review all submitted documents, download files if necessary, and approve the event. Once approved, the event is automatically added to the official calendar, marking it as fully authorized.

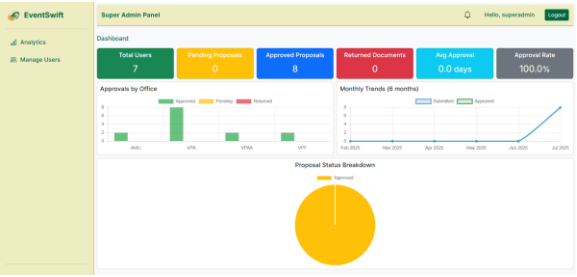

Figure 28. Super Admin Dashboard

Figure 28 displays the Super Admin dashboard, which provides administrators with an overview of system-wide activity. It includes information such as user counts, event proposal statuses, and general system health indicators. This view helps administrators monitor the system's performance and manage users efficiently.

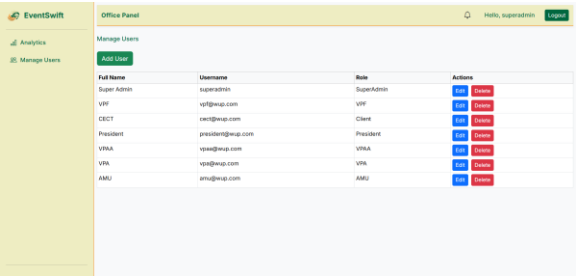

Figure 29: User Management

Figure 29 shows the user management panel where the Super Admin can add, update, or remove users and assign their roles. It's the control center for managing access.

B. End-User Evaluation

A group of five experts comprises of 5 IT Expert from College of Engineering and Computer Technology of Wesleyan University- Philippines reviewed the research instrument and evaluate its content validity before it was distributed to respondents. After applying Aiken's V validity test, the instrument received a V-value of 0.93, confirming its validity.

To assess the internal consistency of the research instrument, the questionnaire was administered to a sample of 10 IT students. Following a demonstration of the system, the students employed the questionnaire to evaluate the system's performance. The instrument is structured around five key criteria, as described in ISO 25010: Functionality, Performance Efficiency, Compatibility Reliability and Security, encompassing a total of 25 specific items. The collected responses were subsequently analyzed using Cronbach's Alpha. Cronbach's Alpha coefficient, calculated via SPSS software, yielded a value of 0.929, which signifies a high degree of internal consistency for the research instrument.

Overall System User Evaluation Descriptives

The table below summarizes the respondents' overall evaluation of EventSwift: Agile Event Planning and Coordination Solution in terms of the ISO 25010 software quality model criteria. The assessment involved 250 student respondents from the College of Engineering and Computer Technology at Wesleyan University–Philippines, consisting of 136 BSIT, 80 BSCpE, and 34 BSEcE students from various year levels.

TABLE 1

Summary of the Overall System User Evaluation Descriptives based on ISO 25010 quality characteristics.

User Evaluation System Criteria	Weighted Mean	Verbal Description	Verbal Interpretation
Functionality	4.3915	Strongly Agree	Highly Functional
Performance Efficiency	4.3302	Strongly Agree	Very Efficient
Compatibility	4.3820	Strongly Agree	Very Compatible
Reliability	4.3915	Strongly Agree	Very Reliable
Security	4.3725	Strongly Agree	Very Secured

As shown in Table 1, the system achieved consistently high ratings across all evaluation criteria, with an overall weighted mean of 4.7136, interpreted as Strongly Agree and described as Highly Acceptable. The highest rating was observed in Compatibility (4.7536), suggesting that EventSwift functions effectively across different devices and browsers. Reliability (4.7200) and Functionality (4.7168) also received high ratings, confirming that users found the system dependable and capable of meeting its intended purpose. The rating for Performance Efficiency (4.6640) indicates that the system performs tasks smoothly

and promptly. These results reflect users' strong confidence in the system's adaptability, effectiveness, security, and ease of use, demonstrating an overall high level of satisfaction with EventSwift's performance and design.

TABLE 2

User evaluation results for the Functionality criterion.

Numerical Scale	Weighted Mean Interval Scale	Verbal Description	Verbal Interpretation
1	4.20 - 5.00	Strongly Agree	Highly Functional
2	3.40 - 4.19	Agree	Functional
3	2.60 - 3.39	Neutral	Moderately Functional
4	1.80 - 2.59	Disagree	Fairly Functional
5	1.00 - 1.79	Strongly Disagree	Not Functional

As shown in Table 2, the Functionality criterion obtained a weighted mean of 4.7168, verbally described as Strongly Agree and interpreted as Highly Functional. This indicates that all core features of EventSwift operated as intended and met user expectations. Respondents agreed that the system effectively facilitated event proposal creation, document uploading, and approval tracking. The high mean value shows that users found the system dependable and responsive in handling essential tasks without noticeable errors or missing features.

TABLE 3

User evaluation results for the Performance Efficiency criterion.

Numerical Scale	Weighted Mean Interval Scale	Verbal Description	Verbal Interpretation
1	4.20 - 5.00	Strongly Agree	Highly Efficient
2	3.40 - 4.19	Agree	Efficient
3	2.60 - 3.39	Neutral	Moderately Efficient
4	1.80 - 2.59	Disagree	Fairly Efficient
5	1.00 - 1.79	Strongly Disagree	Not Efficient

As presented in Table 3, Performance Efficiency recorded a weighted mean of 4.6640, interpreted as Very Efficient. Users strongly agreed that the system performs tasks quickly and without unnecessary delays. This reflects the optimized design and smooth flow of processes implemented in the web application. The system's responsiveness during login, document submission, and approval tracking demonstrates efficient processing and stable performance even when accessed by multiple users simultaneously.

TABLE 4

User evaluation results for the Compatibility criterion.

Numerical Scale	Weighted Mean Interval Scale	Verbal Description	Verbal Interpretation
1	4.20 - 5.00	Strongly Agree	Highly Compatible
2	3.40 - 4.19	Agree	Compatible
3	2.60 - 3.39	Neutral	Moderately Compatible
4	1.80 - 2.59	Disagree	Fairly Compatible
5	1.00 - 1.79	Strongly Disagree	Not Compatible

Table 4 shows that Compatibility achieved the highest weighted mean of 4.7536, categorized as Strongly Agree and interpreted as Very Compatible. This suggests that EventSwift performs reliably across different devices and browsers while maintaining consistent layout and functionality. Its compatibility ensures accessibility for both students and office personnel regardless of their platform. The result confirms that using technologies such as Bootstrap and ASP.NET MVC successfully achieved cross-platform responsiveness.

TABLE 5

User evaluation results for the Reliability criterion.

Numerical Scale	Weighted Mean Interval Scale	Verbal Description	Verbal Interpretation
1	4.20 - 5.00	Strongly Agree	Highly Reliable
2	3.40 - 4.19	Agree	Reliable
3	2.60 - 3.39	Neutral	Moderately Reliable
4	1.80 - 2.59	Disagree	Fairly Reliable
5	1.00 - 1.79	Strongly Disagree	Not Reliable

As indicated in Table 5, Reliability garnered a weighted mean of 4.7200, verbally described as Strongly Agree and interpreted as Very Reliable. The consistency of system operations shows that EventSwift can perform repetitive tasks without failure or data loss. Respondents recognized that the approval process, notification updates, and tracking features remained stable throughout use. This level of reliability reinforces the system's ability to handle critical event-management functions with minimal downtime or technical errors.

TABLE 6

User evaluation results for the Security criterion.

Numerical Scale	Weighted Mean Interval Scale	Verbal Description	Verbal Interpretation
1	4.20 - 5.00	Strongly Agree	Highly Secured
2	3.40 - 4.19	Agree	Secured
3	2.60 - 3.39	Neutral	Moderately Secured

4	1.80 - 2.59	Disagree	Fairly Secured
5	1.00 - 1.79	Strongly Disagree	Not Secured

As shown in Table 6, Security obtained a weighted mean of 4.3725, interpreted as Very Secured. Respondents strongly agreed that the system protects data through password authentication, role-based access, and electronic signature verification. These features increase user confidence, especially since the platform handles sensitive event-proposal documents. The result demonstrates that the developers implemented effective security measures that preserve user privacy and maintain system integrity.

CONCLUSION

The study shows that manual event proposals in the College of Engineering and Computer Technologies at Wesleyan University-Philippines run into clear delays from passing papers between offices, but EventSwift fixes that with its digital workflow. Organizers without a central tracking spot end up chasing status updates and dealing with bad communication, though EventSwift gives real-time updates and notifications to keep people in the loop. Paper records spread across offices make it tough to follow policies and schedules, but EventSwift pulls everything into one digital place for easy checking. The system also scored high on ISO 25010 standards: functionality at 4.72, performance efficiency at 4.66, compatibility at 4.75, reliability at 4.72, and security at 4.37, which proves it works well for college events.

RECOMMENDATIONS

The results point to rolling out EventSwift across all Wesleyan University-Philippines departments, not just the College of Engineering and Computer Technologies, to handle those manual delays, tracking problems, and scattered records everywhere. Add integration with school email for instant notifications, e-signatures you can place anywhere in documents, and automatic merging of all files into one PDF before the VPA or President approves, so they see everything as a single package. Include analytics feature so students can pull data on event attendance, participation trends, and feedback to plan better next time. Keep up regular maintenance and updates to make sure it stays reliable and secure while cutting down on paper use.

ACKNOWLEDGMENT

The researchers sincerely thank Almighty God for providing them with the strength, guidance, and perseverance to accomplish this study despite personal and academic challenges. They would also like to express their deepest gratitude to Mrs. Dolores Baldedara, their Capstone Project Adviser, for her invaluable guidance, unwavering support, and insightful feedback throughout the development of EventSwift: Agile Event Planning and Coordination Solution. Sincere appreciation is extended to Mr. Leo Santos, their statistician, for his assistance in processing and interpreting the data essential to this research. The researchers gratefully acknowledge Mr. Harry Role for approving the EventSwift project during the title defense, making its implementation possible. They also extend they're thanks to the CECT Faculty of Wesleyan University-Philippines for facilitating access to IT professionals who evaluated the system and provided helpful feedback on its functionality, compatibility, and security. Lastly, heartfelt appreciation is given to their families and friends for their continuous encouragement, patience, and emotional support throughout the duration of this capstone journey.

REFERENCES

Cheng, L., & Wu, Y. (2020). Role-based access and sequential routing in workflow systems. Computers & Education, 148, 103794. https://doi.org/10.1016/j.compedu.2019.103794

Gonzalez, R., Smith, J., & Patel, T. (2023). Digital transformation in higher education institutions. Education and Information Technologies, 28(2), 1231–1250. https://doi.org/10.1007/s10639-022-11021-9

Lee, H., & Kim, S. (2021). Multi-level approval systems in university administration. International Review of Administrative Sciences, 87(4), 911–928. https://doi.org/10.1177/0020852319896761

Patel, T. (2021). Document management strategies for universities. Information Management Journal, 55(3), 47–58.

Smith, J., Brown, A., & Lee, H. (2022). Workflow automation in higher education: Reducing administrative burden. Journal of Educational Technology, 39(1), 45–59.