FUNCTIONAL SPECIFICATION DOCUMENT

E-skolarian: PUPSRC Document Management System (ODMS) - Submission Tracker & Version Control Module

DOCUMENT VERSION 2.9

June 21, 2025

AUTHORS

Name	Role	Department
Aliño, Mark B.	Business Analyst	BSIT 3-1
Austria, Jonald Kiel M.	Quality Assurance	BSIT 3-1
Barroso, Lauriene Jade N.	Project Manager	BSIT 3-1
Juan, Zofia Dennise Z.	Quality Assurance	BSIT 3-1
Negrite, Red Aaron P.	Developer	BSIT 3-1

DOCUMENT HISTORY

Date	Version	Document Revision Description	Document Author
04/16/2025	1.0	Created the initial version of the Functional Specification Document (FSD) Added Introduction, Purpose of the Document, Scope of the Document, Related Documents, Terms/Acronyms and Definitions, Other System Requirements/Non-Functional Requirements, Risks and Assumptions, and Mock-up and Use Case	Aliño, Barroso
04/18/2025	1.1	Added Change Impacts	Austria
04/19/2025	1.2	Added Reporting Requirements, Exception Handling / Error Reporting	Austria
04/20/2025	1.3	Modified Terms/Acronyms and Definitions Added Purpose/Description	Barroso
04/23/2025	1.4	Added Module/Solution Overview Modified Related Documents	Barroso
04/25/2025	1.5	Added Project Scope, User Roles and Responsibilities / Authority Requirements, System Dependencies, Functional Specifications, Functional Requirements, Field Level Specifications	Juan, Aliño
04/26/2025	1.6	Added Context Diagram/Interface Diagram /Data Flow Diagram, Application Screen Flow, Sitemap, Process Flow	Aliño
04/27/2025	1.7	Added Integration Requirements	Barroso
04/29/2025	1.8	Added System Configurations, Data Migration/Conversion	Negrite

		Requirements, Data Conversion Strategy, Data Conversion Preparation, Data Conversion Specification	
05/11/2025	1.9	Modified Related Documents, specifically Project Roadmap and Concept Paper, due to schedule re-adjustments	Barroso
05/11/2025	2.0	Added some information in Integration Requirements	Aliño
05/23/2025	2.1	Added changes in Context Flow Diagram, Data Flow Diagrams and System Flow Diagrams	Aliño
05/26/2025	2.2	Modified Diagrams based on the updated system module flow and Updates in Mock-up	Aliño
05/27/2025	2.3	Changing Student Organization President into Student Organization and Database Naming	Aliño
05/28/2025	2.4	Removes Follow-up features and all contents about it.	Aliño
06/04/2025	2.5	Revision of Diagrams and change Resubmission to Return of Documents	Aliño
06/06/2025	2.6	Updated Purpose of the Document, Scope of Document, Purpose/Description according to the changes made within the System (e.g. replacement of the status “resubmits” with “return”) Updated Related Documents by replacing the outdated files/attachments with the updated ones.	Barroso
06/12/2025	2.7	Refinement of Context Flow Diagram, Data Flow Diagrams and System Flow Diagrams then adding changes in Functional Specifications and Use Case	Aliño
06/16/2025	2.8	Changes in Diagrams and some of the contents	Aliño
06/18/2025	2.9	Updated Integration Requirements	Barroso

APPROVALS

Approval Date	Approved Version	Approver Role	Approver
4/17/2025	1.0	Project Manager	Barroso, Lauriene Jade N.
4/18/2025	1.1	Project Manager	Barroso, Lauriene Jade N.
4/19/2025	1.2	Project Manager	Barroso, Lauriene Jade N.
4/20/2025	1.3	Business Analyst	Aliño, Mark B.

4/23/2025	1.4	Business Analyst	Aliño, Mark B.
4/25/2025	1.5	Project Manager	Barroso, Lauriene Jade N.
4/26/2025	1.6	Project Manager	Barroso, Lauriene Jade N.
4/27/2025	1.7	Business Analyst	Aliño, Mark B.
4/29/2025	1.8	Project Manager	Barroso, Lauriene Jade N.
05/11/2025	1.9	Business Analyst	Aliño, Mark B.
05/12/2025	2.0	Project Manager	Barroso, Lauriene Jade N.
05/23/2025	2.1	Project Manager	Barroso, Lauriene Jade N.
05/26/2025	2.2	Project Manager	Barroso, Lauriene Jade N.
05/27/2025	2.3	Project Manager	Barroso, Lauriene Jade N.
05/28/2025	2.4	Project Manager	Barroso, Lauriene Jade N.
06/04/2025	2.5	Project Manager	Barroso, Lauriene Jade N.
06/06/2025	2.6	Business Analyst	Aliño, Mark B.
06/12/2025	2.7	Project Manager	Barroso, Lauriene Jade N.
06/16/2025	2.8	Project Manager	Barroso, Lauriene Jade N.
06/18/2025	2.9	Project Manager	Barroso, Lauriene Jade N.

Table of Contents

1. Introduction

The Submission Tracker & Version Control module is one of the key components of the E-skolarian: PUPSRC Organization Document Management System, developed to address persistent issues in student organization document handling and communication at PUP Santa Rosa Campus. Based on a needs assessment conducted with Engr. Emy Lou Alinsod, Dr. Jonell John Espalto, and student organizations, it became evident that the current system lacks the tools and structure needed for efficient, transparent, and student-centered document management.

While faculty and administrative offices utilize a Document Management System, student organizations still follow a mostly manual process. Documents related to events, activities, and scholarships are physically submitted and often left without clear tracking or reference. According to Dr. Jonell John Espalto, the absence of a centralized repository and formal tracking mechanism hinders organization presidents, especially when managing approved event dates or awaiting document statuses. Additionally, announcements are typically made through informal platforms such as Messenger, further complicating coordination.

President Leo S. Abonita also emphasized the strain placed on student leaders who must travel to the campus just to follow up on pending documents—a challenge for those living far from Santa Rosa. Communication through informal channels is seen as unprofessional and unreliable, and student organizations currently receive no notification or reference code to monitor their submissions. As a result, the approval process can take 2–3 weeks or longer, often depending on persistent in-person follow-ups.

To address these issues, the Submission Tracker & Version Control module offers a centralized platform for submitting and managing student organization documents. Key features include real-time status updates (e.g., Pending, Under Review, Return, Approved), version tracking for returned documents, and a visual timeline for better visibility of progress. A built-in messaging panel also enables direct, professional communication between student organizations and reviewers. Once a document is Approved, it is automatically moved to the History section for record-keeping.

This Functional Specification Document defines the features, workflows, and interface elements of the module to guide its development and ensure it aligns with the broader goals of the E-skolarian system—promoting transparency, efficiency, and accessibility in student organization-led document processes.

1.1 Purpose of the document

The Functional Specification Document provides detailed information on how the system solution will function and the requested behavior. This document is created based on the high-level requirements identified in the Business Requirements Document and provides traceability on the functional specifications back to the business requirements. Included in this document will be the detailed functional requirements including use cases, system inputs and outputs, process flows, diagrams, and mockups.

1.2 Project Scope

The project is primarily intended to enhance the workflow for tracking and revising academic or organizational documents submitted by students. It is comprised of two modules: Submission Tracker and Version Control. Each module has its own purpose, but they can also work in partnership to improve the workflow process.

The Submission Tracker module is designed to give students a birds-eye view of all the documents, they have submitted and user-friendly access to track them, in real time, whether they are pending, under review, approved or returned. The Submission Tracker also includes features including search, filters, pagination, and sorting features to help users navigate documents easily across a large number of submissions.

When a submission is placed in review, the process moves to the Version Control module. This module allows students to upload more than one version of the same document. Each student submission is dated and collected sequentially to show the history of submissions. This module also provides automatic version numbering as well as a change log of edits and reviewer comments, ensuring transparency and accountability. Both modules use PUP's Webmail for secure login authentication, have role-based access, and are integrated. Overall, the project aims to address the tracking and revision process, increase communication between students and reviewers, and decrease the chance of miscommunication or lost documents.

1.3 Scope of the document

This Functional Specification Document focuses specifically on the Submission Tracker & Version Control module of the E-skolarian: PUPSRC Organization Document Management System. While the overall system aims to streamline and digitize the handling of documents related to student organizations, this module addresses the tracking, review, and communication process for submitted documents.
Some of the functions of the Submission Tracker & Version Control module are:

- Provide real-time status updates (e.g., Pending, Under Review, Approved, Returned)
- Display a visual timeline of the document's review progress
- Allow feedback and comments from reviewers
- Enable return of documents with version history
- Facilitate direct communication through a built-in messaging panel

1.4 Related documents

Component	Name (with link to the document)	Description
Project Charter - Submission Tracker and Version Control	Group-2-Project-Charter.pdf	Outlines the project's goals, team roles, and scope. Describes the business need for a student-centered document

Module		management system to improve submission tracking and transparency.
UI/UX Wireframes	UI/UX wireframes	Full visual design of the E-skolarian: PUPSRC Organization Document Management System. The Submission Tracker & Version Control module is one of the included components, featuring layouts for document tables, status tracking, confirmation messages, visual timelines, and the built-in messaging panel.
Concept Paper – Submission Tracker and Version Control Module	Group2_Concept Paper(1.pdf	A focused proposal discussing the specific problems, objectives, and initial solution design for the Submission Tracker & Version Control module. It highlights the module's importance in improving tracking, feedback, and transparency in student document submissions.
Needs Assessment	Needs Assessment - folder	This folder contains the needs assessment documents gathered from key stakeholders of the E-skolarian: PUPSRC Organization Document Management System. It includes inputs from Dr. Jonell John O. Espalto, Pres. Leo S. Abonita, and Engr. Emy Lou Alinsod. The assessments provided essential guidance in identifying core features, prioritizing functionality, and ensuring that the proposed system aligns with the actual workflow and documentation requirements of student organizations at PUP Santa Rosa.
Project Roadmap	Group 2_Roadmap (4).pdf	This document outlines the week-by-week development timeline and sprint goals for the Submission Tracker & Version Control module, aligned with the E-skolarian system milestones. It shows the progression from planning and design to development, testing, and deployment.
Tech Stack	GROUP 2 TECH STACK.pdf	This file details the technologies for the development of the Submission Tracker & Version Control module.
UI/UX Wireframe – Submission Tracker and Version Control Module	Group_2_Submission-Tracker-and-Version-Control_Wireframe.pdf	This wireframe presents the layout, structure, and visual flow of the Submission Tracker & Version Control module.
Minutes of the Meeting	Minutes of the Meeting	This folder contains the Minutes of the Meeting (MOTM) conducted at the beginning of each sprint.

1.5 Terms/Acronyms and Definitions

Term/Acronym	Definition	Description
FSD	Functional Specification Document	A formal document that outlines the features, behavior, and interface of a module within the system.
ODMS	Organization Document Management System	The overall system used to manage, track, and process student organization documents at PUP Santa Rosa Campus.
Admin	Campus Administrator	Refers to the assigned reviewer or campus official who validates, approves, or rejects submissions.
Super Admin	Super Administrator	A user role with the highest level of access in the system, responsible for creating accounts, assigning roles, and overseeing system-wide settings and access control.
UI/UX	User Interface/ User Experience	UI refers to the process of designing and creating the interactive and visual user interface while UX refers to the process of creating digital products or services that primarily caters user's needs, preferences, and behaviors. (Hamidli, 2023)
PUPSRC	Polytechnic University of the Philippines Santa Rosa Campus	The Polytechnic University of the Philippines Santa Rosa Campus (PUP-SRC) is a public higher education institution established in 2003 in Santa Rosa, Laguna, offering programs in fields such as Accountancy, Education, Engineering, Psychology, Business, and Information Technology. (Polytechnic University of the Philippines – Santa Rosa Campus, 2023)

1.6 Risks and Assumptions

There are many factors that can affect the functionality and design of the module during its implementation. These factors may include limitations of the system, resource availability and problems that might be encountered. We can proactively address potential challenges and expectations for the development of the module by identifying the risks and assumptions.

Here are risks that the project might encounter in the future:

a. Real-Time Status Inconsistencies - Document status changes such as Pending, Under Review, Approved and Returned might not show updates instantly due to system processing delays or network problems thus creating confusion among users who may miss out on necessary actions.
b. Document Version Management Conflicts - Students can submit their documents repeatedly during the assignment process. A clear version tracking mechanism becomes essential to prevent errors in both evaluation and revision because it ensures proper association of feedback and status with each version.
c. Feedback Linking Errors - Users will not achieve optimal results from the module because they may update documents based on outdated or irrelevant feedback when version links remain unclear.
d. Unrestricted File Uploads - The system faces vulnerability issues from large document uploads because it does not have any file size restrictions in place right now. The system could experience slowing down in loading times and decreased system responsiveness along with reduced storage accessibility due to excessive file sizes.
e. User Misunderstanding or Misuse - The confused understanding of the submission process and version handling creates errors where users submit incorrectly or fail to get their required feedback.

The assumptions outline the foundational conditions and expectations considered during the design and development of the module. This the list of assumptions of the module:

a. Clear User Roles - The system contains established roles including Student, Student Organization and Campus Administration which control document submission procedures and feedback approval processes.
b. Consistent User Engagement - Users must actively participate in the module by frequently checking status updates along with providing feedback and managing re-submissions properly.
c. Defined Submission Workflow - The functional design relies on a defined submission and review workflow which each user must use starting from upload through review and feedback and ending with final approval.

2. Module/ Solution Overview

The Submission Tracker & Version Control module of E-skolarian: PUPSRC Organization Document Management System is designed to make it easier for student organizations to stay on top of their document submissions. Its purpose is to help users keep track of where each document is in the approval process and ensure that every update or revision is properly recorded. With this module, students and advisers can easily follow submission timelines, view feedback, and access the full version history of each file—all in one place.

By providing a clear and organized way to manage documents, the module supports smoother coordination, reduces confusion over updates, and helps everyone stay aligned with deadlines. It's built to encourage accountability, improve transparency, and save time by cutting down on back-and-forth about which version is the latest or whether something has been submitted at all.

2.1 Context Diagram/ Interface Diagram/ Data Flow Diagram, Application Screen Flow, Sitemap, Process Flow

Figure 1. Context Diagram - Submission Tracker and Version Control Module

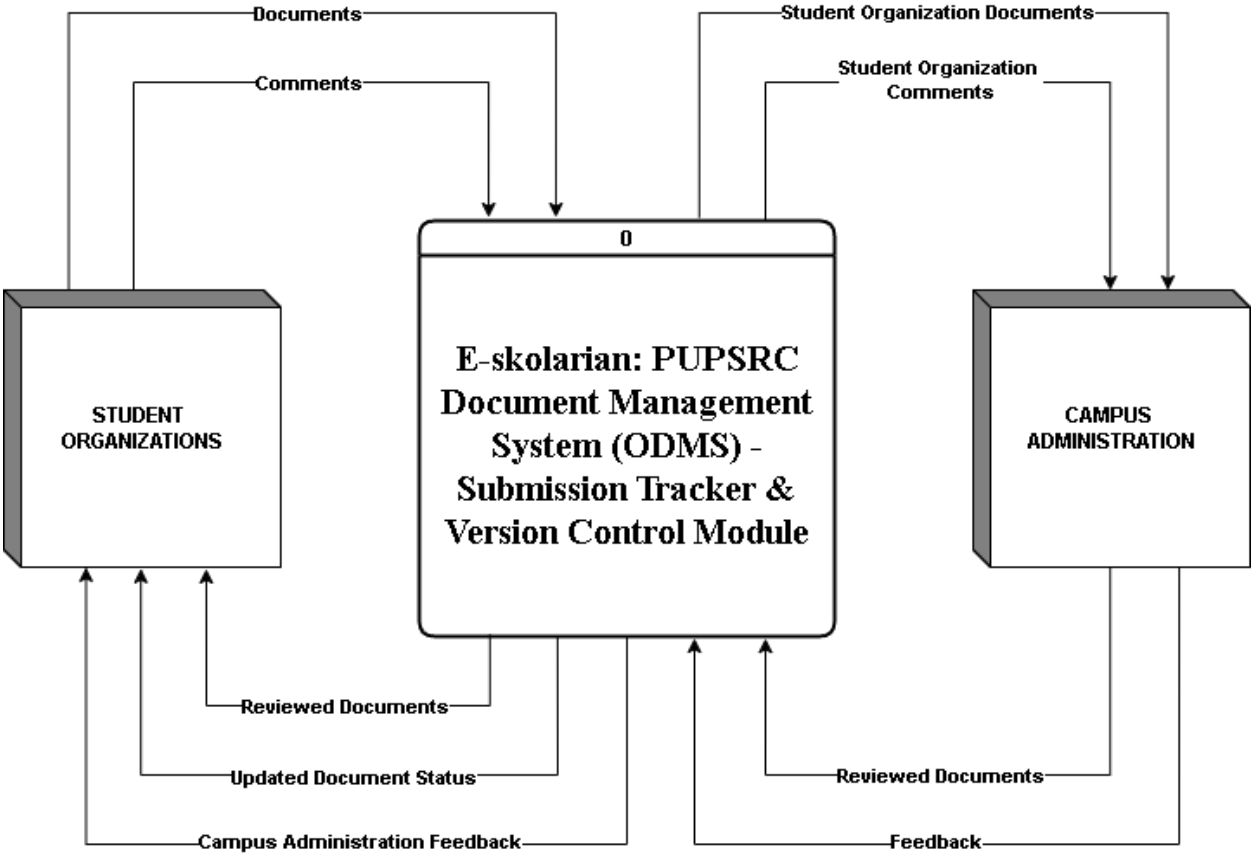


Figure 2. Data Flow Diagram – Submission Tracker and Version Control Module Level 0

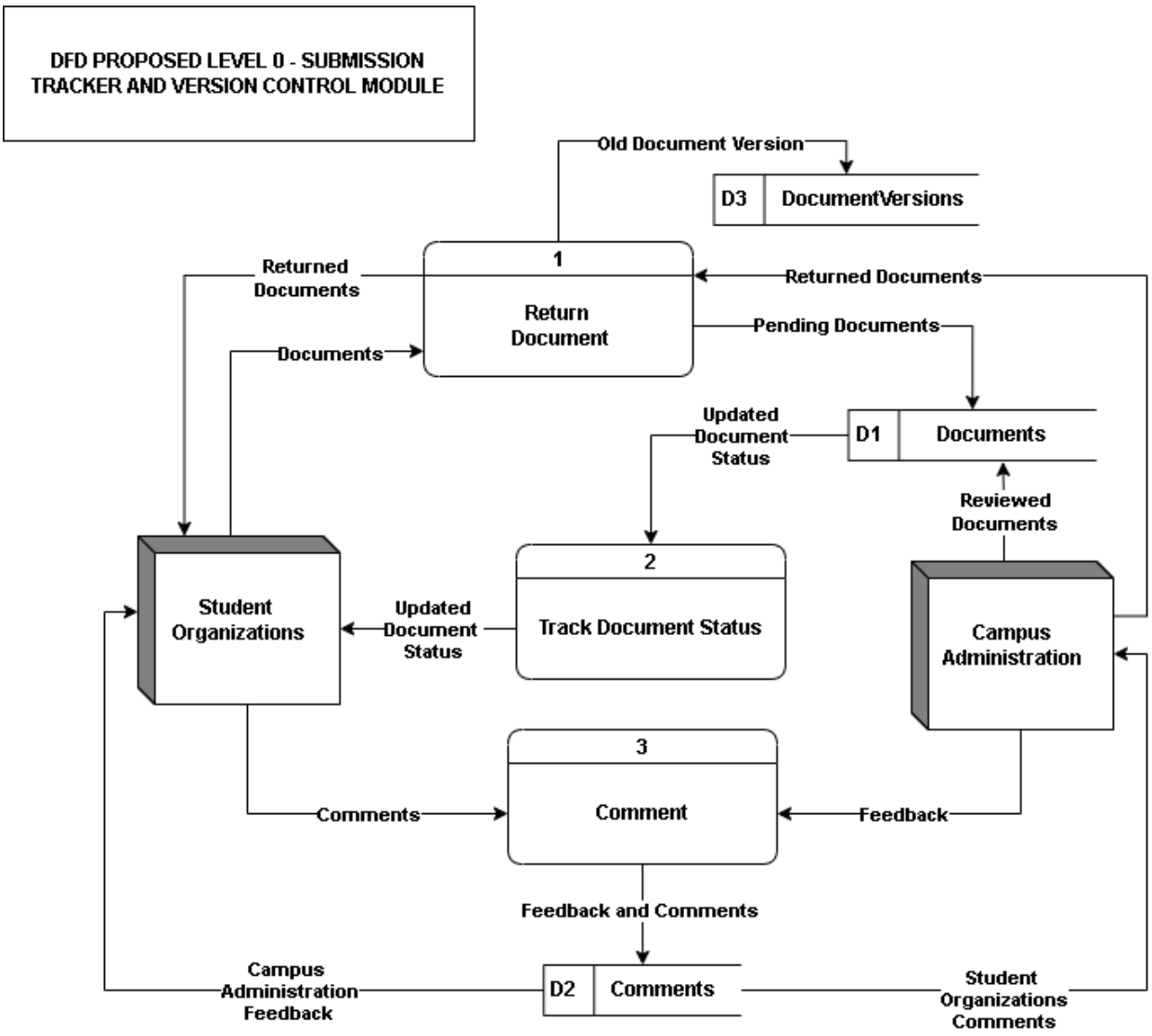


Figure 3. Data Flow Diagram – Submission Tracker & Version Control Module Level 1

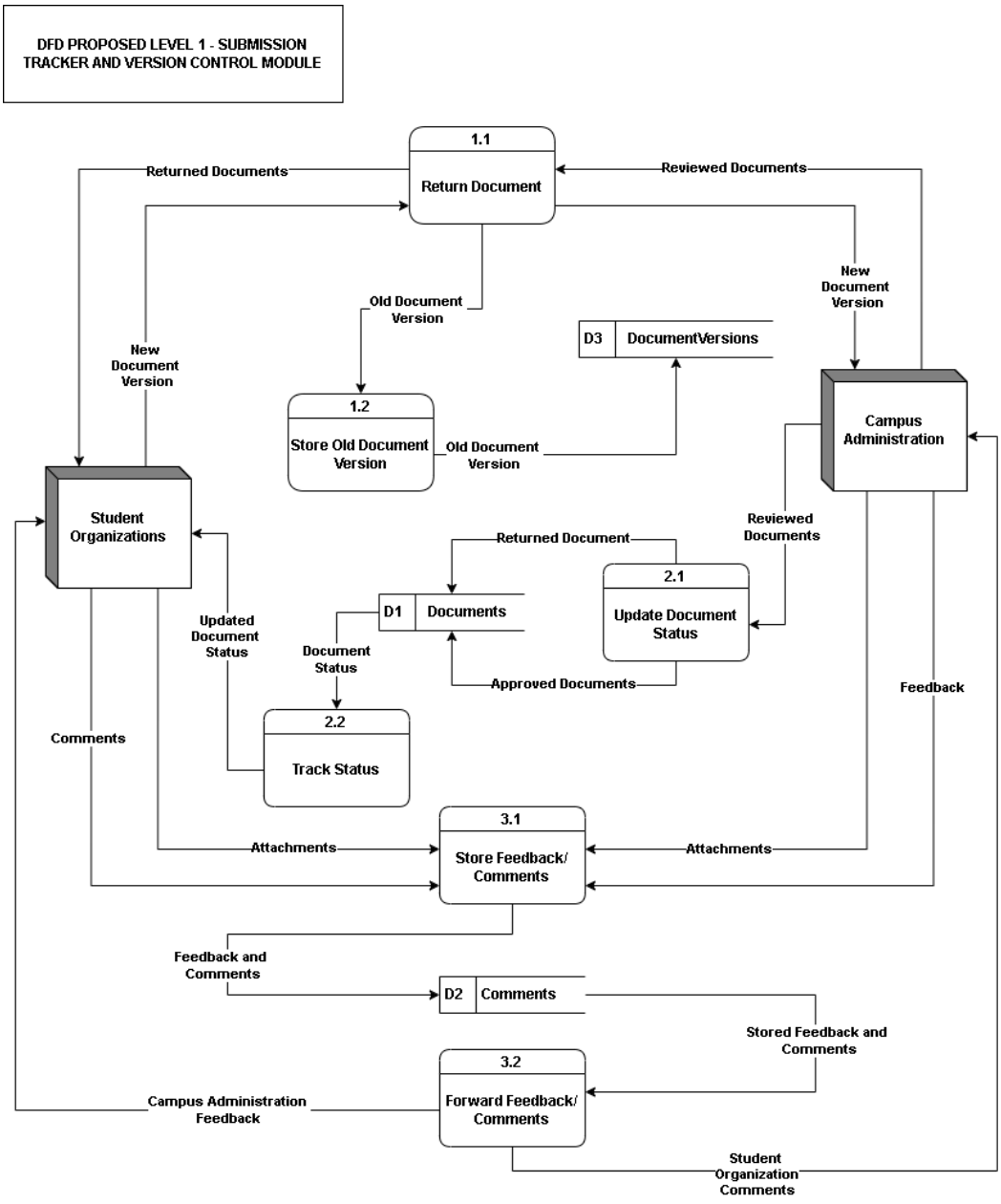


Figure 4. Business Process Flow – Submission Tracker and Version Control Module

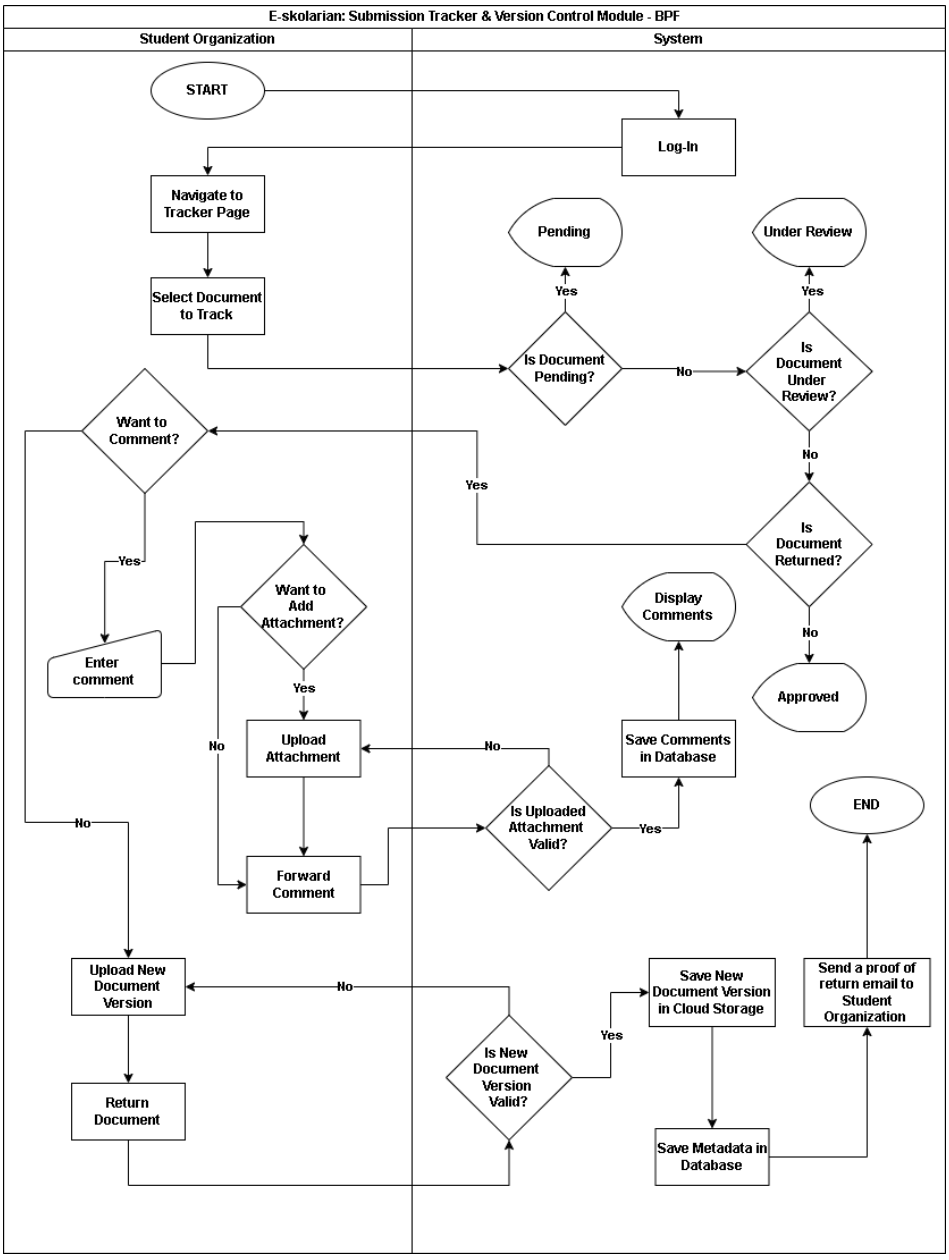


Figure 5. System Flow Diagram – Track Document Status

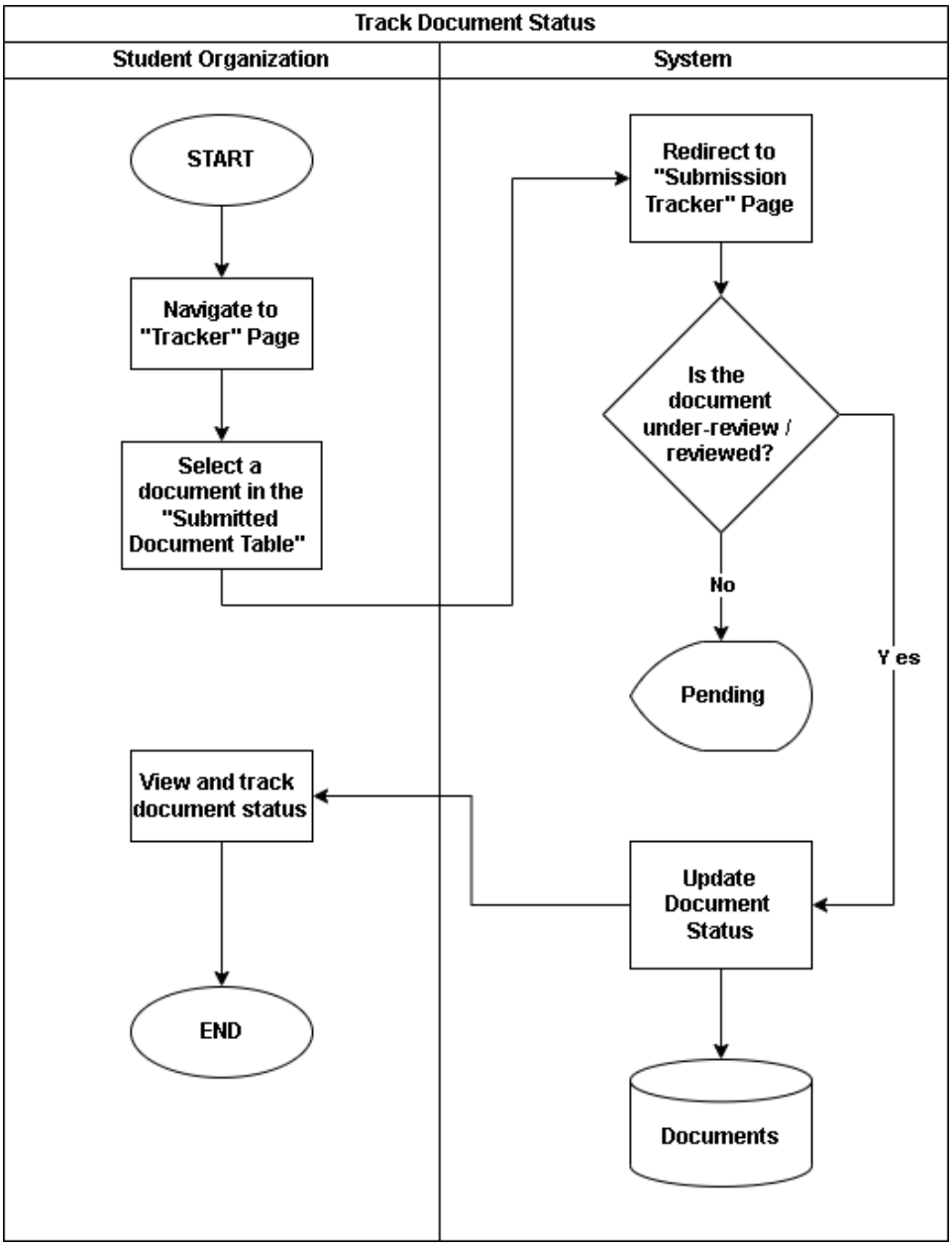


Figure 6. System Flow Diagram – Feedback and Comments

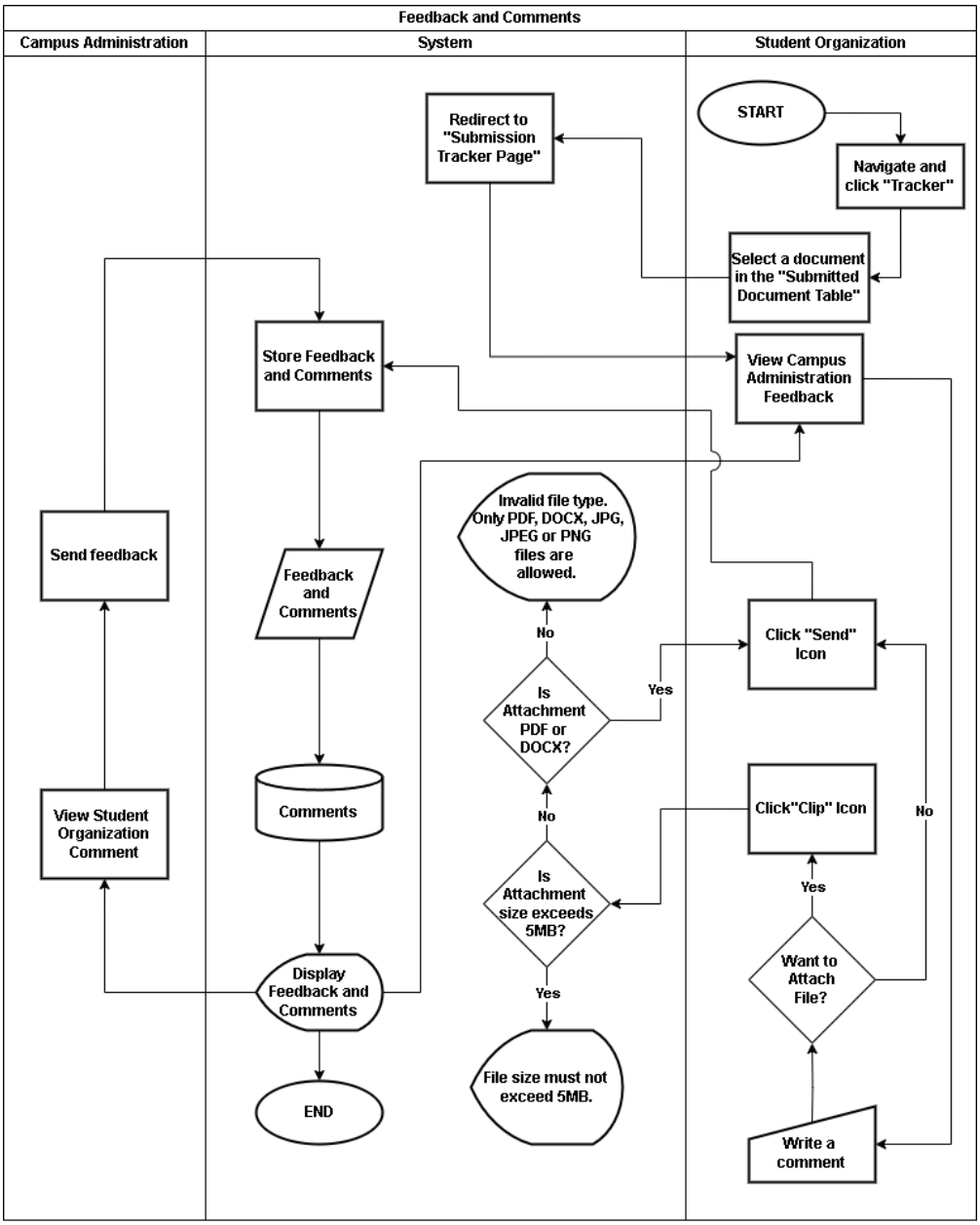


Figure 7. System Flow Diagram – Return Document

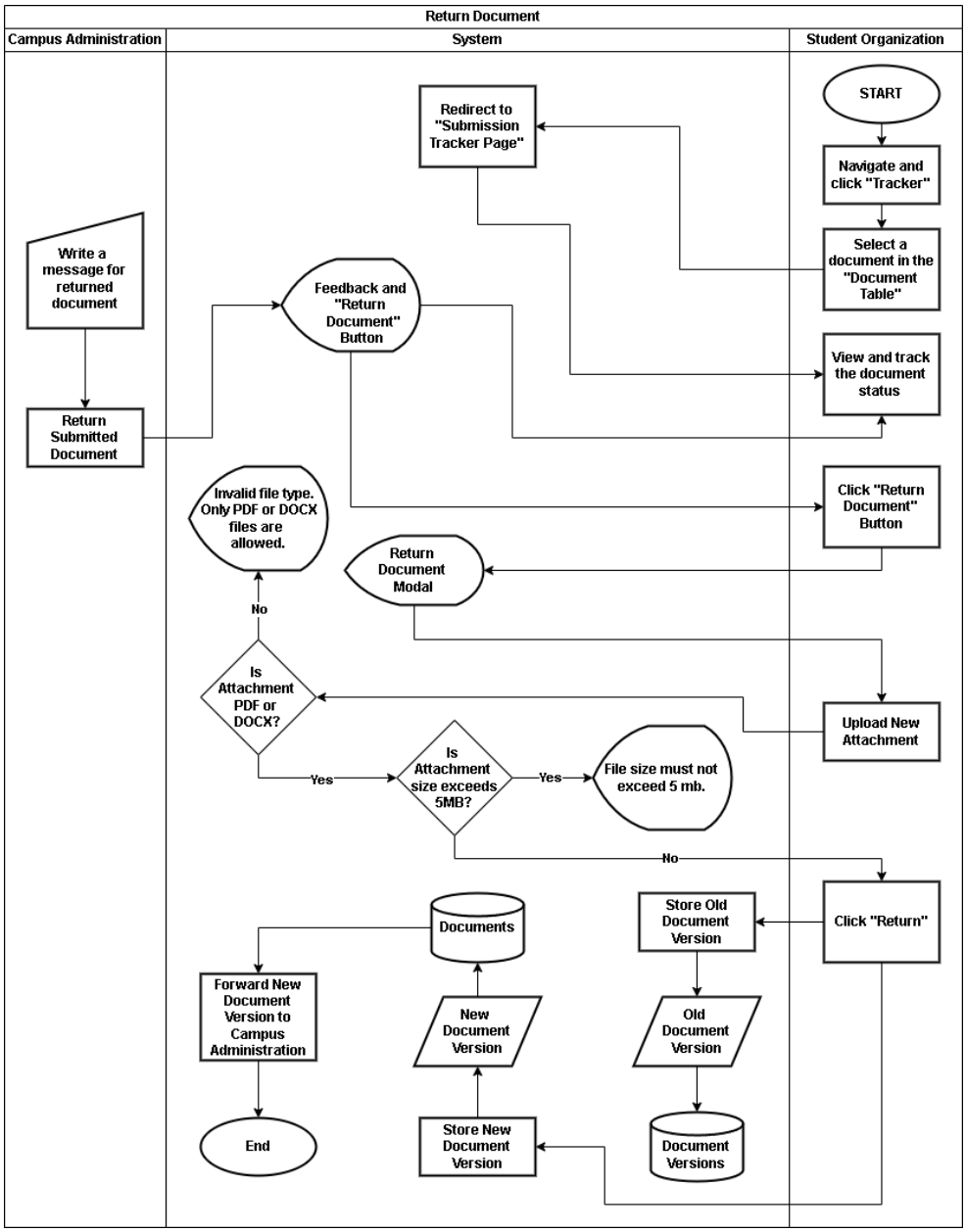


Figure 8. Database Schema

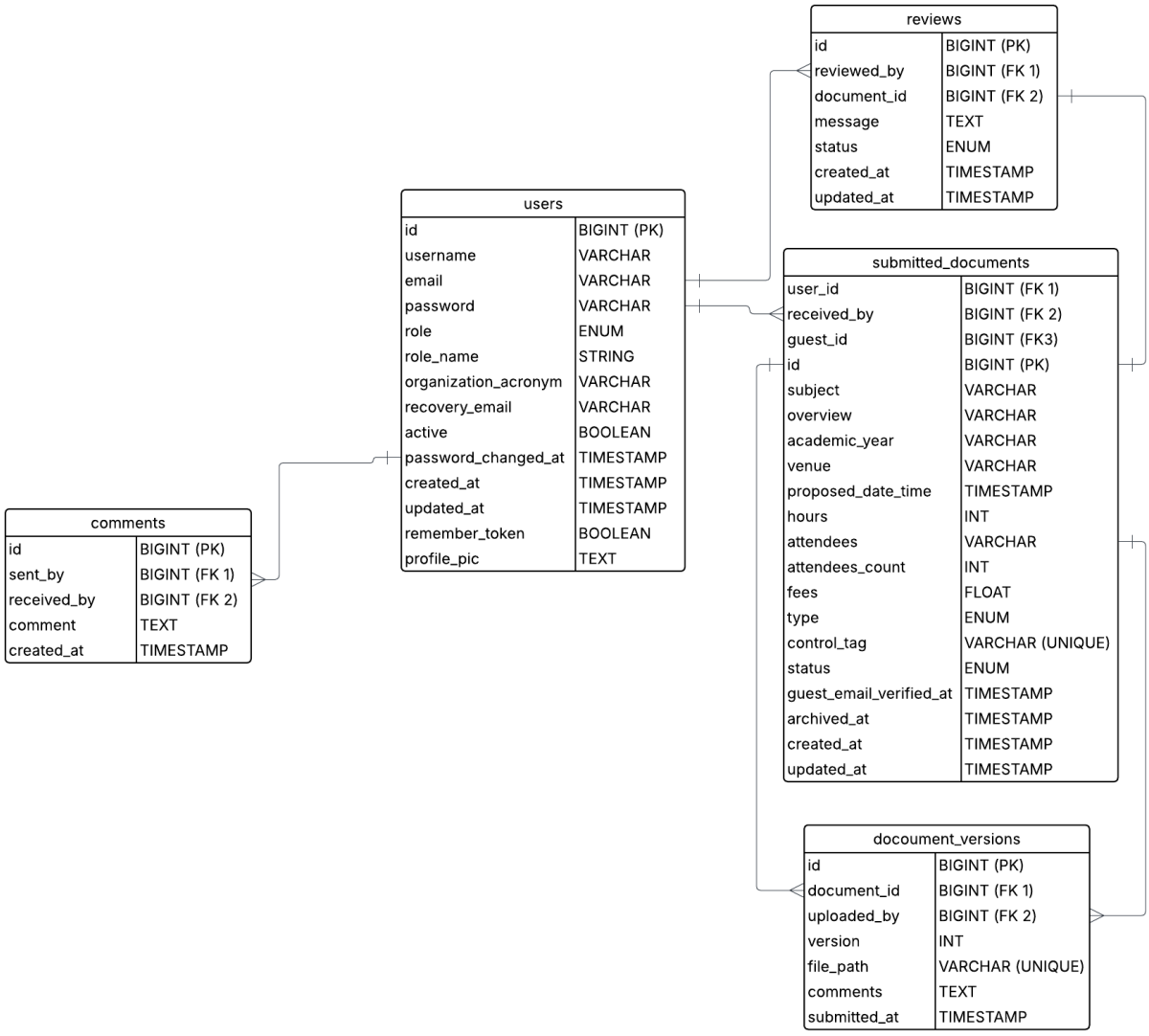
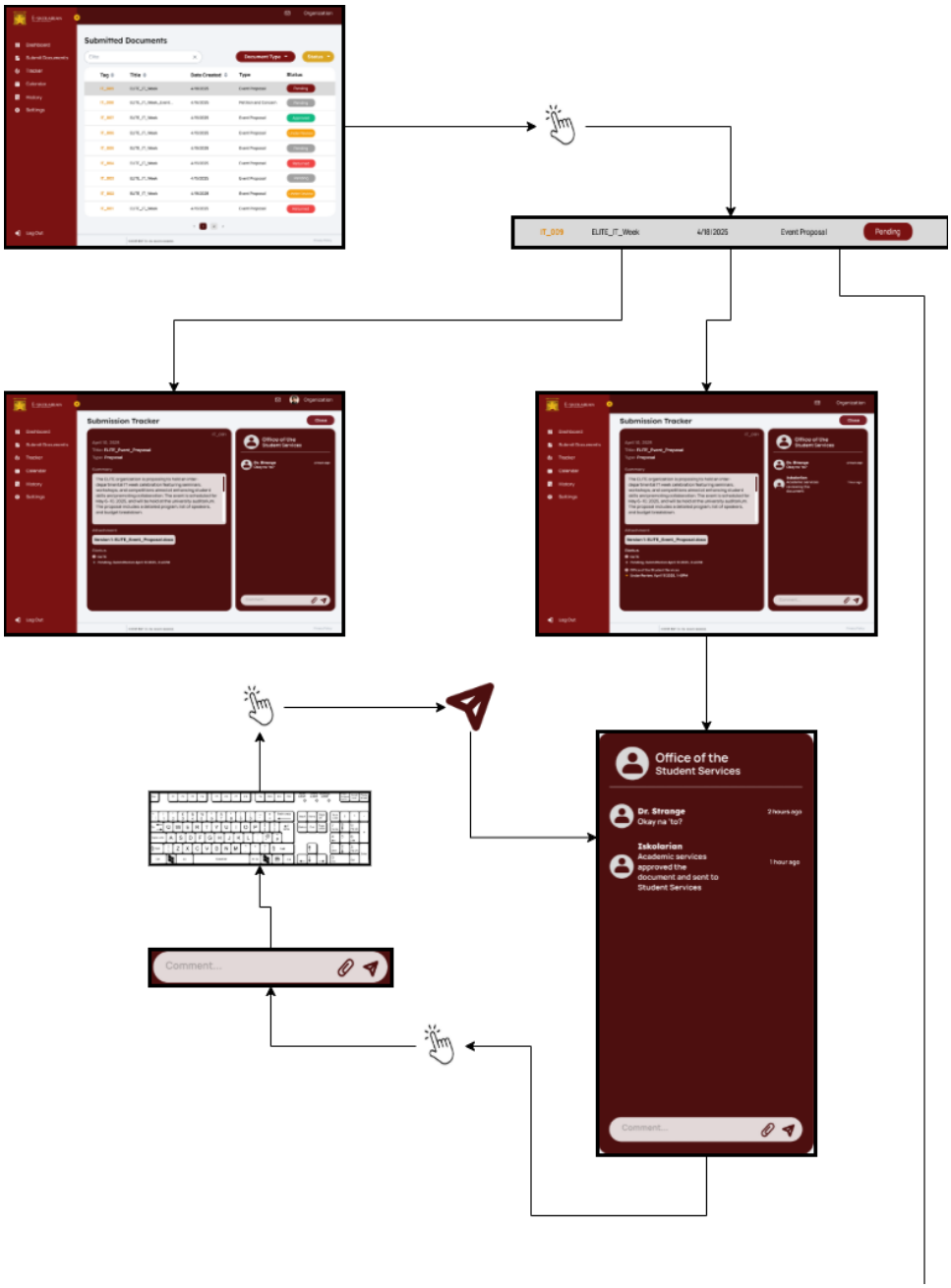


Figure 9. Application Screen Flow


Submitted Documents
IT_009
ELITE_IT_Week
4/18/2025
Event Proposal
Pending
Submission Tracker
Submission Tracker
Office of the Student Services
Dr. Strange
Okay na 'to?
2 hours ago
Iskolarian
Academic services approved the document and sent to Student Services
1 hour ago
Comment...
Comment...

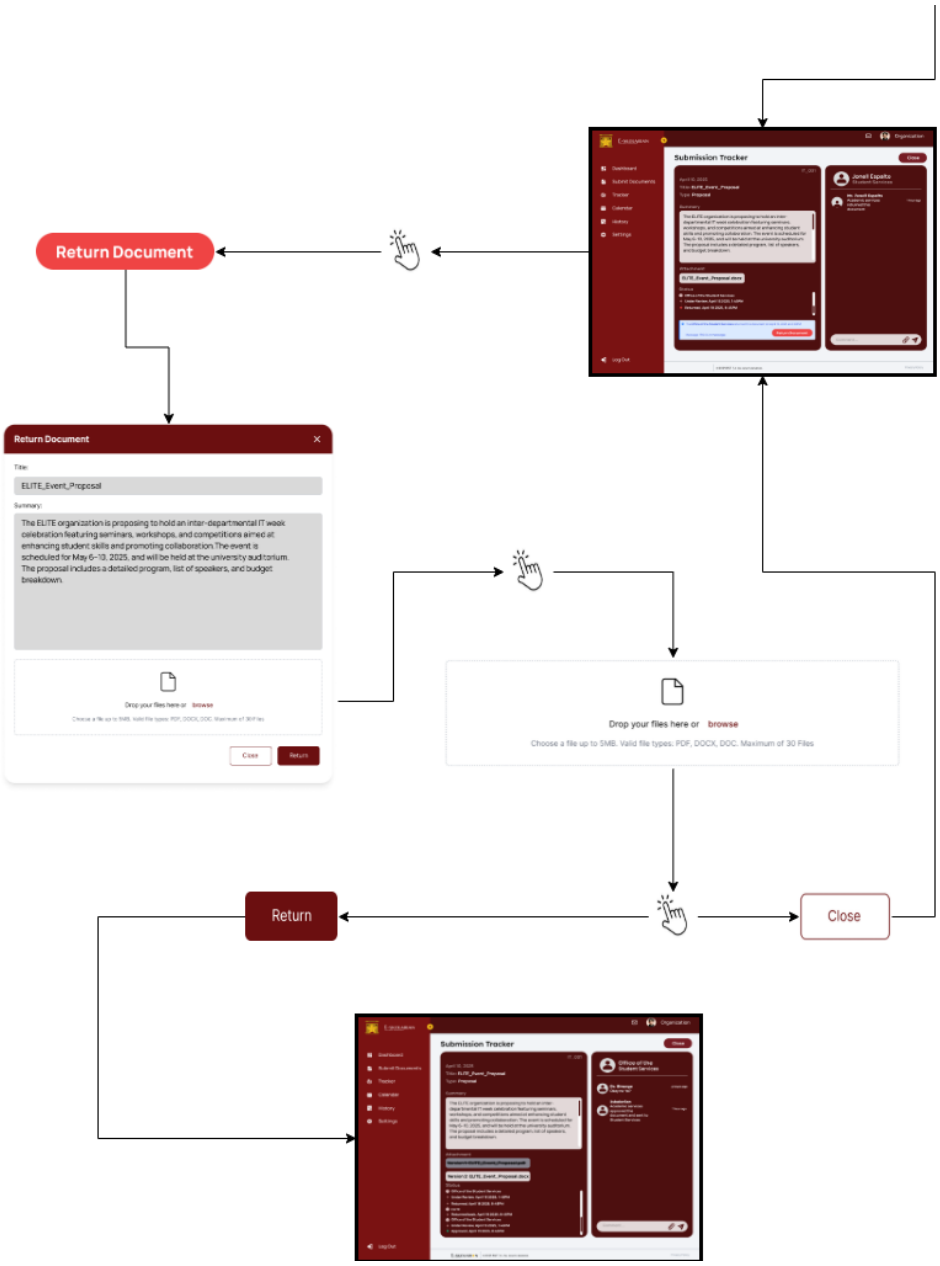
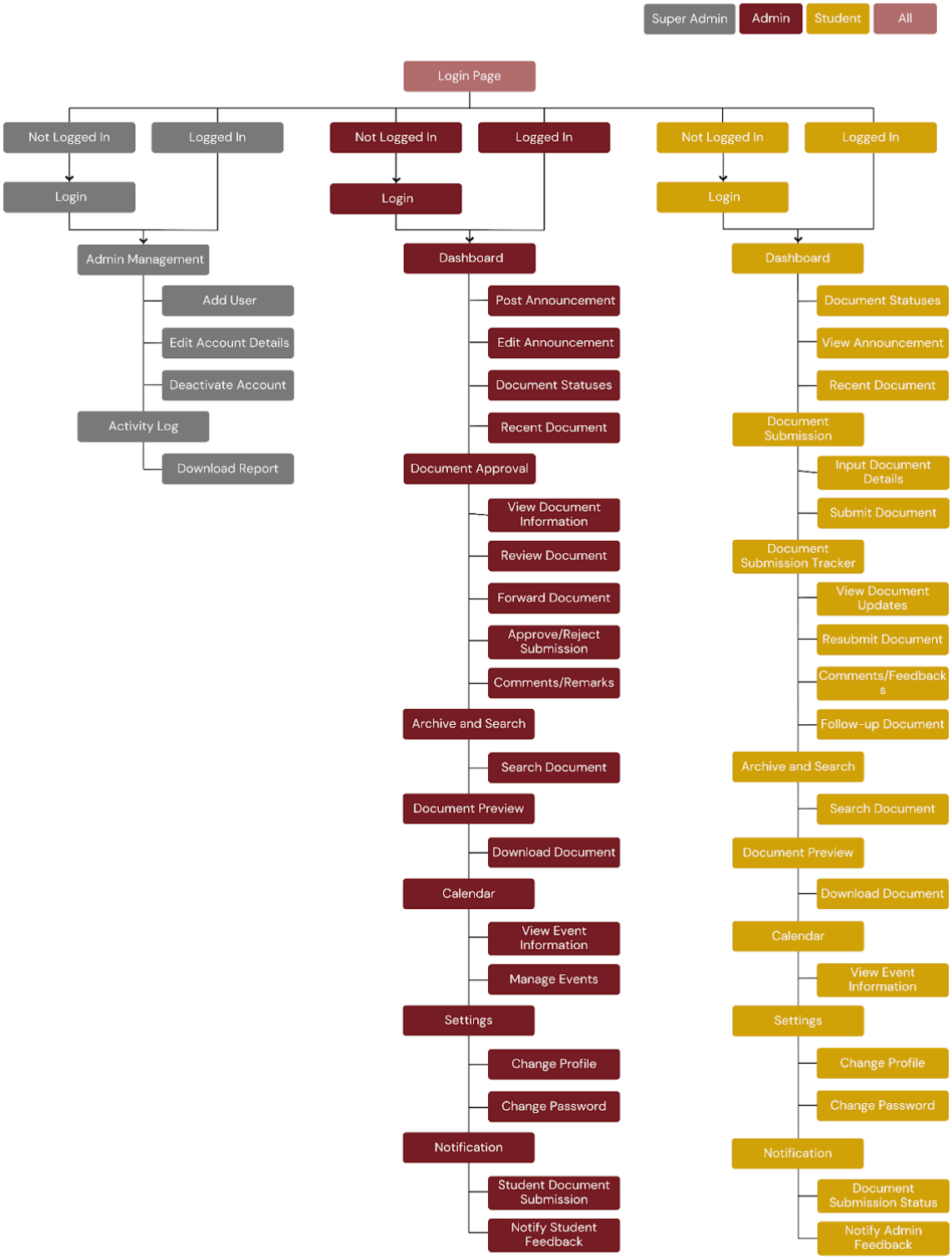


Figure 10. Site Map


Super Admin
Admin
Student
All
Login Page
Not Logged In
Logged In
Not Logged In
Logged In
Not Logged In
Logged In
Login
Login
Login
Admin Management
Dashboard
Dashboard
Add User
Edit Account Details
Deactivate Account
Activity Log
Download Report
Post Announcement
Edit Announcement
Document Statuses
Recent Document
Document Approval
View Document Information
Review Document
Forward Document
Approve/Reject Submission
Comments/Remarks
Archive and Search
Search Document
Document Preview
Download Document
Calendar
View Event Information
Manage Events
Settings
Change Profile
Change Password
Notification
Student Document Submission
Notify Student Feedback
Document Statuses
View Announcement
Recent Document
Document Submission
Input Document Details
Submit Document
Document Submission Tracker
View Document Updates
Resubmit Document
Comments/Feedbacks
Follow-up Document
Archive and Search
Search Document
Document Preview
Download Document
Calendar
View Event Information
Settings
Change Profile
Change Password
Notification
Document Submission Status
Notify Admin Feedback

2.2 System Actors

2.2.1 User Roles and Responsibilities / Authority Requirements

User/Role	Example	Frequency of Use	Security/Access, Features Used	Additional Notes
Student Organization	Member of PUP Santa Rosa student organization submitting documents	High	• Access to Submit Documents • View Submission Tracker • Return Documents • View and reply to feedback/comments	Cannot edit document once submitted but can return if requested.
Campus Administrator	Organization adviser, PUP staff member assigned as document reviewer	Medium	• Review documents • Approve or return • Add feedback/comments • View submission timeline	Can only approve or return at the admin level then forwards the documents to Director after initial approval.
Director/Final Reviewer	Department Director or Organizational Head	Medium	• Final review of documents • Approve or reject final submissions • Add final feedback/comments (optional) • Final status update triggers system archiving	Director's decision is final whether it is approved or Return.
System (Automated Actions)	Background system process	Continuous	• Update real-time status • Log timestamps	Performs background functions not manually triggered by users.

2.3 Dependencies and Change Impacts

2.3.1 System Dependencies

1. **Database System**
   - **Description:** The entire submission tracking data including document IDs, status updates, timestamps, comments, and version history is stored here.
   - **Purpose:** It allows students to see the current state of submission as documentation processes occur, while allowing structured storage of multiple versions of documents.
   - **Impact:** If the database system becomes unavailable, students will not be able to view the progress of their submission or view previously uploaded versions of documents.
   - **Mitigation:** Automatic or daily backups, high-availability replication, and monitoring so that if the system encounters issues, it can be detected and solved quickly.

2. **File Storage System**
   - **Description:** The system used to store the versioned files for every upload made for each submission, as well as upload metadata regarding when they were uploaded and the file type used for uploads.
   - **Purpose:** It provides version management by preserving each document update, including time stamps and filetype in a structured and easily retrievable form.
   - **Impact:** A failure in the file storage system may cause the loss of versions or make them inaccessible. Without access to revision history of the version with access storage, tracking or revision becomes impossible.
   - **Mitigation:** Store files using secured and redundant sources, such as hybrid storage, periodically backed up, and with checked version integrity.

3. **Email Notification Service**
   - **Description:** Sends automated notifications about document status changes (e.g. approved or returned) and version uploads to students.
   - **Purpose:** To keep students up to date with the progress of their submissions and any related actions.
   - **Impact:** Should the service fail to deliver the email, students may miss critical updates which may delay the revision process or render them confused around whether a submission was received or approved or returned.
   - **Mitigation:** Entail retry mechanisms, email delivery tracking and in-app notification alternatives to safeguard that critical messages are sen to recipients.

4. **Submission Status and Versioning Logic Engine:**
   - **Description:** Executes the backend code logic for setting a document's status and multiple version entries per submission.
   - **Purpose:** Ensures version uploads happen in logical sequence and the appropriate status updates are recorded and displayed to students.
   - **Impact:** If the logic engine fails, students may see inappropriate status labels or version mismatches which may result in confusion or submission error.
   - **Mitigation:** Create unit tests continuously, create try/catch logic for revision conflicts, and log every action within the engine for auditing, and rollback if necessary.

2.3.2 Change Impacts

The implementation of the Submission Tracker & Version Control module will impact the following existing systems, processes, and stakeholders at PUP Santa Rosa Campus:

1. **Current Student Organization Document Processes**
   - **Manual Submission & Tracking**: The existing paper-based submission process will be replaced by a digital workflow, reducing physical document handling and in-person follow-ups.
   - **Informal Communication Channels**: Reliance on platforms like Messenger for announcements and updates will be phased out in favor of the module's built-in messaging and notification system.

2. **Faculty/Campus Administration Document Management System (DMS)**
   - **Integration Requirements**: While faculty/campus administration offices already use a DMS, this module will introduce a parallel (but interconnected) system for student organizations, potentially requiring:
     - Data synchronization for cross-departmental approvals (e.g., student activity permits).
     - User role adjustments to accommodate student organization and reviewers.

3. **IT Infrastructure**
   - **Server/Storage Expansion**: Additional database resources may be needed to host document submissions, version histories, and user records.
   - **Security Policies**: New authentication and access control measures must align with PUP's data privacy policies (e.g., role-based permissions for student organizations and campus administration).

4. **Stakeholder Workflows**
   - **Student Leaders**: Will transition from physical follow-ups to digital tracking, requiring training on the new system.
   - **Reviewers (Campus Administration)**: Must adapt to centralized review queues, status updates, and in-system messaging instead of email/Messenger.
   - **IT Support Team**: Will assume maintenance responsibilities (e.g., troubleshooting, backups) for the new module.

5. **Campus Network**
   - **Increased Bandwidth Demand**: Higher reliance on document uploads/downloads may necessitate network capacity reviews.

3. Functional Specifications

This section describes the functional characteristics for the Submission Tracker & Version Control Module within E-skolarian ODMS. The system's fundamental requirements together with use cases get transformed into practical features which enable user requirements fulfillment. This section organizes system features by module and user roles, using separate system interfaces to clarify how users and campus administrators interact with the system.

3.1. E-skolarian: PUPSRC Organization Document Management System – Submission Tracker & Version Control Module

3.1.1 Purpose/ Description

This section outlines the specifications for the Submission Tracker & Version Control Module of the E-skolarian: PUPSRC Organization Document Management System. The module is built to make document submission process easier for student organizations at the Polytechnic University of the Philippines – Santa Rosa Campus (PUP-SRC). Its primary purpose is to provide transparency, accountability, and efficiency in tracking document statuses and managing multiple submission versions.

Key features include:

- Real-time status updates that reflect document progress through various stages such as Pending, Under Review, Approved, or Returned.
- Return of documents capabilities with complete version history tracking, allowing users to update documents without losing previous data.
- A visual timeline interface that helps both students and reviewers easily understand the document's journey and status immediately.

Table of Functionality

Functionality	Description	Linked Use Case(s)
Real-Time Status Update	The status system displays changes in incoming requests automatically through Pending, Under Review or Approved or Returned categorizations.	UC-1
Feedback and Comments	An interface allows campus administration to provide feedback which student organizations can reply to.	UC-2

Return of Documents	Through this system students can post updated documents which maintain stored versions with full tracking accessible in a version history log.	UC-3

3.1.2 Use case

UC-1	**Document Tracker**
Primary Actor(s)	Student Organization, System
Stakeholders and Interest	Student Organization wants to know the current status of their submitted document.
Trigger	The user clicks on the row corresponding to the document they want to track from Submitted Document Page in “Tracker” menu in the sidebar.
Pre-conditions	The document must already be submitted and under review process.
Post-conditions	Student Organization sees the updated status in real-time.
Main Success Scenario	1. The user navigates to the Submitted Documents Page by selecting the "Tracker" option from the sidebar menu. 2. The users select the row with the document that they want to track with a “Pending” status. 3. The user is redirected to the “Submission Tracker” page where the status of the selected document is displayed.
Extensions	3.1. If the status update fails, refresh or notify the user. 3.2. Use visual indicators like color-coded badges or icons.
Priority	High
Special Requirements	Use asynchronous updates to reflect changes instantly without reloading the page.
Open Questions	How should the system prevent status conflicts if multiple campus administrators view or attempt to update the same document at the same time?

UC-2	**Feedback and Comments**

Primary Actor(s)	Campus Administrators, Student Organization
Stakeholders and Interest	Campus Administrators provide feedback to guide document submissions while Students Organization Representative can view and reply to comments for clarification or further discussion.
Trigger	The user navigates and click the “Comment” field in the right section of the “Submission Tracker”.
Pre-conditions	A document has been submitted.
Post-conditions	Feedback and comments are saved and visible in the comment section in “Submission Tracker” page.
Main Success Scenario	1. The user navigates to the Submitted Documents Page by selecting the "Tracker" option from the sidebar menu System updates the status. 2. The users select the row with the document that they want to track. 3. The user is redirected to the “Submission Tracker” page where the status of the selected document is displayed. 4. The user navigates and click the “Comment” field in the right section of the “Submission Tracker”. 5. The user comment or reply to the feedback of the campus administrator.
Extensions	5.1. If feedback/comments fail to save prompt the user with an error message.
Priority	Medium
Special Requirements	• Feedback must be version-specific and persistent across sessions.
Open Questions	How can the system ensure that feedback remains version-specific and persists across multiple sessions for both the Campus Administrator and the Student Organization?

UC-3	**Return of Document**
Primary Actor(s)	Student Organization
Stakeholders and Interest	Users possess the ability to return previously submitted documents which required correction work or additional requirements after returns.
Trigger	The document required rework according to campus administrator or reviewer decisions making the "Return Document" button accessible to

	users.
Pre-conditions	The user needs to obtain a document return request either from a campus administrator or director.
Post-conditions	The return document is uploaded, replacing the previous version, and its status is updated in the tracker.
Main Success Scenario	1. The user navigates to the Submitted Documents Page by selecting the "Tracker" option from the sidebar menu System updates the status. 2. The users select the row with the desired document just before tracking if its status shows “Pending”. 3. The user is redirected to the “Submission Tracker” page where the status of the selected document is displayed. 4. The user views a document returned status with metadata and message from the campus administrator. 5. The user clicks the "Return Document" button. 6. The user opens the return document modal. 7. The user clicks the dropbox button to upload a new file. 8. The user clicks the “Return” button.
Extensions	7.1. If the user uploads an invalid file, the system shows an error and prevents submission until corrected.
Priority	High
Special Requirements	• The system must validate file format restrict access to the original submitter and log the return of document timestamp.
Open Questions	Should the system send confirmation notifications and display a visible version history to users upon return of document?

3.1.3 Mock-up

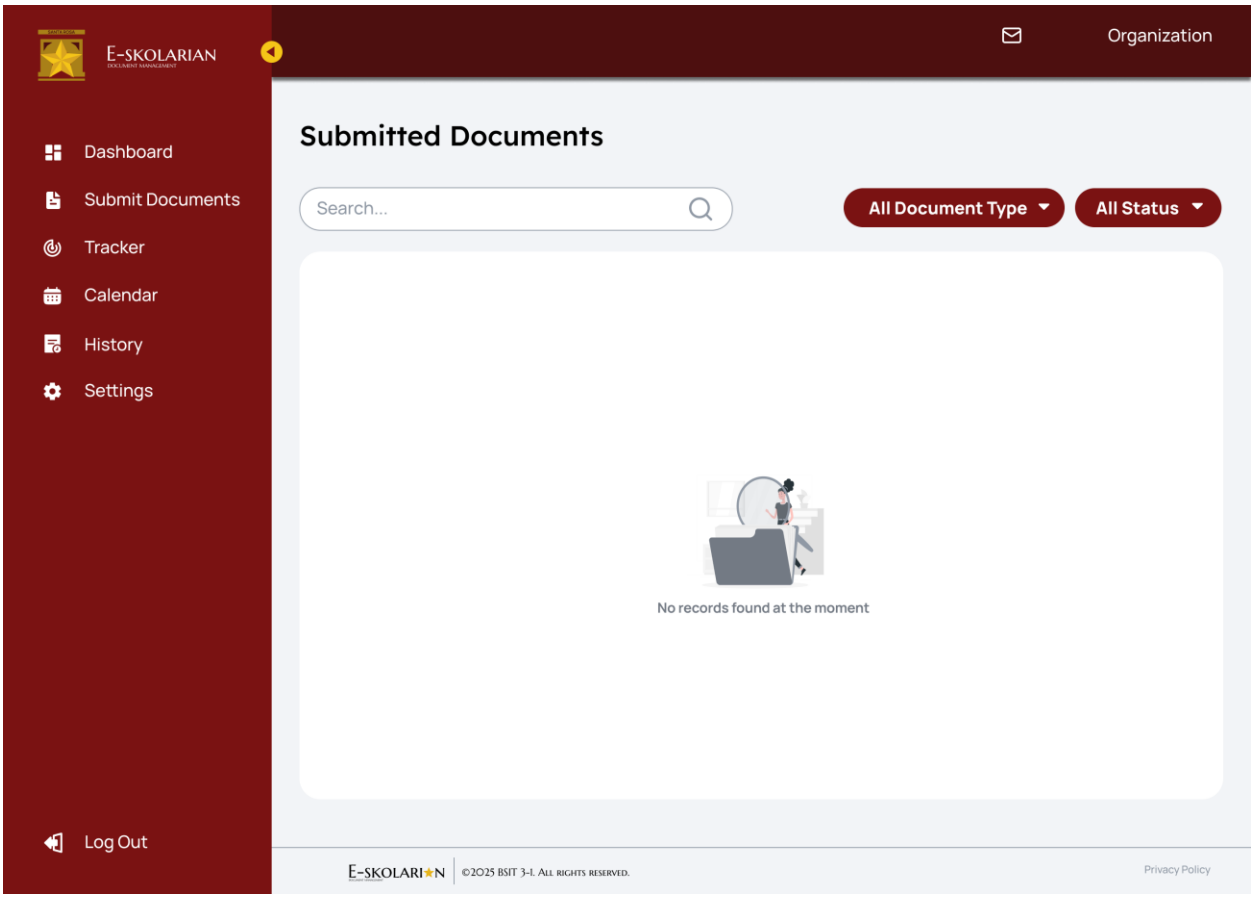


Figure 11. Submitted Document Page (Empty)

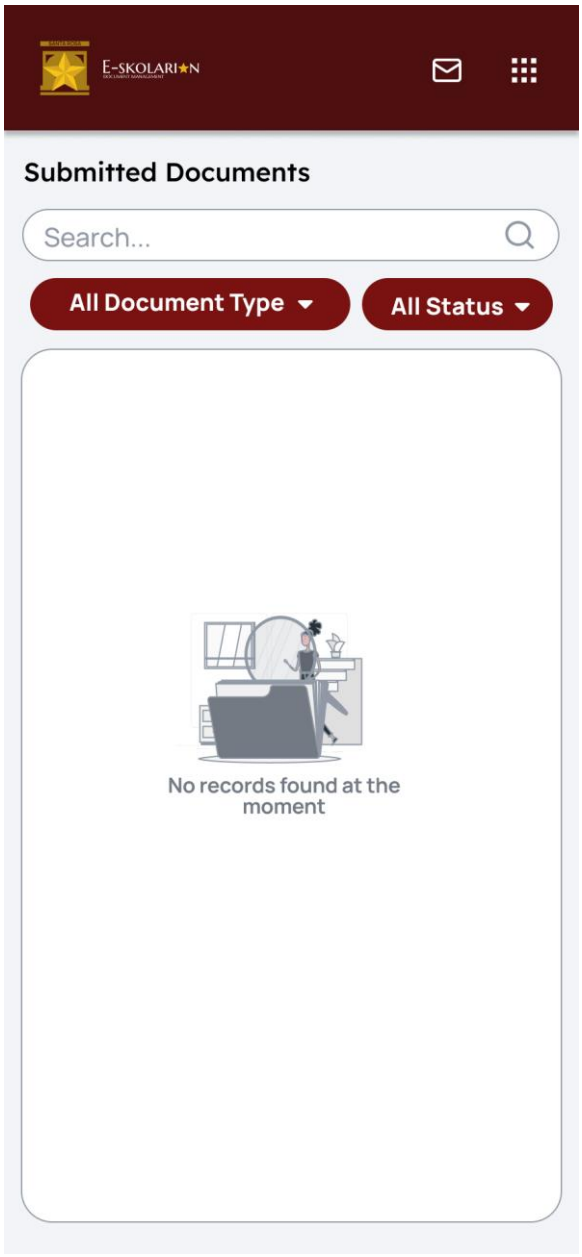


Figure 12. Submitted Document Page Mobile View (Empty)

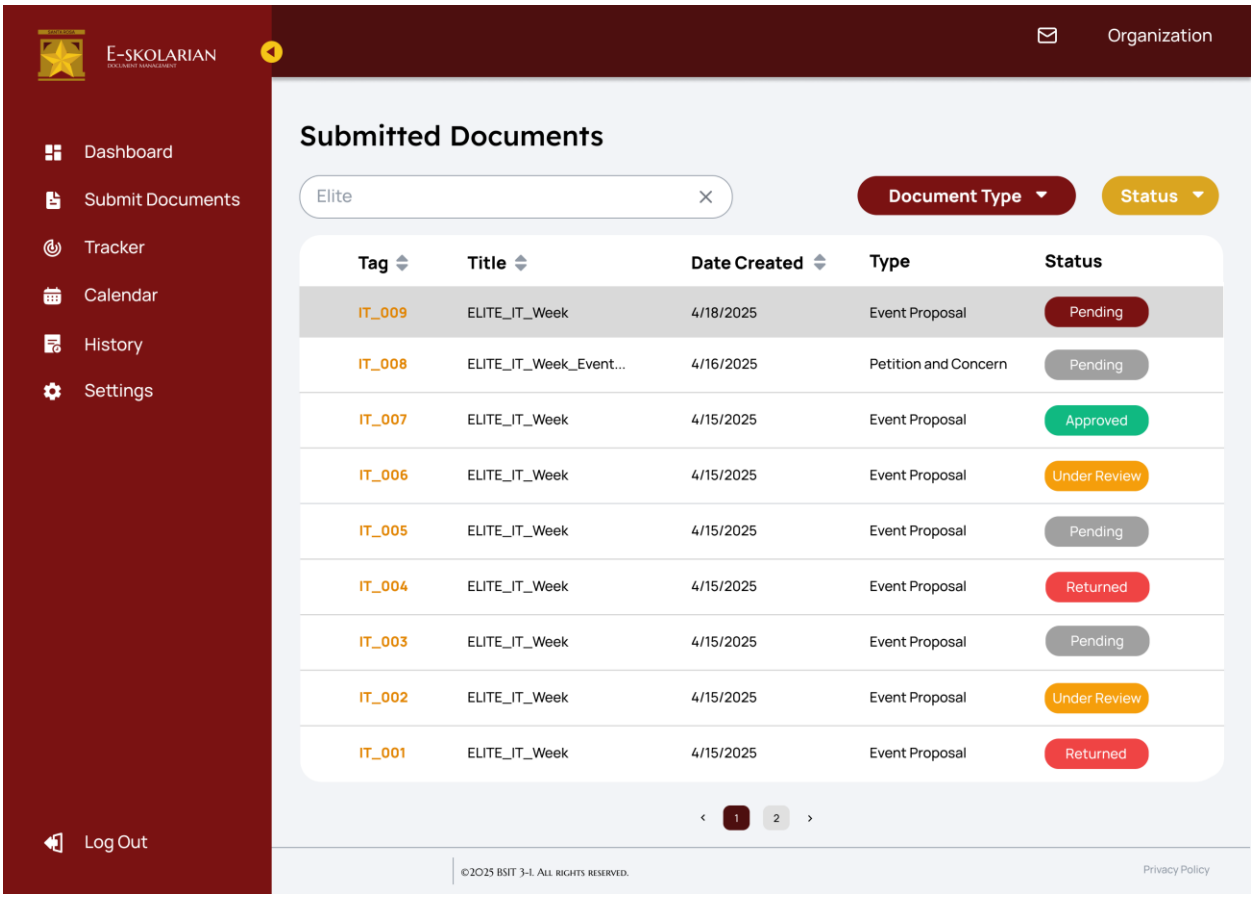


Figure 13. Submitted Document Page

Figure 14. Submitted Document Page Mobile View

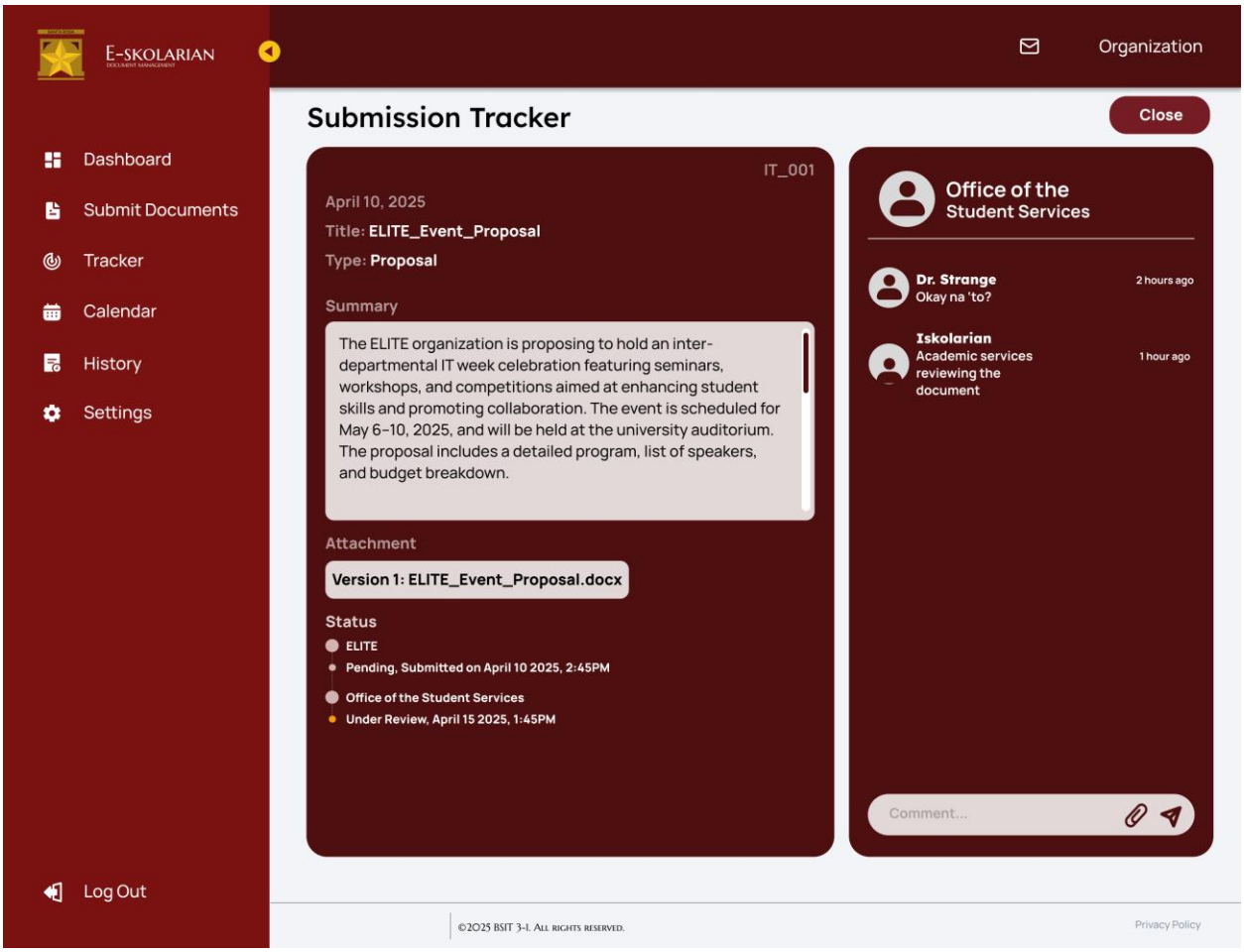


Figure 15. Submission Tracker Page

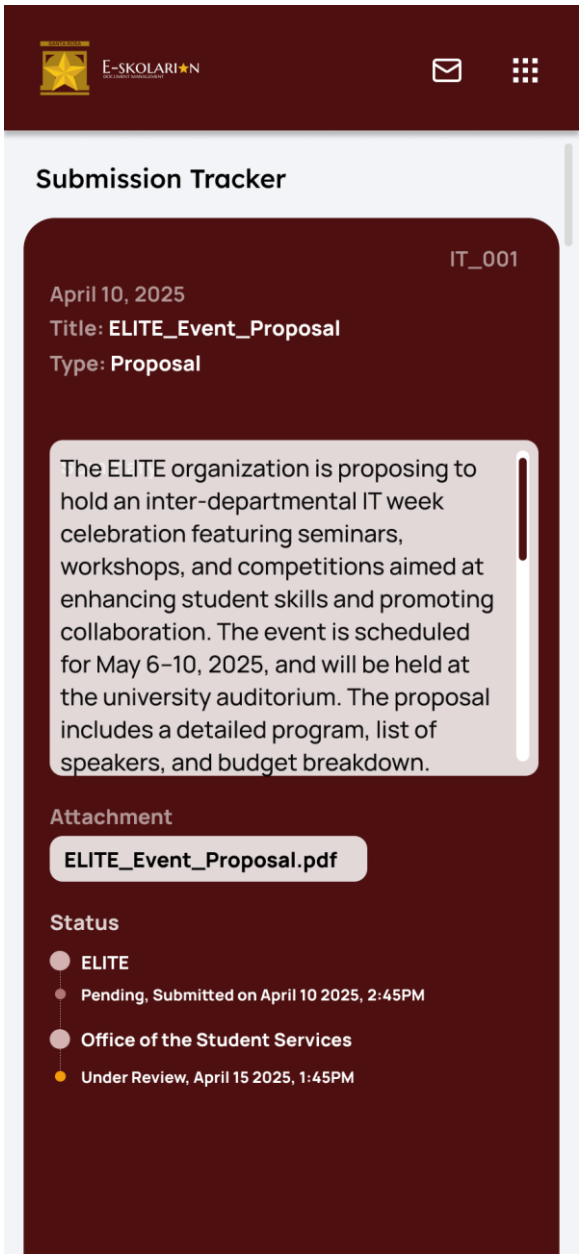


Figure 16. Submission Tracker Page Mobile View

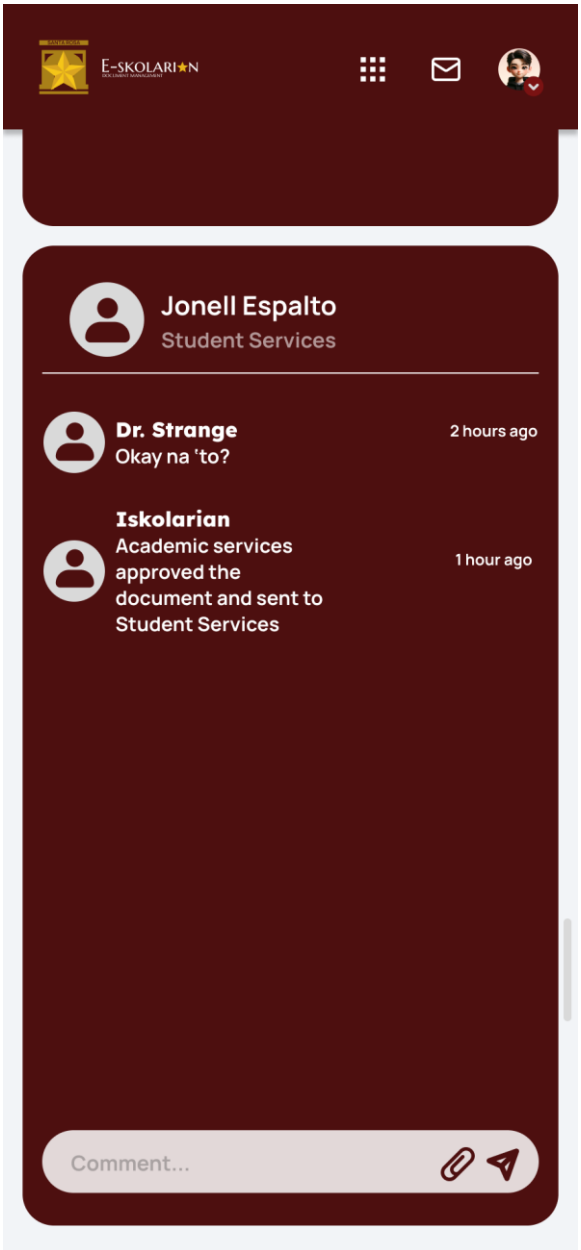


Figure 17. Feedback and Comments Section Mobile View

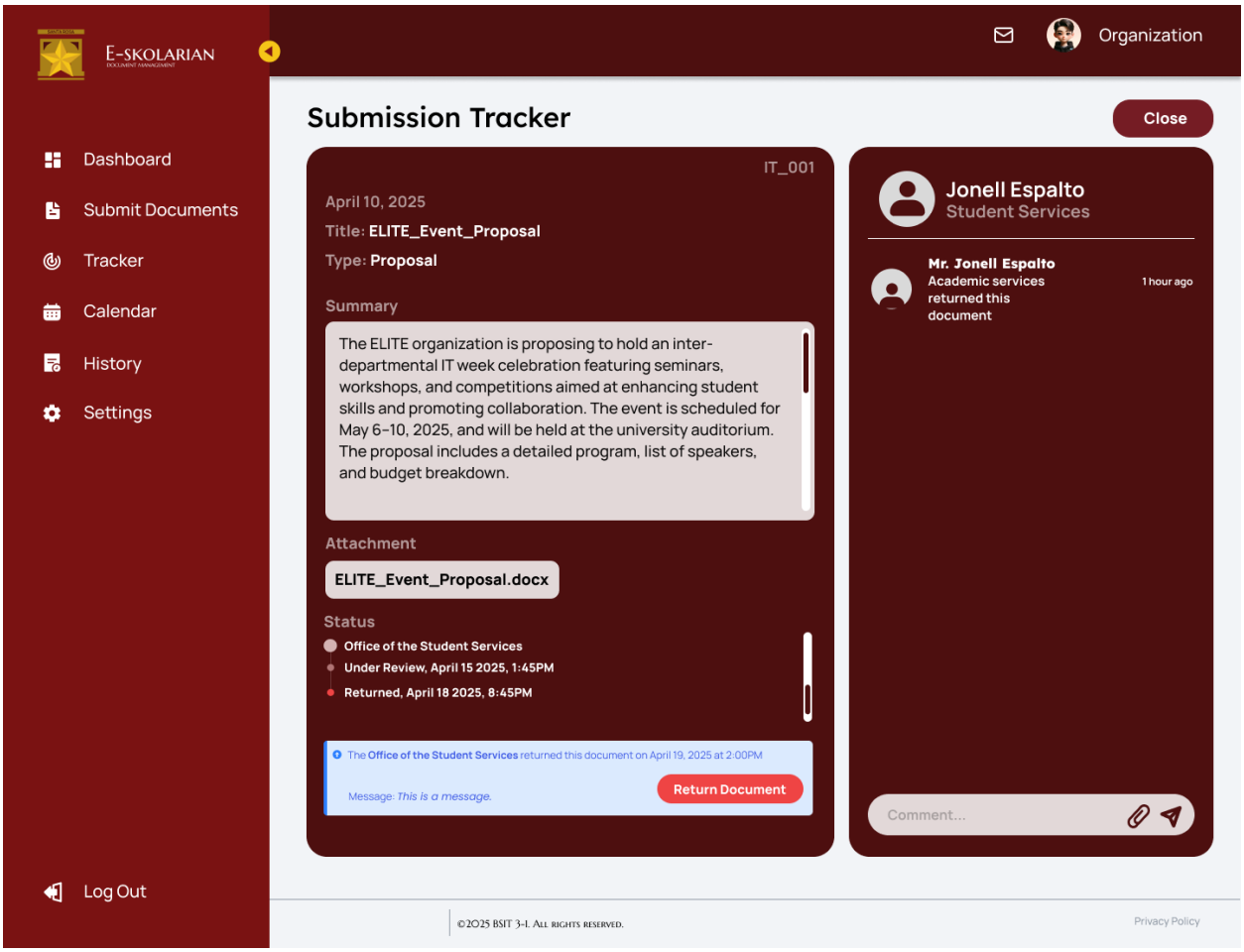


Figure 18. Submission Tracker Page (Return Document Preview)

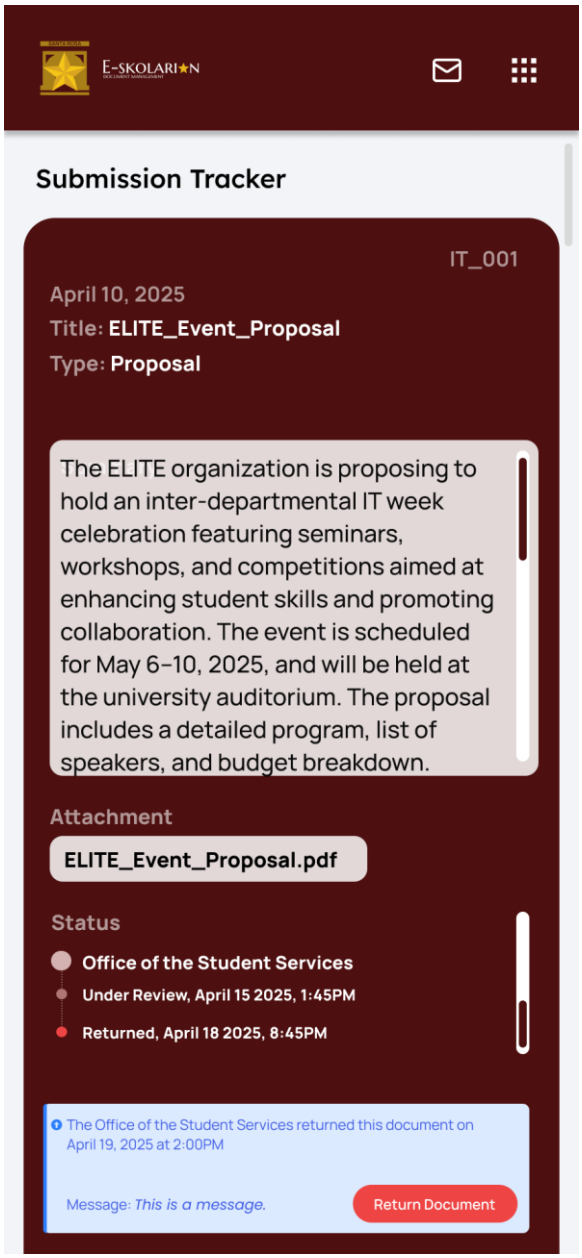


Figure 19. Submission Tracker Page (Return Document Preview) Mobile

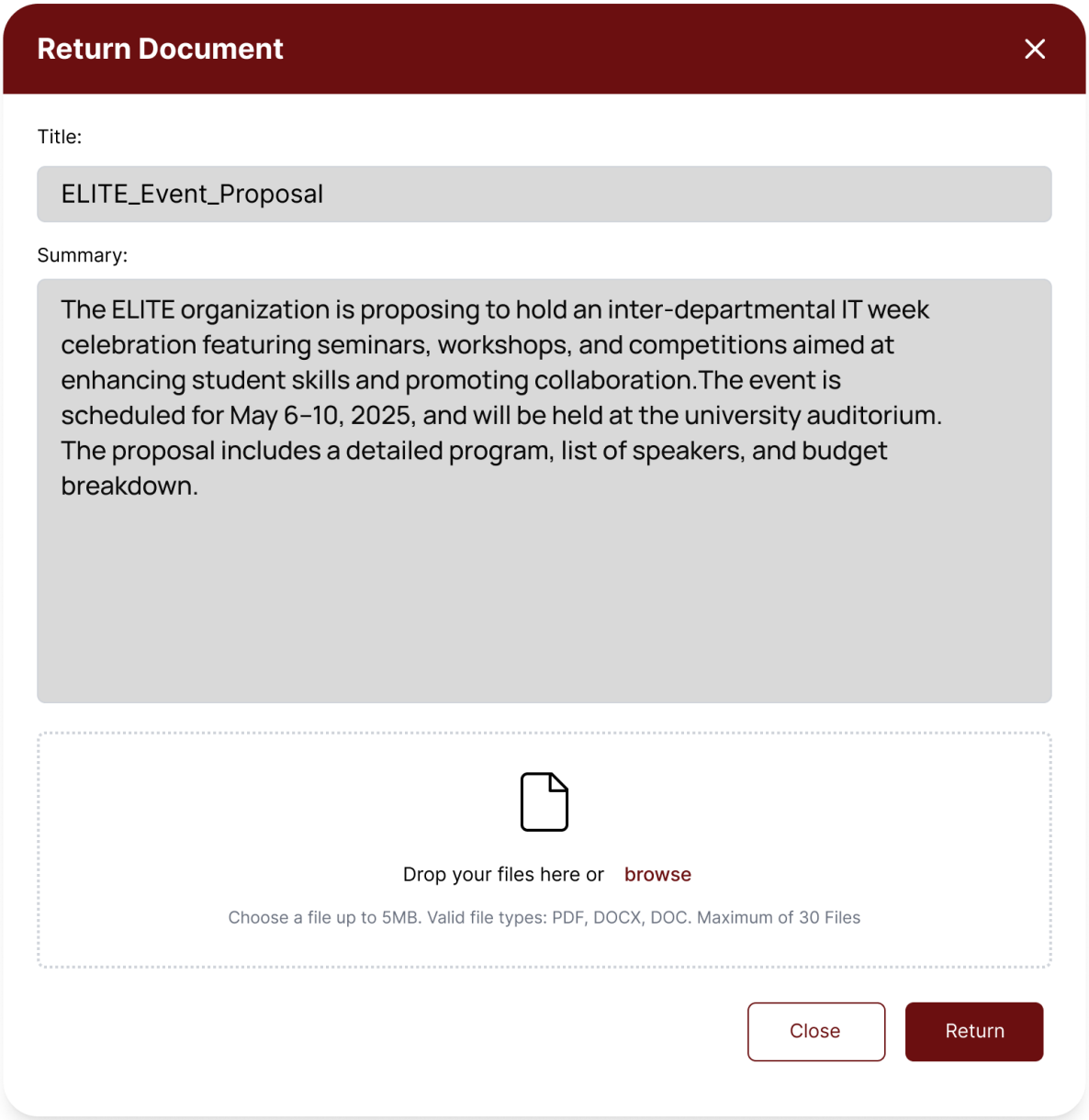


Figure 20. Return Document Pop-up Modal

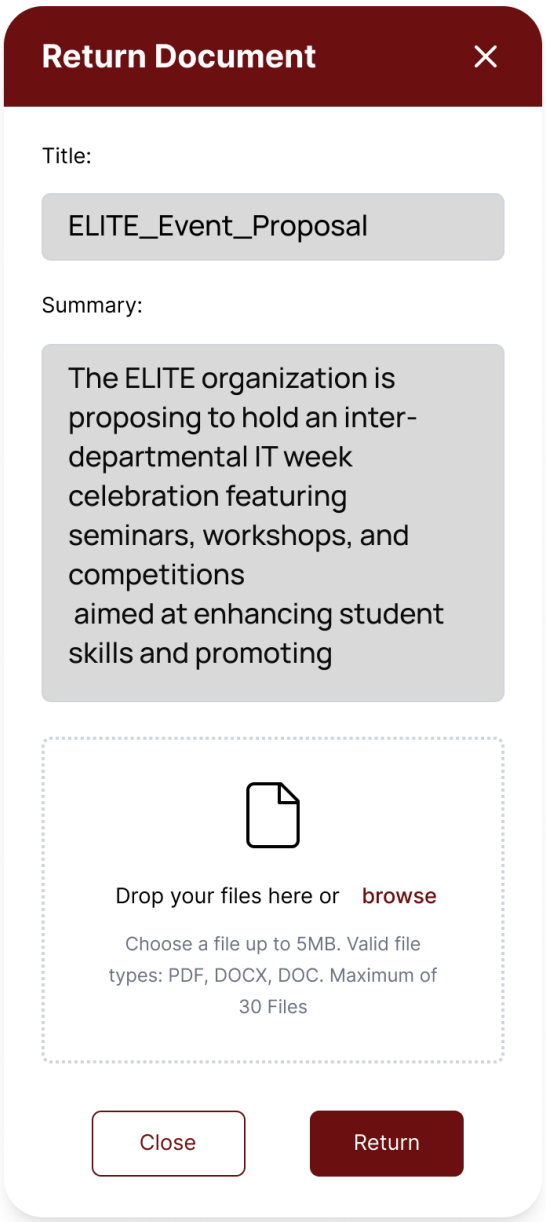


Figure 21. Return Document Pop-up Modal Mobile View

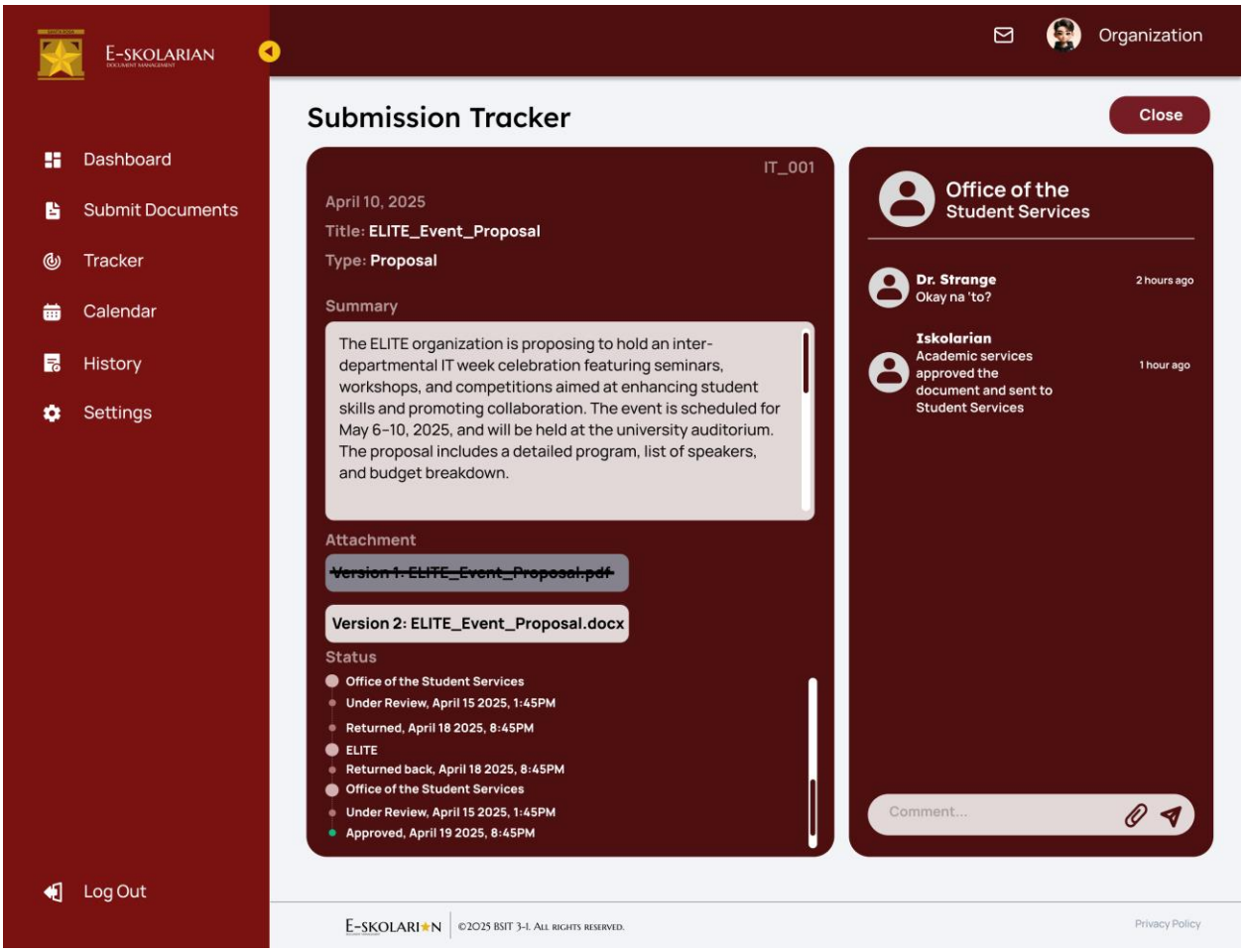


Figure 22. Submission Tracker Page (Approved Document Preview)

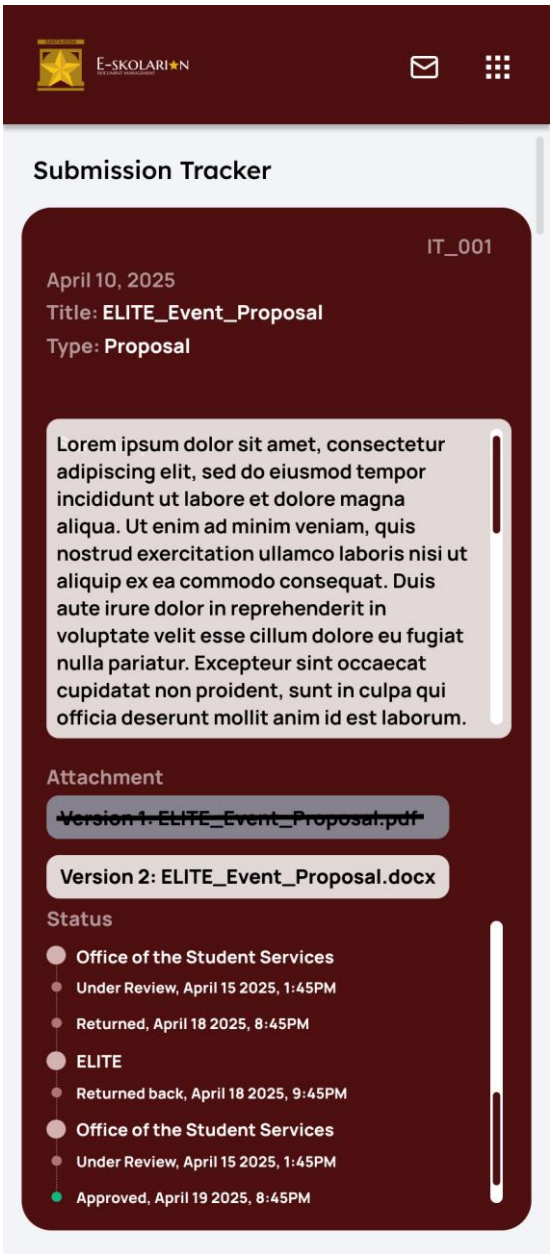


Figure 23. Submission Tracker Page (Approved Document Preview) Mobile

3.1.4 Functional Requirements

Spec ID	Specification	Business Rules/ Data Dependency

	Description	
STS-001	The Documents page should present a sortable search-based table displaying Tag, Title, Date, Type and Status information.	The table can be filtered through Search by matching any part of the Title, Tag or Type values.
STS-002	Documents should display their active workflow status badges including Pending, Under Review, Approved or Return.	Status values produced by the document approval workflow engine use colored markers for easy recognition.
STS-003	The user will be directed to the Submission Tracker page of a particular document after selecting a row through a click.	The tracking page receives document metadata either from query parameters or session state.
STS-004	A Tracker Status Flow must indicate real-time updates which display the present process stage.	The review process determines when status timeline progresses while visual signals connect to the workflow rules.
STS-005	The comments section within the Tracker page should present information in descending chronological order using user names and their roles with timestamps.	The document thread table contains comments that relate through document ID values.
STS-006	Users can write text messages while having the option to add file attachments to their comments with this input field.	Owners of documents along with reviewers possess sole access to comment functions. Files that users need to attach to the document must use the approved file types such as PDF, DOCX, JPG, JPEG or PNG and the file size should not exceed to 5MB, and the written text messages should not exceed to 255 words.
STS-007	Users can access the "Return Document"	The program draws information from the document status field retrieved from the

	button specifically when documents appear with the "Returned" status in the tracker.	database.

3.1.5 Field level specifications

Form Elements:

Submitted Documents Page

Call-out	Field Label	UI Control	Mand?	Editable?	Data Type	Value Set	Default Value	Data Example	Data Source
STF-01	Tag	Text Label	Yes	No	String	Alphanumeric, 6–10 characters	N/A	IT_004	Submission Document DB
STF-02	Title	Hyperlink Text	Yes	Yes (via modal)	String	Free text (max 100 characters)	N/A	Event Proposal	Document table
STF-03	Date Submitted	Text Label	No	No	Date	yyyy-mm-dd	N/A	2024-10-22	System-generated
STF-04	Document Type	Dropdown Filter (above table)	No	No	Enum	Request, Form, Letter, etc.	All Types	Event Proposal	Document metadata table
STF-05	Status	Label	No	No	Enum	Pending, Under Review, Approved, Returned	N/A	Under Review	Workflow engine status
STF-06	Search Field	Text Input	No	Yes	String	Free text	Empty	Document Metadata	Client-side

									filter
STF-07	Pagination Control	Navigation Buttons	No	No	Integer	Pages based on record count	Page 1	1, 2, 3, etc.	Front-end logic

Submission Tracker Page

Call-out	Field Label	UI Control	Mandatory?	Editable?	Data Type	Value Set	Default Value	Data Example	Data Source
STF-08	Document Title	Header Text	Yes	No	String	Auto-fetched	N/A	Request for Reservation	Document metadata
STF-09	Document Status Flow	Progress Bar	Yes	No	Enum/Steps	Pending, Under Review, Approved, Returned	Pending	Under Review of Campus Administrator	Workflow engine
STF-10	Comment Section	Textarea + File Input	Yes	Yes	Text + File	Free text (max 255 chars), Allowed files: .pdf, .docx, .jpg, .png	Empty	Please update document. + file	Comment DB (linked to doc ID)
STF-11	Add Comment Button	Button	No	Yes	Trigger	N/A	N/A	Click	Event-based insert
STF-12	Comments Log	Thread Display	No	No	Record Group	Chronological display	None	Emy (Campus Administrator)	Chatbox

STF-13	Attach File	File Input	No	Yes	File	Allowed file types	Empty	General Activity.docx	Uploaded files table
STF-14	Upload File Drop box	File Input	Yes	Yes	File	Allowed file types such as PDF and DOCX.	Empty	Elite Event Proposal.pdf	Uploaded files table
STF-15	Return Document Button	Button	No	No	Boolean	Yes/No	Disabled if Pending or Approved	Returned, April 18, 2025, 8:45PM	User action to notification

Form Business Rules and Dependencies:

Submitted Documents Page

Field Label	Validation / Business Rules	Error Messages	Data Dependencies	Additional Info / Notes
Tag	Auto generated, alphanumeric, read-only	N/A	Generated from DB upon document creation	Unique identifier for each document
Title	Required, must not exceed 100 characters	Title is required. Title must be under 100 characters.	Editable only before final approval	Acts as a hyperlink to view submission tracker
Date Submitted	System-generated on document upload	N/A	Populated based on upload timestamp	Read-only for users
Document Type	Selectable from predefined list in dropdown filter	Invalid document type selected.	Filters table records based on selected	Used for categorization and filtering

			type	
Status	Updated based on document workflow stages	N/A	Changes based on reviewer/campus administrator actions	Styled as colored badge (visual feedback)
Search Field	Must not include special characters except dash and underscore	Invalid character used in search.	Filters only based on document titles or tags	Client-side dynamic filtering
Pagination Control	Only accessible when results exceed limit (e.g., 10 per page)	N/A	Depends on total record count	Default display is 10 documents per page

Submission Tracker Page

Field Label	Validation / Business Rules	Error Messages	Data Dependencies	Additional Info / Notes
Document Title	Auto filled from database, non-editable	N/A	Depends on document selected from Documents List	Displayed as heading of the tracker view
Document Status Flow	Status automatically updates depending on campus administrator/editor actions	N/A	Tied to document status field in database	Each stage is visually represented in a progress bar
Comment Section	Required if adding follow-up or feedback; file optional	Comment cannot be empty.	Links to document ID and user session	Accepts attachments; supports threaded conversations
Add Comment Button	Enabled only when comment field is not	Please type a comment before	Validates comment	Submits data to the database and

	empty	submitting.	input	refreshes comment log
Comments Log	Auto updated after each comment submission	N/A	Pulls from comments table based on document ID	Displays commenter name, role, timestamp, and message
Attach File	Only accepts specific formats;	Invalid file type.	Optional unless required by campus administrator	Allowed: PDF, DOCX.
Upload File Dropbox	Only accepts specific formats;	Invalid file type.	Required by campus administrator	Allowed: PDF, DOCX, JPG, JPEG or PNG.
Return Document Button	The button is only visible if the document status is "Returned" and the user is the original owner.	Depends on current document status and user session data.	Depends on current document status and user session data.	Disabled if document is already approved, returned, or under.

Buttons, Links and Icons:

Submitted Documents Page

Button / Link / Icon Label	OnClick Event	Other Event	Visible	Enabled vs Disabled	Navigate To	Validation	Dependencies
Document Metadata Row (Link)	Opens the submission tracker of document	Hover (underline)	Always	Always enabled	Submission Tracker Page	None	Depends on availability of document ID
Filter	Filters documents	On	Alwa	Always	N/A	Must match	Updates document list

Dropdown	by type/status	change	ys	enabled		predefined value set	dynamically
Search Field (Icon)	Filters document list based on query	On Enter key	Always	Disabled if empty	N/A	Must match allowed input format	Works only if input text is valid
Pagination Controls	Loads next/prev set of records	Hover highlight	If records > 10	Disabled on first/last page	N/A	None	Depends on total document count

Submission Tracker Page

Button / Link / Icon Label	OnClick Event	Other Event	Visible	Enabled vs Disabled	Navigate To	Validation	Dependencies
Add Comment (Button)	Submits comment and optional attachment	Press "Enter" key	Always	Always enabled	Stay on current page	Requires valid comment input	Depends on comment box content
Attachment (Icon)	Opens file upload dialog	Click "Attach file" Icon	Always	Enabled if file type is allowed	N/A	Validates file type/size	Only available in comment section
Dropbox (Button)	Opens file upload dialog	Click “Dropbox” Button	If return document modal is open	Enabled if file type is allowed	Open “Return Document Modal”	Requires file types	
Return Document (Button)	Open “Submit Documen	N/A	If status is	Disabled if status is “Pending or Approved”	Opens “Return Docum	None	Depends on document status

	t Page”		Returned Document		ent Modal”		

% of work done: Total Number of Completed Tasks/Total Number of Tasks * 100

4. System Configurations

This section outlines the technical system configuration necessary for the deployment, operation, and maintenance of the E-skolarian: PUPSRC Organization Document Management System (ODMS) — Submission Tracker & Version Control Module. It details the requirements for installation, database setup, user account management, security enforcement, application integration, testing, and documentation. These configurations are designed to ensure that the system operates efficiently, securely, and in alignment with the institution’s existing technological infrastructure and standards.

4.1. Installation and Setting Up

Purpose: Install a browser-accessible web application hosted on Amazon Web Services (AWS) and configure the necessary components to enable the Submission Tracker & Version Control Module within the institution’s environment.

Alternatives: Deployment can be adjusted for on-premises servers if needed, using internal infrastructure. The following system supports Windows, MacOS, and Linux environments.

Dependencies: The following system requires the following dependencies:

Web Server: Apache Web Server.
PHP Environment: PHP8.1 or latest version.
PHP Dependency Manager: Composer version 2.8.8 or latest.
Database: MySQL or another Laravel-supported Database.
Frontend Development Tools and Packages: Node.js and npm.

4.2. Database Configuration

Purpose: Set up the backend database (MySQL) to manage and access the students’ user account, document submissions, feedback comments, and reviewer logs.

Alternatives: The following possible alternatives may include database services like Amazon RDS if scaling or redundancy is necessary.

Customizations: A custom database table will be generated for document types, version control tracking, comment histories and submission categories that reflect student organization workflows.

Dependencies: It is a necessary step to ensure proper indexing for fast querying and data integrity checking mechanisms.

4.3. Security Configuration

Purpose: The configuration allows the following module to enhance security measures to protect document data, version records, the students' credentials and the system communication against unauthorized access or data breaches.

Alternatives: In addition to SSL/TLS encryption, optional features like intrusion detection systems, and Web Application Firewalls can be integrated later.

Customizations: Permission levels for document visibility and actions, audit logging for all document activities, and role-based access control (RBAC) will be enforced. In addition to this, Data encryption at rest and in transit will be strictly implemented.

Dependencies: Mandatory use of SSL certificates (HTTPS), encryption for stored documents and user data, and compliance with the Philippine Data Privacy Act of 2012, and secure AWS configuration.

4.4. Application Integration

Purpose: The following module should enable integration with other E-Skolarian modules in the future and prepare the system to interface with potential institutional platforms like the Notification Module.

Alternatives: Use RESTful APIs or middleware services for integration.

Customizations: Configure document metadata exchange, automate submission and status notification, and manage version control tracking during system-by-system integration.

Dependencies: Ensure availability of API endpoints, standard JSON data formats, and consistent system versioning to avoid compatibility issues.

4.5. Testing and Validation

Purpose: The following module should be capacitated to test its reliability under different conditions before full deployment. This will be done by the Quality Assurance assigned.

Alternatives: Testing will involve the following:

Unit Testing: This will be done to test the following Laravel back-end logic.
Functional Testing: To test the Submission workflow of the module.
Integration Testing: To test the modules' ability to store documents and its User Interface.
Performance Testing: To test the modules performance under different environments.
User Acceptance Testing: To test the module with actual users and campus administrators.

Customizations: Custom test cases might include multiple version uploads, review comments, notifications, return of documents, and visual tracking of statuses.

Dependencies: Availability of a staging environment on AWS, test datasets, dedicated QA team members, and involvement of student organization for User Acceptance Testing.

4.6. Documentation and Testing

Purpose: The documentation should be provided to support users in accordance to the Role-Based Access Control (RBAC).

Alternatives: Documentation delivery can be in also in a form of the following:

- User Manual
- Campus Administration Guide
- Video Walkthroughs
- Frequency Asked Questions (FAQs)

Customizations: Specific instructions will be provided for document uploading, version updating, reviewer feedback viewing, notification management, and troubleshooting common errors.

Dependencies: Ensure materials are mobile-friendly, updated with each new system, feature or patch.

5. Other System Requirements/ Non-Functional Requirements

This section outlines the non-functional requirements of the system, focusing on performance, reliability, security, and other quality-related expectations as defined by stakeholders. These requirements ensure the system operates efficiently and meets implicit operational standards under various conditions.

- Performance: The system shall respond within 2 seconds under normal load and within 5 seconds during peak usage, supporting at least 100 concurrent users.
- Availability and Recovery: The system shall maintain 99.5% uptime, with recovery time not exceeding 15 minutes in the event of failure.

- Security and Access Control: Access shall be role-based, with mandatory user authentication and encrypted (HTTPS) data transmission. Compliance with relevant data protection laws is required.
- Compatibility: The system shall be fully responsive and accessible across desktop, tablet, and mobile devices on both Android and iOS platforms.

6. Reporting Requirements

Scope:

This report provides an overview of all student organization document submissions, including their current status (e.g., Pending, Approved, Returned), processing time, and reviewer actions.

Format:

- **Primary View**: Tabular format with filters (e.g., by organization, date range, document type).
- **Supplementary Visuals**: Pie charts/bar graphs for status distribution and average processing time trends.
- **Mockup**: (Insert placeholder for a table + chart mockup showing sample data: Organization Name, Document Type, Date Submitted, Status, Reviewer Name, Days Pending.)

Data Elements and Contents Required:

- Organization Name
- Document Type (e.g., Event Proposal, Budget Request, Scholarship Form)
- Submission Date
- Current Status (Pending/Under Review/Approved/Returned)
- Reviewer Name/Department
- Processing Time (in days)
- Return of Documents Count (if applicable)

File Types and Extraction Mechanisms:

- **Export formats**: PDF (for official use), Excel/CSV (for data analysis).
- **Extraction**: Generated on-demand via the module's "Generate Report" button or scheduled auto-email to Campus Administration (e.g., weekly).

User Base and Accessibility Levels:

- **Accessible to**: Organization presidents, faculty advisers, and campus administrators.

- **Permissions**:

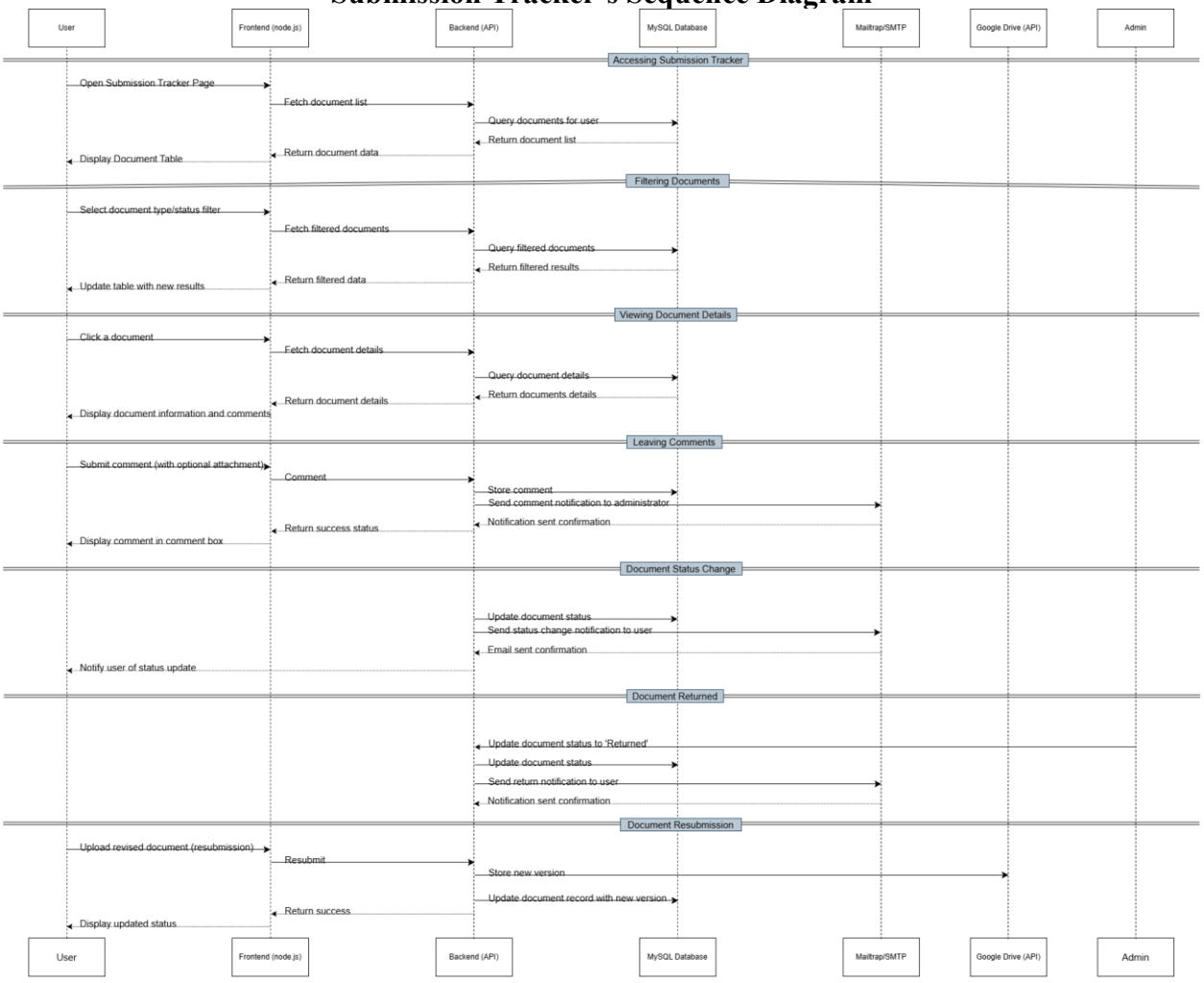
  - **Student Organizations**: View own organization's reports.
  - **Reviewers/Campus Administrators:** View all reports + export privileges.

Frequency of Extraction:

- **Real-time**: Available on-demand for tracking.
- **Scheduled**: Weekly/Monthly summaries sent to admin.istrator

7. Integration Requirements

Submission Tracker's Sequence Diagram



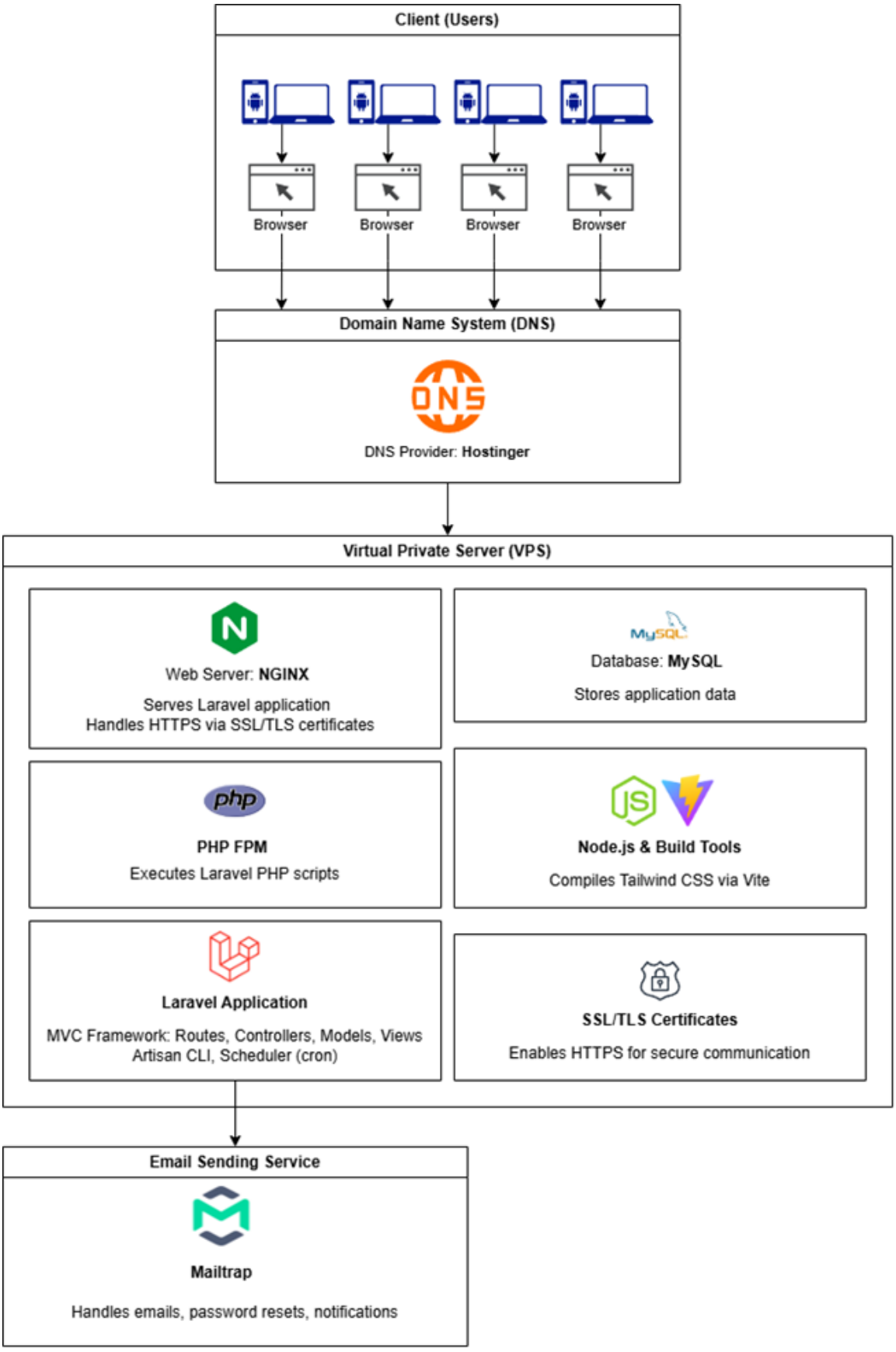
The sequence diagram illustrates the detailed interaction flow within the Submission Tracker module of the system. It begins when the user opens the Submission Tracker page through the frontend interface, which triggers a request to the backend API to fetch the list of submitted documents. The backend then queries the MySQL database for the user's documents and returns the document list to the frontend, which displays the document table to the user. The user can then filter the documents by selecting specific document types or statuses. This action prompts the frontend to request the backend for filtered documents, which again queries the database and sends the filtered results back to the frontend for display.

When the user clicks on a document to view its details, the frontend sends a request to the backend to retrieve the specific document information. The backend fetches the details from the database and returns them to the frontend, which then presents the detailed document information and associated comments to the user. The user can submit a comment, optionally with an attachment, which the frontend forwards to the backend. The backend stores the comment in the database and sends a notification to the Mailtrap/SMTP service to email the administrator. Once the email is sent, the backend confirms the successful comment submission, and the frontend updates the comment section accordingly.

If the document’s status changes, such as moving to 'Under Review' or 'Approved', the backend updates the status in the database and automatically sends a status change notification email to the user through Mailtrap/SMTP. The system confirms the email delivery and the user is notified about the document status update either via the application or email. In the event that the document is returned, the backend updates the document’s status to 'Returned' in the database and sends a return notification to the user, also confirmed via email.

For resubmissions, the user uploads the revised document through the frontend, which sends the new version to the backend. The backend stores the new file in Google Drive and updates the corresponding document record in the MySQL database with the new version link. After successfully storing the updated version, the backend sends a success response to the frontend, which then displays the updated status to the user. This entire sequence ensures seamless document tracking, commenting, status updates, and resubmission processes with real-time email notifications integrated throughout the system.

Architecture Overview Diagram


Client (Users)
Browser
Browser
Browser
Browser
Domain Name System (DNS)
DNS
DNS Provider: Hostinger
Virtual Private Server (VPS)
Web Server: NGINX
Serves Laravel application
Handles HTTPS via SSL/TLS certificates
Database: MySQL
Stores application data
PHP FPM
Executes Laravel PHP scripts
Node.js & Build Tools
Compiles Tailwind CSS via Vite
Laravel Application
MVC Framework: Routes, Controllers, Models, Views
Artisan CLI, Scheduler (cron)
SSL/TLS Certificates
Enables HTTPS for secure communication
Email Sending Service
Mailtrap
Handles emails, password resets, notifications

This integration requirements diagram shows the different integration needs for deploying and operating a Laravel-based web application.

- **Client Interaction:** Users access the application through web browsers on various devices. Their requests are directed through a **Domain Name System (DNS)** managed by **Hostinger**, which resolves domain names to the server's IP address.
- **Server Infrastructure:** The application is hosted on a **Virtual Private Server (VPS)**, which integrates several core components:
    - **NGINX Web Server**: Serves the Laravel application and manages HTTPS traffic using SSL/TLS certificates.
    - **PHP-FPM**: Executes Laravel's PHP scripts efficiently.
    - **Laravel Application**: The core application following the MVC (Model-View-Controller) architecture. It also utilizes Artisan CLI and scheduled tasks (cron jobs).
    - **MySQL Database**: Stores and manages all application data securely.
    - **Node.js and Build Tools**: Compiles assets such as Tailwind CSS using Vite for frontend optimization.
    - **SSL/TLS Certificates**: Ensures all communications are encrypted and secure via HTTPS.
- **Email Services:** For sending emails like password resets and notifications, the application integrates with **Mailtrap**, an external email testing service.

This architecture ensures a seamless, secure, and efficient environment where different systems work together to support the application's functionality and user experience.

Security and Compliance

- Encryption: AES-256-CBC (Laravel Default)
- Access Control: RBAC

7.1 Exception Handling/ Error Reporting

Exception/ Error ID	Error	Cause	Solution Strategy
ERR-STVC-001	Document Upload Failure	Large file size (>5MB), unsupported format (e.g., .exe), or network timeout.	Enforce file size/type validation upfront; retry mechanism for transient failures.
ERR-STVC-002	Submission Status Update Delay	Database latency or reviewer inactivity beyond SLA (e.g., 72 hours).	Automated reminders to reviewers; fallback to alternate approvers if unresponsive.
ERR-STVC-003	Version Conflict on Return of Document	Concurrent edits by multiple users or	Lock editing during reviews; highlight

		outdated draft overwrites.	version differences before saving.
ERR-STVC-004	Login Authentication Failure	Invalid credentials, expired sessions, or LDAP/SSO service outages.	Account lockout after 3 attempts; sync with PUP's central auth system.
ERR-STVC-005	Notification Delivery Failure	Email/SMS gateway downtime or invalid recipient details.	Retry 3x; log failures in dashboard for campus administrator follow-up.
ERR-STVC-006	Data Export/Report Generation Error	Insufficient permissions or corrupted templates.	Validate user roles; regenerate templates from backups.

8. Data Migration/ Conversion Requirements

The transition from manual and decentralized document handling to an integrated online system requires a thoughtful approach to data migration. This section outlines the necessary steps and considerations for converting existing student organization document records into the E-skolarian ODMS – Submission Tracker & Version Control Module.

Given that prior submissions are either in physical form or stored in Google Drive, the migration process will involve careful preparation, cleaning, and mapping of data to ensure consistency, accuracy, and security. These requirements aim to support a seamless and reliable transfer of historical records, allowing the system to function effectively from day one while maintaining data integrity and accessibility for all users.

Purpose: The following purpose of the module is to describe how to securely transfer existing student organization document records from paper-based and repositories into the E-skolarian ODMS - Submission Tracker & Version Control Module, ensuring continuity, data integrity, and immediate system usability upon launch.

Current Data Sources: The following list are the current data sources provided by potential users of the module:

- Physical paper documents submitted by relative student organizations.
- Digitized documents stored in a storage platform.

Types of Data to be Migrated: The following list are the types of data included in a document need to be migrated:

- Submitted documents in a Portable Document Format (PDF).
- Metadata related to document submissions like event titles, organization names, and submission dates.
- Document approval statuses.

Data Volume: In accordance to the Needs Assessment Document, it can be inferred that it was estimated to receive a maximum of 100 document submission per academic year.

Data Cleaning and Preparation: For data cleaning and preparation, a user must follow a standardized naming convention for their to be submitted documents and metadata and it must ensure completeness of key fields such as submission titles, the organization name, and the submission date.

Migration Methods: It is necessary for the module to determine the methods of migration of submitted documents; it is defined that it can be submitted by manually uploading the documents and/or manual entry or assisted uploads for basic metadata.

Security and Privacy During Migration: For the security and privacy during migration, the following files and data are needed to be encrypted during migration, however no personal data masking will be implemented, but instead strict access controls will be maintained to protect sensitive academic organization information.

8.1 Data Conversion Strategy

In this module, the following Data Conversion strategy that will be executed will be Manual and Assisted Conversion strategy to ensure the accurate migration of student organization documents, review logs, user accounts, and audit trails from legacy sources (paper records and Google Drive storage) into the new Campus Administration Document Review Module.

Steps:

1. **Data Extraction:**
    - Manually collect existing documents from physical archives and Google Drive folders.
    - Scan paper documents if necessary and convert them into digital formats (PDF, DOCX).

2. **Data Cleansing and Preparation:**
    - Review extracted files for completeness, remove duplicates, and standardize naming conventions (e.g., <OrganizationName>_<DocumentType>_<Date>).
    - Organize documents into folders categorized by organization, academic year, or document type.

3. **MetaData Structuring:**
    - Assign essential metadata to each document (e.g., student name, organization, submission date, document category) during the migration preparation phase.
    - For review logs, manually capture feedback and status changes from any available historical records.

4. **Data Import:**

- Upload cleaned and structured documents into the Campus Administration Document Review Module via the system's document management interface or database seeding scripts (if available).

5. **Data Validation:**
   - Cross-check uploaded documents against original records to ensure accuracy and completeness.
   - Test document accessibility, metadata accuracy, and visibility based on user roles (student, admin).

6. **Security Measures:**
   - Encrypt sensitive files during transfer.
   - Apply access controls to ensure only authorized users can view or modify the migrated documents.

7. **Audit Trail Reconstruction:**
   - Where possible, recreate historical timestamps and actions within the system to preserve continuity and accountability.

8.2 Data Conversion Preparation

The Data Conversion Preparation document outlines the key steps, strategies, and guidelines for transitioning student organization documents from their existing storage systems to the new Campus Administration Document Review Module within the E-skolarian platform. This migration process is essential to ensure that all historical document submissions, statuses, review feedback, and associated metadata are accurately transferred and properly aligned with the new system's requirements.

The goal of this preparation process is to provide a clear and actionable framework for data migration, ensuring that all student and administrative records are accurately reflected in the new system, preserving both the integrity and accessibility of the data. This document serves as a guide to make sure that the migration is seamless, minimizes disruption, and meets institutional and legal data governance standards.

1. **Inventory of Existing Data**
   - **Location:** The current documents are stored in the student organizations' local storage.
   - **Organization:** The documents are systematically organized within each student organization's designated folders.
   - **Types of Documents:** The primary file formats include PDFs and Word documents.
2. **Data Formats**
   - File Types:

- PDFs
- Word documents
- Any additional file formats, if applicable, can be managed similarly.

3. **Mapping to New System**

The specific details regarding data mapping will be clarified as the project progresses. Here is an initial draft:

- **File Names:** These may be mapped to "Document Title" or "Event Name."
- **Document Upload Date:** This will correspond to the "Submission Date" field in the new system.
- **File Type/Category:** This may relate to a "Document Type" field, based on how the student organizations categorize their documents (e.g., Event Reports, Activity Proposals, Scholarship Applications).
- **File Link:** The URL or reference link to a storage where the document is hosted can be included in the system as a document's accessible link.

Further mapping will be refined as the database schema is set up in the MySQL database.

4. **Resources for Preparation**
   - **Personnel Involved:** The individual responsible for the migration will be handling the process directly, but should there be a need for additional support, other team members may assist in the task.
   - **Tools Needed:**
     - Google Sheets for progress tracking.
     - A backup tool for Google Drive (or manual download/upload to a secondary Google Drive).
     - File conversion tools, if necessary, to standardize formats across all documents.
     - Laravel, for handling API-based uploads to the application.
5. **Data Cleaning**
   - **Action Plan:** Before migration, any incomplete, duplicate, or corrupted files should be removed to ensure a cleaner dataset and smoother migration process.
   - **Method:** This can be accomplished by manually reviewing each student organization folder or using Google Sheets to track and organize files for any inconsistencies.

6. **Scheduling**
   - Migration will begin once the module is fully developed, allowing ample time for preparation and testing of the migration process.

7. **Validation and Testing Plan**
   - For the validation and testing of the data migration, the following steps are recommended:

     1. **Spot Checking:** After migrating a sample of 10-20 documents, confirm that the documents are correctly listed in the new system (e.g., document name, category, submission date) and are accessible within the document viewer.
     2. **Sample Document Upload Test:** Test the document upload process by submitting a test document (PDF or Word) and tracking its journey from submission to approval, version control, and retrieval.
     3. **Comparison Check:** Perform a comparison between the migrated documents and the original files in Google Drive to ensure no data is missing or altered during the migration.
     4. **Automated Data Checks (Optional):** If feasible, implement automated checks through scripting to verify the integrity of the files (e.g., file size, metadata) against the original files in Google Drive.

8.3 Data Conversion Specifications

This structure helps ensure all data that needs to be converted and migrated into the Submission Tracker & Version Control Module is handled in an organized and systematic way. Each entry focuses on the source data, conversion process, and the corresponding notes to ensure data integrity and functionality in the new system.

Source	Source Data Element	Target	Target Data Element	Conversion Rules	Notes
Student Organization Storage	Document File	Document Review Module	Document File	No conversion is needed for the file format. The document will be uploaded as is, but ensure it is properly organized in the new system's file storage structure.	Ensure the document remains in its original format (PDF, DOCX, etc.) to ensure compatibility with the system's viewer.

Student Organization Storage	Submission Date	Administrator Document Review Module	Submission Timestamp	Convert the submission date into a timestamp format (e.g., YYYY-MM-DD HH:MM:SS).	The timestamp must account for the time zone difference (PST). Ensure accurate mapping from date format to timestamp format.
Student Organization Storage	Document Status (e.g., Pending, Approved, Returned)	Administrator Document Review Module	Submission Status	Convert the status field into corresponding numeric codes. For example, 0 = Pending, 1 = Approved, 2 = Returned.	Ensure proper mapping of status values to avoid errors in document tracking.
Student Organization Storage	Student Name	Administrator Document Review Module	Student ID or Student Name	If student ID is available, map the student name to the ID. If not, retain the student name as the identifier.	Ensure that the student IDs in the target system match the records in the source system for accurate tracking.
Review Logs	Review Feedback/Comments	Administrator Document Review Module	Review Feedback	Direct mapping of feedback/comments from the source system to the new system, no conversion needed.	Ensure that comments are properly formatted to prevent data corruption during transfer.
Review Logs	Review Timestamp	Administrator Document Review	Review Timestamp	Convert the timestamp to a consistent format (YYYY-MM-DD	Ensure the review timestamp is properly

		Module		HH:MM:SS).	mapped to the document's review history to preserve order and context.
Student Organization Storage	File Metadata (e.g., size, last modified)	Administrator Document Review Module	File Metadata (size, last modified)	No conversion required; metadata will be captured during migration for auditing and tracking purposes.	Include metadata to track document versioning and prevent duplication.
Version Control Logs	Version Number/History	Administrator Document Review Module	Document Version History	The version number will be converted into a version history format, maintaining a sequential log of document versions.	Ensure that version control is consistent to allow for accurate tracking and retrieval of older versions.

9. References

AIIM. (n.d.). What is Document Management? Retrieved from: https://www.aiim.org/What-is-Document-Management

Nesi Hemidli. (2023, March 6). *Introduction to UI/UX Design: Key Concepts and Principles*. Academia.edu.https://www.academia.edu/98036432/Introduction_to_UI_UX_Design_Key_Concepts_and_Principles?

PUP : PUP Sta. Rosa, Laguna Campus. (2023). Polytechnic University of the Philippines.https://www.pup.edu.ph/starosa/

Pup. (n.d.). Polytechnic University of the Philippines. Polytechnic University of the Philippines. https://www.pup.edu.ph/about/

Saptutorials. (2024, July 1). How to write a functional specification document. Saptutorials.in. https://www.saptutorials.in/how-to-write-functional-specification-document/

10. Open Issues

Issue ID	Issue	Raised By	Raised On	Solution/ Decision	Resolved By	Resolved On	Status
ST-085	The under-review status does not reflect from the administrator's side when the document is being under-review	Austria, Jonald Kiel M.	06/13/2025	The opening of the document in the administratos's side should already count as an under-review status	N/A	N/A	
ST-039	The interface is not responsive on mobile and tablet devices.	Juan, Zofia Dennise Z.	06/13/25	Review and update CSS breakpoints to ensure consistent layout behavior across different screen sizes.	N/A	N/A	Fail
ST-063	Unable to preview the file that was sent through the system.	Juan, Zofia Dennise Z.	06/13/25	Validate the file preview module to ensure compatibility with	N/A	N/A	Fail

				supported formats.			
ST-089	There is no character limit for the messages in the chatbox.	Juan, Zofia Dennise Z.	06/17/25	Implement a character limit for chatbox messages to prevent overflow and ensure system stability. Display a real-time counter and alert when limit is exceeded	N/A	N/A	Fail
ST-098	Error message appears after clicking; status updates only after page refresh.	Juan, Zofia Dennise Z.	06/19/25	Investigate the cause of the error message on click action. Ensure the status component is updated in real-time without requiring manual page refresh. Implement proper error handling logic.	N/A	N/A	Fail

Appendix

Mobile Supply Chain Management (MSCM) MSCM-0001