

**Ramkolek: Document Management System for Research and Project Papers Submission**

Project Documentation Submitted to the Faculty of the

School of Computing and Information Technologies

Asia Pacific College

In Partial Fulfillment of the Requirements for

Systems Analysis and Detailed Design for CS

SSYADD1

By

|  |  |
| --- | --- |
| Leila Angela B. Arcega | Jeb Vincent G. Cajayon |
| Jonlord A. Mirando | Daniella Diana H. Soquiat |
| Lyka C. Tesorero |  |

Table of Contents

[List of Figures iv](#_Toc150399946)

[List of Tables v](#_Toc150399947)

[Introduction 1](#_Toc150399948)

[Project Context 2](#_Toc150399949)

[Statement of the Problem 3](#_Toc150399950)

[Main Problem 4](#_Toc150399951)

[Specific Problems 4](#_Toc150399952)

[Objectives 4](#_Toc150399953)

[Significance of the Project 5](#_Toc150399954)

[Scope and Limitations 6](#_Toc150399955)

[Review of Related Literature / Systems 8](#_Toc150399956)

[Current System 11](#_Toc150399957)

[Technical Background 15](#_Toc150399958)

[Methodology 18](#_Toc150399959)

[Analyses 20](#_Toc150399960)

[Requirements Analysis 20](#_Toc150399961)

[User Roles and Description 20](#_Toc150399962)

[User Stories 21](#_Toc150399977)

[Event Tables 24](#_Toc150399978)

[Use Case Diagram/s 26](#_Toc150399979)

[Use Case Full Description 27](#_Toc150399980)

[System Analysis and Design 33](#_Toc150399981)

[Context Diagram 33](#_Toc150399982)

[Data Flow Diagrams 34](#_Toc150399983)

[Entity-Relationship Diagrams 37](#_Toc150399984)

[Activity Diagrams 41](#_Toc150399985)

[Class Diagram 50](#_Toc150399986)

[Sequence Diagrams 51](#_Toc150399987)

[State Machine Diagrams 55](#_Toc150399988)

[Package Diagram 56](#_Toc150399989)

[Component Diagram 56](#_Toc150399990)

[Deployment Diagram 57](#_Toc150399991)

[Conclusion 58](#_Toc150399992)

[References 59](#_Toc150399993)

[Appendices 61](#_Toc150399994)

[Appendix A: Product Roadmap 61](#_Toc150399995)

[Appendix B: Minutes of the Meetings 63](#_Toc150399996)

# List of Figures

[Figure 1 Current proofreading request process 11](#_Toc151359227)

[Figure 2 Proofreading process 12](#_Toc151359228)

[Figure 3 Final paper submission’s level of approval 13](#_Toc151359229)

[Figure 4 Use Case diagram 26](#_Toc151359230)

[Figure 5 Context diagram 34](#_Toc151359231)

[Figure 6 Data Flow Diagram Level 1 35](#_Toc151359232)

[Figure 7 Data Flow Diagram Level 2 Manage Request 36](#_Toc151359233)

[Figure 8 Data Flow Diagram Level 2 Manage Project 36](#_Toc151359234)

[Figure 9 Data Flow Diagram Level 2 manage User 37](#_Toc151359235)

[Figure 10 Entity Relationship Diagram 38](#_Toc151359236)

[Figure 11 Activity Diagram Manage Project Create 41](#_Toc151359237)

[Figure 12 Activity Diagram Manage Project Approve 42](#_Toc151359238)

[Figure 13 Activity Diagram Manage Project Published 43](#_Toc151359239)

[Figure 14 Activity Diagram Manage Proofreading Create 44](#_Toc151359240)

[Figure 15 Activity Diagram Managing Proofreading Request Endorse/Approval 45](#_Toc151359241)

[Figure 16 Activity Diagram Managing Proofreading Request Assign 46](#_Toc151359242)

[Figure 17 Activity Diagram Access System 47](#_Toc151359243)

[Figure 18 Activity Diagram Manage User 48](#_Toc151359244)

[Figure 19 Activity Diagram Generate Report 49](#_Toc151359245)

[Figure 20 Class Diagram 50](#_Toc151359246)

[Figure 21 System Sequence Diagram Manage Project Create 51](#_Toc151359247)

[Figure 22 System Sequence Diagram Manage Project Approve 51](#_Toc151359248)

[Figure 23 System Sequence Diagram Manage Project Publish 52](#_Toc151359249)

[Figure 24 System Sequence Diagram Manage Proofreading Request Create 52](#_Toc151359250)

[Figure 25 System Sequence Diagram Manage Proofreading Request Assign 53](#_Toc151359251)

[Figure 26 System Sequence Diagram Manage User 53](#_Toc151359252)

[Figure 27 System Sequence Diagram Generate Report 54](#_Toc151359253)

[Figure 28 System Sequence Diagram Access System 54](#_Toc151359254)

[Figure 29 State Machine Diagram Proofreading Approval 55](#_Toc151359255)

[Figure 30 State Machine Diagram Project Approval 55](#_Toc151359256)

[Figure 31 Package diagram 56](#_Toc151359257)

[Figure 32 Component Diagram 56](#_Toc151359258)

[Figure 33 Deployment Diagram 57](#_Toc151359259)

# List of Tables

[Table 1 Business process 17](#_Toc151358492)

[Table 2 Sprint table 19](#_Toc151358493)

[Table 4 User Roles and Description 20](#_Toc151358494)

[Table 5 User stories 24](#_Toc151358495)

[Table 6 Event table 25](#_Toc151358496)

[Table 7 Use Case Full Description Manage Project 28](#_Toc151358497)

[Table 8 Use Case Full Description Manage Proofreading Request 30](#_Toc151358498)

[Table 9 Use Case Full Description Access System 31](#_Toc151358499)

[Table 10 Use Case Full Description Manage User 32](#_Toc151358500)

[Table 11 Use Case Full Description Generates Reports 33](#_Toc151358501)

[Table 12 Data Dictionary 41](#_Toc151358502)

# Introduction

With the rapid rise and prevalence of digitalization, technology has been imbedded into every aspect of regular life, including education. The wave of digitalization changed the way students learn, collaborate, and engage with information. As a side effect of the pandemic, schools have been opened to employ virtual classrooms, online modules, and collaboration tools as part of their operations among other means [1]. As traditional methods of teaching and learning gave way to more innovative approaches, one such distinctive technique that emerged along with the others is gamification.

It is a crucial element of an effective document management system (DMS), designed with a focus on user needs. This system allows for organized document management, easy searchability by title or keyword, team collaboration, and efficient document storage. These features contribute to enhanced information retrieval, heightened security, better management of academic resources, and cost-effective document handling and access.

Asia Pacific College (APC) utilizes email and MS Teams in the collection of research and project paper submission from students. File transfers and communication between the people involved in the process are done through email while the temporary storage of files is held on MS Teams. The proposed project aims to address the challenges faced in the current system. The system will aid in students and faculty members of Asia Pacific College in a hybrid learning environment where they can request to edit submitted files from student, request editing and submit documents by the library department for all the members of Asia Pacific College to see.

## Project Context

Asia Pacific College (APC) is a tertiary level scholastic institution dedicated to producing students equipped with integrity and professionalism to excel in the business and the information and communications technology industry in both the local and global market [2]. For this proposal, the specific clients are the APC library and professors for subjects that require the final versions of a project or research paper submission, assuming the paper has been proofread and successfully defended.

After APC students successfully defended their project or research, they must submit at least two printed copies of their final work to their professor and the library. However, due to the sudden COVID pandemic that struck in 2020, schools had to adapt to a more digital environment. This led to the alteration of the previous method involving physical copies, instead having the students submit the final version of their papers by sending softcopies through email or by uploading them in a folder located in MS Teams with the final versions stored in a repository also in MS Teams.

Submitting papers through email can pose some challenges for both the students and professors since sending and receiving them through email can risk getting them buried under other unrelated emails in the inbox. It is also difficult to keep track of each group’s progress through email since organizing the submissions will have to be done by the professors themselves. This method also requires the professors to manually download the files, upload them to the paper repositories managed by the librarians on MS Teams and then notify the library of the new paper's addition.

Submissions through MS Teams folders make it easier for the professor to track groups and submission dates. It is also convenient on the side of the professor and the students since they use Teams for holding classes anyway. However, putting all the students’ works together in one folder that can be accessed by anyone who is part of that Team leads to a problem. This means that files can be downloaded, altered, and deleted by students, which is a serious security issue. In fact, it is for a similar reason another one of APC’s project repositories, the APC Wiki and APC Wiki2 were abandoned.

During the pandemic, there was another method that was used for paper submissions which was the APC Wiki and APC Wiki2. The APC Wikis were created with MediaWiki, which is the same open-source software that powers Wikipedia. Both APC Wikis were available to the public, not only just APC related individuals. This allowed the papers to be accessible to anyone, which posed some similar security issues. Accounts could be made by just anyone and these outsiders can change the contents of the Wiki. In addition to that, for students to put up their works on the Wikis, it was necessary that they had at least the basic knowledge of HTML since encoding their papers and editing the contents on the Wikis required the usage of HTML. After putting their paper on the website, they still need to submit a softcopy of the paper for it to be made available in the APC library system so sending files through email is still part of this method. Because of the security issues and difficulties in managing the Wikis, it was decided that this method was not ideal for project submissions and the system using only emails and MS Teams was favored instead.

## Statement of the Problem

For this project, the team aims to address the following issues:

## Main Problem

The current process of submitting project papers lacks a dedicated and organized platform for viewing, managing, and submitting project and research papers.

## Specific Problems

1. Using email for project submissions when dealing with many students can be inefficient and challenging when keeping track of the files, changes made to it, and the submission’s progress as communication and file transfers are done through email.

2. For publishing on the library system, librarians manually download, catalogue, and enter the metadata for project papers individually, which can be time consuming.

3. The current system lacks a way to generate reports on the submissions.

## Objectives

To meet the client’s needs, this project aims to do the following:

**Main Objective**

To develop a document management system designed for APC students and faculty to submit and manage project documents so that the document submission process is streamlined and the overall efficiency of managing project papers is enhanced.

**Specific Objectives**

* To implement a system that will make the project submission process transparent and centralize the storage of submitted files for organization.
* To reduce the amount of information that the librarians have to fill out when publishing a paper to the library system.
* To generate reports on submission summary, metadata analysis, and user behavior.

## Significance of the Project

The Ramkolek will benefit the following people:

**APC Students.** Students in subjects that require the final project and research papers will benefit from this project because it is intended for them to use the repository to submit their project and research papers.

**APC Faculty.** Proofreaders from the English cluster, professors of PBL (Project Based Learning) and research subjects that generate final papers, and PBL heads will benefit from this project as well because this will make it easier for them to view, evaluate, and give feedback to the students’ papers before giving it their approval.

**APC Librarians.** The project will shorten the amount of time Librarians spend on cataloging and encoding the approved research papers submitted to them. In addition to that, the project will reduce the possibility of errors in the librarians’ side since most of the encoded information comes from the people who submit them.

**Program Directors.** Having reports on the numbers and figures of papers by APC students can be used to gain insight into the changes in trends within the school. This information can be used in the updating and development of the curriculum and ensure that the school’s programs remain relevant to the trends.

**Future Developers.** The project will provide a valuable case study and framework for future development in repository systems and document management in educational institutions. These developers can learn from the Ramkolek project to create more efficient and advanced systems for managing educational documents and research papers.

**Quality Education (Sustainable Development Goal 4).** This initiative helps improve the standard of education by simplifying the procedures for managing, submitting, and accessing academic papers. It makes educational materials more accessible, encourages better teaching and learning, and speeds up curriculum development—all of which are vital components of this SDG.

## Scope and Limitations

* APC accounts will be used for logging in.
* The research and project papers must be in both Word and PDF files. Files in other formats may be included but they will not be uploaded to the Koha library system.
* Notifications for the Ramkolek will be sent through the webapp itself and email.
* The approval workflow will begin with the student request for proofreader up to the library uploading the papers to the library system.
* Reports can be generated by all types of users.
* The repository will not be available to non-APC individuals without APC accounts.
* The bar code and access record must be manually entered by the librarians.
* Ramkolek will be able to transmit data and files onto the Library System but not retrieve them.
* To measure the effectiveness of Ramkolek, at the end of the last PBL subject, surveys and interviews will be conducted on select individuals who will be provided with a demo of the webapp.
* Files uploaded can be opened in the repository but adding and removing files in a submission that is ongoing is not allowed if the paper was not returned.

# Review of Related Literature / Systems

The rapid advancement of information technology and the Internet has transformed various sectors, including education, where digital repositories play a pivotal role in document management and organization [3]. The current system at Asia Pacific College (APC) relies on email and MS Teams for paper submissions, but it faces efficiency and collaboration limitations. To address these challenges, a more dedicated and organized platform for managing and submitting project and thesis papers is proposed.

Microsoft Teams, serving as a comprehensive collaboration platform, significantly enriches online interactions by consolidating an array of communication and document management features [4]. Its seamless integration with Office 365 services augments efficient collaboration and connectivity among users. Notably, Microsoft Teams extends interactions among students and instructors, particularly in online classes, surpassing the capabilities of conventional Learning Management Systems [5]. The platform's cross-platform compatibility ensures ubiquitous access across diverse devices and operating systems, rendering it a valuable resource for educational institutions.

OneDrive, a cloud-based storage solution, offers a host of merits aligning with the objectives of our project, even without planned direct integration [6]. By providing cloud-based storage, it not only ensures data backup but also enables effortless access from multiple devices. The fine-grained control over access privileges supports personalized approaches to document sharing, which can inform our project's strategy for managing student papers. OneDrive's approach resonates with the broader trend of transitioning to cloud-based storage solutions, offering flexibility and accessibility in the management of digital resources [7].

Dropbox is an online application run by Dropbox Inc. It allows you to store files and collaborate with colleagues. You can synchronize and store files online with it with your gadgets. Dropbox has the advantage of allowing you to access your files from anywhere at any time using the internet. A sizable user base on the Internet has taken notice and adopted this innovative service, which goes beyond the standard client/server or peer-to-peer file hosting methods that mainly focused on storage [8].

GitHub, a flexible platform made for version control, code hosting, and easy teamwork. It enables remote peers and you to work together productively on projects, promoting teamwork and good project management [9]. Recently, a growing number of computer science and software engineering instructors have adopted GitHub as their main platform for a variety of instructional uses [10]. This demonstrates the platform's increasing importance in education for promoting an interactive and collaborative learning environment. It also covers hosting course materials and managing the submission and assessment of assignments.

Folderit, founded in 2014, is a software firm with roots in Germany and Estonia. Its main objective is to provide a better document management system. Three key pillars—cost-effectiveness, robust security, and user-friendliness—are seamlessly integrated by the Folderit Document Management System, which sets itself apart from other competitors [11]. Featuring powerful capabilities including automated workflows, custom metadata, advanced OCR (Optical Character Recognition) search, and access control, Folderit is a complete document management solution. It guarantees secure collaboration and effective document management, and you can easily access and manage your papers from any device thanks to its mobile-friendly design. To further ensure the safety and accessibility of your priceless material, Folderit also offers the option for local backups and upholds strict security measures, such as data encryption and numerous backups.

In summary, the current paper submission system at APC grapples with efficiency and collaboration challenges, necessitating the introduction of a dedicated platform for managing and accessing project and thesis papers. This platform is designed to streamline the submission process, enhance collaboration, and prioritize security. While digital repositories in educational institutions offer significant benefits, their effective implementation demands careful consideration of security measures and the active involvement of librarians to ensure the preservation and accessibility of research data.

Furthermore, the exploration of comparable systems like Microsoft Teams, OneDrive, Dropbox, GitHub, and Folderit underscores the profound importance of optimizing document management, storage, and collaboration within our project. By incorporating insights from these systems, we can enhance the efficiency and effectiveness of our repository system, contributing to the advancement of education and collaborative work while embracing digitalization and the principles of open knowledge exchange. These insights serve as guiding principles as we shape the direction of our project.

# Current System

In this part, we'll delve into the process from initial proofreading requests to the final submission in the Koha Library System. Professors, executive directors, proofreaders, and librarians all have essential roles, supported by tools like MS Outlook and MS Teams. This ensures that research papers undergo rigorous approval and become readily accessible for future students and researchers.

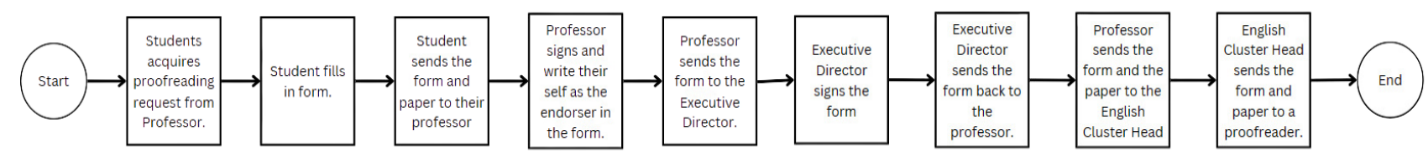


Figure 1 Current proofreading request process

Before the students defend their research or project, their paper is required to be approved by a proofreader first. To start the proofreading process for their paper, the student starts with acquiring the proofreading request form from their professor first. After acquiring and filling out the proofreading form, the student sends the form to their professor along with a copy of their paper. The professor will write down themselves as the endorser, sign the form, then send the form to the executive director. The executive director signs the form and sends the form back to the professor. Now that the form has been sufficiently filled in and signed by both the professor and the executive director, the professor will finally send the form along with the copy of the paper to the English cluster head. The English cluster head will assign the request to a proofreader and send them the form and the paper.

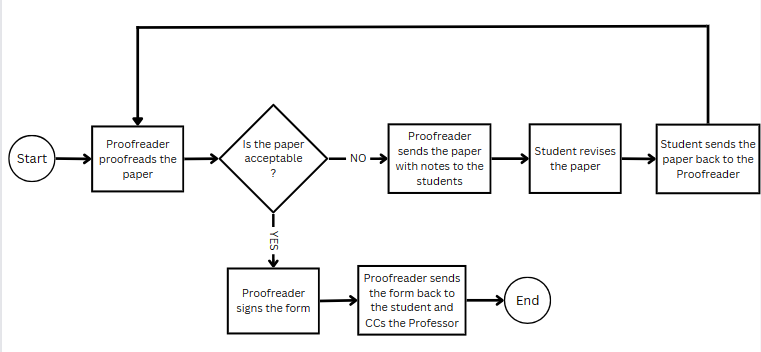


Figure 2 Proofreading process

Once the proofreader receives the form and paper, they are given up to five working days to work on the paper. The time given to the proofreader may be extended by request if the paper exceeds a hundred pages. The proofreader proofreads the paper and deems if the paper is acceptable. If not, they write down notes on paper and send it back to the student. The student will revise the paper based on the proofreader’s notes and send the paper back to the proofreader. The proofreader will check the paper again and deem if it is acceptable. If the paper is acceptable to the proofreader, they will sign the form and send the form back to the student and CCs the professor to inform them that the paper has passed proofreading.

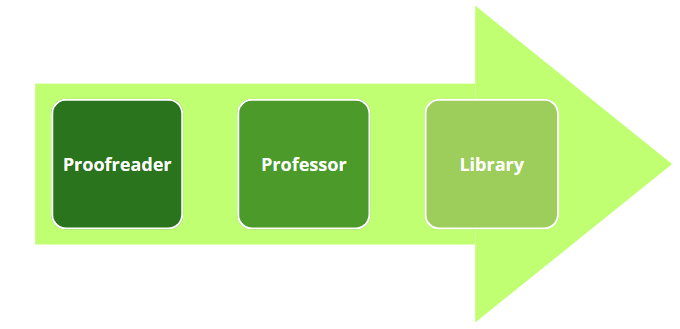


Figure 3 Final paper submission’s level of approval

After the file has been sufficiently proofread and the paper has been successfully defended, the student emails the final paper to the professor, including PDF and Doc versions of the document. The student ensures compatibility and flexibility for the professor in accessing and reviewing the work by sending the paper in PDF and Doc formats. PDF and Doc formats are also the formats that the library expects. The professor collects the final versions of the students’ papers, communicates with the executive director, and forwards the papers to them.

The professor uploads the fully approved final copies of the paper to one of the project repositories on Microsoft Teams and informs the library through email that a new paper has been added. This repository provides a consolidated platform for securely storing project documentation and making it accessible to only the parties necessary to the concluding end of the process. This stage guarantees that the student's finished document is saved in an organized location for the librarians to retrieve the papers.

Following that, a librarian retrieves the files from one of the project repositories on teams where the professor uploaded the paper. The librarian then goes on to encode the final paper's additional details, such as the title, authors, abstract, categories, keywords, barcode, and call number. The final paper, together with its accompanying details, is uploaded to the APC library system after the librarian completes the encoding process. Koha, an open-source library management system, provides efficient classification and retrieval features for library resources. By putting the student's final paper into the Koha library system, the academic community gains quick access and search capabilities to the document, allowing future students and researchers to reference and examine.

The process above involves the use of the following tools:

* **MS Outlook.** Outlook is Microsoft’s email service provider. Because APC uses Microsoft 365, Outlook is the software used by students, faculty, and library for communication.
* **MS Teams.** Teamsis a collaboration platform developed by Microsoft. It provides a centralized hub for team communication and collaboration, allowing users to chat, hold video meetings, share files, and work on documents together in real-time [12]. The professor uses Microsoft Teams to post the final copies of the student's paper to the Ramkolek Teams. This ensures that the document is securely stored in a specific location and easily accessible to all relevant parties involved.
* **Koha Library System.** Kohais an open-source library management system used for cataloging, classifying, and managing library resources [13]. It offers features for organizing and retrieving information about books, papers, and other materials in a library. After the librarian encodes the additional details of the final paper, including the title, authors, abstract, subject, and keywords, the paper is uploaded to the APC library system using Koha. This allows the academic community to efficiently classify and retrieve the document, providing quick access and search capabilities for future students and researchers.

## Technical Background

The proposed project, Ramkolek, is a web application that aims to serve as a portal for students to submit their project and research papers by establishing a centralized and organized environment for uploading the papers and managing the project documents during the submission process. The project will consolidate all the file transfers involved in paper submission into one platform and create a multi-layered submission approval workflow system that will include the actors in the current process from the proofreader up to the librarian for access control. The proofreading request process, proofreading process, and submission approval workflow will be based on the process used by the current system, except file transfers and the movement of the submission along the process will be done through access control on the submitted file. The files will be editable before it gets the final approval from the professor to allow the users to input their signature on them. As a dedicated platform, it will serve as a hub for students, faculty, and the library to access and manage the submitted papers, streamline the submission process, and reduce the amount of data encoded by the librarian.

For the development of the Ramkolek webapp, the following will be used:

* **Amazon Web Services (AWS).** AWS is the leading choice for cloud platforms with it dominating the market at 32% shares for first quarter of 2023 [14]. Its strengths lie in storage, analytics, developer tools, security, and enterprise applications [15]. It is also the cloud platform used by APC websites and with the library system being planned to be moved to AWS, it is the cloud platform choice that aligns the best for the proposed web application system Ramkolek.
* **CodeIgniter.** CodeIgniter is a powerful open-source PHP framework with a small footprint, exceptional performance, and strong security [16]. It is also known to be a beginner-friendly framework, and the team has had experience working with CodeIgniter before this project. For these reasons, it was chosen as the framework to be used on the project.

**Business Processes**

|  |  |  |
| --- | --- | --- |
| Process ID | Process  Name | Process  Details |
| P001 | Submit Paper | 1. The student sends the professor the paper’s softcopy (.pdf, .doc) through email or Teams. |
| P002 | English Proofreader Request | 1. Students fill in the Proofreading Requisition and Certification Form. 2. Students email the form to their professor. 3. Professor signs the form. 4. Professor forwards the email to the Executive Director. 5. The Executive Director signs the form and sends it back. 6. Professor sends the signed form to the English Cluster Head. 7. English Cluster Head assigns the paper to a proofreader. |
| P003 | English Proofreader Feedback | 1. Proofreader goes through the paper. 2. Proofreader sends the paper back to the student with feedback through email. |
| P004 | Approve Submission | 1. Professor receives the student’s proofread paper in email or Teams. 2. The professor reviews the paper. 3. The professor downloads the files and uploads them to the APC Teams Repository. 4. Professor emails the library to notify them of the new paper. |
| P005 | Upload to Koha | 1. The librarian downloads the files from APC Teams Repository. 2. The librarian encodes the necessary information. 3. The librarian uploads the files to the Koha system. |

Table 1 Business process

Table 1 contains a list of the processes used in the submission of final project and research papers that is being used currently in APC.

# Methodology

In developing the Ramkolek system, the team utilizes the Scrum framework to enhance collaboration, communication, and adaptability. The development process is organized into sprints, each focusing on specific items from the Product Backlog. Each sprint will primarily involve developing and testing the backlog items specified for that sprint. In the last sprint, the finished product will be presented to the client at the end of each sprint to demonstrate progress and receive feedback.

To ensure responsiveness to client needs and feedback, the team conducts Sprint Planning meetings, Daily Scrum meetings for updates and issue resolution, and Sprint Retrospectives to reflect on outcomes. This iterative and collaborative methodology ensures the evolution of the Ramkolek system into a more efficient and user-friendly document management solution.

Currently, the team is actively working on completing the system analysis and design, which includes various diagrams. This process involves collaborative meetings with the project adviser, class adviser, and project consultant to ensure a comprehensive development approach. Standup meetings are also held in order to keep track of each member’s progress.

The Scrum framework is instrumental in delivering a flexible and adaptable solution that aligns with the evolving requirements of the Asia Pacific College community.

|  |  |  |
| --- | --- | --- |
| SCSPROJ | | |
| Sprint 1(Weeks 1-4) | Sprint 2 (Weeks5-8) | Sprint 3 (Weeks9-13) |
| User Authentication | Notifications | Report Generation |
| Project Submission Form | Roles |  |
| Proofreading Request Form | Reviewing |  |
| Dashboard |  |  |

Table 2 Sprint table

# Analyses

## Requirements Analysis

### User Roles and Description

|  |  |
| --- | --- |
| Roles | Description |
| Basic User | The basic user can create, view, edit, comment on, and track their project paper submissions and proofreading requests on the system. The basic user can only generate status reports and summary reports on their own projects and proofreading requests. Students will be assigned as basic users. |
| Project Manager | This user can view, approve, return, and comment on the project submissions. For proofreading requests, they can view, endorse, return, and comment. Project managers can generate status reports, summary, and metadata reports. Professors and proofreaders are assigned as project managers. |
| Project Admin | This user can view, approve, return, and comment on the project submissions and proofreading requests. They can also assign users to be part of the approval process for projects. Project admins can generate status reports, summary, and metadata reports on all projects in the entire system. The PBL coordinator, program directors, and executive directors will be assigned as project admins. |
| Library Admin | This user can view, return, and comment on submissions from basic users after they have been approved. They categorize the project papers and publish them to the library system. Library admins can generate status reports, summary, and metadata on all projects in the entire system. Librarians will be assigned as library admins. |
| System Admin | The administrator can manages the users and can assign roles to them. They can generate status, summary, metadata, and system reports on the The system admin accounts will be given to ITRO personnel. |

Table 4 User Roles and Description

### User Stories

|  |  |  |
| --- | --- | --- |
| **Title / Use Case Name** | **User Story** | **Acceptance Criteria** |
| Manage Project | As a basic user, I want to be able to submit projects so that a project manager and/or project admin can assess them. | 1. Project submission form will appear. 2. Form cannot submit if not filled out properly. 3. A confirmation must appear whether the submission is successful or not. |
| Manage Proofreading Request | As a basic user, I want to be able to make proofreading requests for my project so that a project manager from the English department can check my work. | 1. Proofreading form will appear. 2. A proofreading request cannot be made if a project already has an open proofreading request. 3. A confirmation must appear whether the submission is successful or not. |
| Manage Proofreading Request | As a basic user, project manager, or project admin, I want to be able to see proofreading requests so that I can review the contents. | 1. Proofreading form contents will appear.  2. Basic users should only be able to access their own proofreading requests.  3. Only the project managers selected during the creation of a proofreading request should have access that particular request. 3. Project admins and the system admin can see all requests. |
| Manage Proofreading Request | As a project manager or project admin, I want to be able to review proofreading requests so that I can endorse or return them. | 1. Proofreading request contents will appear. 2. Select project managers can endorse or reject a request. |
| Manage Proofreading Request | As a project admin, I want to be able to assign proofreading tasks to project managers from the English department so that the proofreading requests can be fulfilled. | 1. Proofreading form contents will appear. 2. Only project managers from the English department will appear on the list of choices. 3. Each project manager's number of ongoing proofreading tasks is visible. |
| Manage Project | As a basic user, project manager, project admin, or library admin, I want to be able to see project submissions so I can review the contents and keep track of their progress. | 1. Project contents will appear. 2. Basic users should be able to access only their own submissions. 3. Only the professors and executive director that are indicated in the form can access the submission. |
| Manage Project | As a basic user, I want to be able to edit my project submissions so that I can correct mistakes. | 1. The filled in submission form will appear. 2. A confirmation will appear if the edit is successful or not. |
| Manage Project | As a project manager or project admin, I want to be able to approve project submissions so that I can ensure that the projects have met my standards or return them to project’s owner with feedback. | 1. Project contents will appear. 2. The status of the project is visible along with the people who have approved. 3. The feedback field is open. |
| Generate report | As a basic user, project manager, project admin, library admin, or system admin, I want to be able to generate reports on the status and progress of projects, requests, users, and so that I can assess the information. | 1. Basic users can only generate status reports and summary reports on their own projects and requests.  2. Project managers can generate status reports, summary reports, and metadata reports on projects and requests that they have access to. 3. Project admins can generate status reports and summary reports on all projects, requests, and users in the system.  4. System admins can generate all status reports, summary reports, and system reports on all projects, requests, and users on the system. |
| Manage Project | As a library admin, I want to be able to publish fully approved project paper submissions so they are integrated into the library system. | 1. Project contents will appear. 2. Project will be available on the repository and on the library system. |
| Manage user | As system admin, I want to be able to assign and change the roles of other users so that they can use the repository according to their purposes. | 1. All usernames and roles are displayed. 2. There is an option to assign a role or change it. |
| Access system | As a basic user, project manager, project admin, library admin, or system admin I want to be able to log into the system using my APC account so that I can keep an eye on submissions. | 1. Display login page  2. Access dashboard if successful. |

Table 5 User stories

### Event Tables

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Event** | **Trigger** | **Source** | **Use Case** | **Response** | **Destination** |
| User wants to manage project submission | Project submission is managed or modified | Basic user, project manager, project admin, library admin | Manage project | Create project | Basic user |
| View project | Basic user, project manager, project admin, library admin |
| Comment on project | Basic user, project manager, project admin, library admin |
| Edit project | Basic user |
| Approve or return project | Project manager, project admin |
| Publish project to library | Library admin |
| Track progress | Basic user, project manager, project admin, library admin |
| Send notification | Basic user, project manager, project admin, library admin |
| User wants to manage proofreading request | Proofreading request is managed or modified | Basic user, project manager, project admin | Manage proofreading request | Create proofreading request | Basic user |
| View proofreading request | Basic user, project manager, project admin |
| Comment on proofreading request | Basic user, project manager, project admin |
| Edit proofreading request | Basic user |
| Approve or return proofreading request | Project manager, project admin |
| Assign proofreading request | Project admin |
| Track progress | Basic user, project manager, project admin |
| Send notification | Basic user, project manager, project admin |
| User wants to request for system reports | Generate report | Basic user, project manager, project admin, library admin, system admin | Generate report | Report generation | Basic user, project manager, project admin, library admin, system admin |
| Report download | Basic user, project manager, project admin, library admin, system admin |
| System admin wants to manage a user | User is managed or modified | Administrator | Manage user | Assign role | System admin |
| Display user information | System admin |
| Send notification | Basic user, project manager, project admin, library admin, system admin |
| User wants to access the system | User accesses system | Basic user, project manager, project admin, library admin, system admin | Access system | Login | Basic user, project manager, project admin, library admin, system admin |

Table 6 Event table

### Use Case Diagram/s

In this diagram this will show a simplified version of the RAMKOLEK system for Proofreading and Project Submission Management interacting with six different users, as each user has different ways to interact with the features provided by the system.



Figure 4 Use Case diagram

### Use Case Full Description

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Manage Project | |
| Scenario: | Submit a project paper | |
| Triggering Event: | Project submission is managed or modified | |
| Brief  Description: | When a basic user starts a project submission by creating a new project, fills in the form with the project details, selects the professor from the list of project managers and project admins, and then submits it. After the creation of a project, the basic user must finish the proofreading process. The selected project manager then reviews it for approval or return. Once the project has been approved and proofread, it goes to the library admin for publishing. The library admin inputs the bar code and call number and publishes it to this system and the library system. | |
| Actors: | Basic user, project manager, project admin, library admin | |
| Related Use Cases: | Manage Proofreading Request | |
| Stakeholders: | Basic user: to submit their project papers.  Project manager: to approve the project submission.  Project admin: to approve the submission request.  Library admin: to publish the project. | |
| Preconditions: | Users are logged in. | |
| Postconditions: | Project is published to the library. | |
| Flow of Activities: | Actor | System |
| 1. Basic user creates a new project. 2. Basic user fills out the necessary information on the project form and uploads the project document in the required format. 3. Basic user selects the professor from the list of project managers and project admins. 4. Basic user submits the project. 5. Basic user goes through the proofreading process. 6. The selected project manager reviews the project. 7. Project manager chooses to approve or return it. 8. If the project is approved and proofread, the library admin receives the project for publishing. 9. Library admin fills in the call number. 10. Library admin publishes the project to the library. | 1.1 Display the project form.  4.1 Validate the project details and file format.  4.2 Save the form contents.  4.3 Display a confirmation message upon successful submission.  4.4 Send a notification of a new project to the selected project manager.  6.1 Display project details and content.  7.1 Update project status.  7.2 Display confirmation message if approval or return is successful.  7.3 Send a notification to the project’s owner and the library admin.  8.1 Display project contents and details.  10.1 Validate the data input.  10.2 Display a confirmation message upon successful publishing.  10.3 Send notification to the project’s owner. |
| Exception Conditions: | 3.1 If the basic user attempts to submit a project with invalid data, the process will not proceed until the input is corrected.  3.2 If the system encounters an issue during submission, the system will display an error message and prompt the student to retry submission.  6.1 If a project manager was not assigned to this project, they cannot access it.  6.2 If the system encounters an issue during updating the project status, the system will display an error message and prompt the professor to retry.  9.1 If the system encounters an issue during publishing, the system will display an error message and prompt the librarian to retry. | |

Table 7 Use Case Full Description Manage Project

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Manage Proofreading Request | |
| Scenario: | Submit a request to proofread a project. | |
| Triggering Event: | Proofreading request is managed or modified | |
| Brief  Description: | When a basic user requests for their project to be proofread by creating a new proofreading request, fills in the form, selects the professor and executive director from list of project managers and project admins, and then submits it. The selected project manager then reviews it for endorsement. After that, the selected project admin will review the request and sign it. Once the request has been endorsed and signed by the necessary project managers and project admins, it goes to the project admin for assignment to a project manager who will proofread it. The project admin assigns the task to a project manager to proofread it. The chosen project manager will review the project and return it with their feedback. The basic user will comply with the feedback and resubmit the proofread file. The project manager who proofread the paper will review the paper again then give their approval. | |
| Actors: | Basic user, project manager, project admin | |
| Related Use Cases: | Manage Project | |
| Stakeholders: | Basic user: to get their project proofread.  Project manager: to endorse the proofreading request or to proofread the paper.  Project admin: to approve the proofreading request or to assign the request to a proofreader. | |
| Preconditions: | Users are logged in.  Basic user has an existing and pending project submission.  The file to be proofread must be available in doc format. | |
| Postconditions: | Proofreading requests are created. | |
| Flow of Activities: | Actor | System |
| Flow of Activities:  Exception Conditions: | 1. Basic user create a proofreading request. 2. Basic user fills out all the necessary information in the form and uploads the file in a doc format. 3. Basic user selects the professor and executive director from the list of project managers and project admins. 4. Basic user submit the proofreading request form. 5. The selected project manager reviews the proofreading request and chooses to endorse or return the proofreading request. 6. Selected project admin approves or returns the proofreading request. 7. Project admin from the English program reviews the request and assigns the proofreading task to a project manager. 8. Project manager selected for proofreading reviews the project paper and returns it with their feedback. 9. Basic user submits the proofread project paper. 10. Project manager selected for proofreading reviews the project paper and approves it. | 1.1 Display the proofreading request form.  4.1 Validate the request details and file format.  4.2 Save the form contents.  4.3 Display a confirmation message upon successful submission.  4.4 Send a notification of a new proofreading request to the selected project manager.  5.1 Display request details and content.  5.2 Update project status.  5.3 Display confirmation message if approval or return is successful.  5.4 Send a notification to the basic user and project admin.  6.1 Display request details and content.  6.2 Update project status.  6.3 Display confirmation message if approval or return is successful.  6.4 Send a notification to the basic user and project admin.  7.1 Display request details and content.  7.2 Update project status.  7.3 Display confirmation message if approval or return is successful.  7.4 Send a notification to the basic user and project manager.  8.1 Display request details and content.  8.2 Update project status.  8.3 Display confirmation message if approval or return is successful.  8.4 Send a notification to the basic user.  9.1 Update request status.  9.2 Display a confirmation message upon successful return.  9.3 Send notification to the project manager.  10.1 Update project status  10.2 Send notification to the basic user. |
| Exception Conditions: | * 1. If the basic user has an existing request for the project submission, then they cannot create a new request.   2. If the basic user does not have an existing project submission, then they cannot create a new request.   4.1 If the basic user attempts to submit a request with invalid data, the process will not proceed until the input is corrected.  4.2 If the system encounters an issue during submission, the system will display an error message and prompt the basic user to retry submission  5.1 If a project manager was not assigned to this request, they cannot access it.  5.2 If the system encounters an issue during updating the project status, the system will display an error message and prompt the project manager to retry.  6.1 If a project admin was not assigned to this request, they cannot review it.  6.2 If the system encounters an issue during updating the project status, the system will display an error message and prompt the project admin to retry.  7.1 If the system encounters an issue during updating the project status, the system will display an error message and prompt the project admin to retry.  8.1 If a project manager was not assigned as the proofreader, they cannot access it.  8.2 If the system encounters an issue during updating the project status, the system will display an error message and prompt the project manager to retry.  9.1 If the system encounters an issue during updating the project status, the system will display an error message and prompt the basic user to retry.  10.1 If the system encounters an issue during updating the project status, the system will display an error message and prompt the project manager to retry. | |

Table 8 Use Case Full Description Manage Proofreading Request

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Access system | |
| Scenario: | Log into the system | |
| Triggering Event: | System is accessed | |
| Brief  Description: | When a basic user, project manager, project admin, library admin, or system admin logs into the system using their APC account. | |
| Actors: | Basic user, project manager, project admin, library admin, system admin | |
| Related Use Cases: | Manage User | |
| Stakeholders: | Basic user: to submit their project.  Project manager: to check on their students’ works;  to provide feedback and corrections to maintain the quality of projects.  Project admin: track the progress and performance of students in PBL.  Library admin: to publish the approved projects on the library system.  System admin: to manage system and ensure its functionality. | |
| Preconditions: | User has an APC account. | |
| Postconditions: | User is given access to the system.  User is redirected to their dashboard. | |
| Flow of Activities: | Actor | System |
| 1. User opens the system. 2. User navigates to the log in page. 3. User fills in the log in form. 4. User submits the form. | 2.1. Display log in form.  4.1 Validate the form content.  4.2 Confirm if the email is an APC email address.  4.3 Authenticate user account.  4.4 Complete user log in.  4.5 Redirect to dashboard. |
| Exception Conditions: | 1.1. If the user is already logged in, they will be redirected to the dashboard.  4.1. If the email is not recognized as an existing APC email address, the user cannot proceed. | |

Table 9 Use Case Full Description Access System

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Manage User | |
| Scenario: | A user’s role is changed. | |
| Triggering Event: | User is managed or modified | |
| Brief  Description: | When a system admin wants to view a user’s information and assign a role, they select the specific user they want to retrieve information about. They may change the user’s role. | |
| Actors: | System admin | |
| Related Use Cases: |  | |
| Stakeholders: | System admin: to manage a user’s role. | |
| Preconditions: | System admin must be logged into the system.  User to be viewed and assigned a role must exist.  The system should have defined roles that can be assigned to users. | |
| Postconditions: | The user’s role is successfully assigned or modified. | |
| Flow of Activities: | Actor | System |
| 1. System admin navigates to the user's dashboard. 2. System admin selects the user they want to manage. 3. System admin specifies the new role for the user or modifies the existing role. 4. System admin saves changes. | * 1. Display existing users.   2.1 Display user’s information.  4.1 Updates the user’s role.  4.2 Display a confirmation message upon successful assignment. |
| Exception Conditions: | 4.1 If the system encounters an issue during assignment, the system will display an error message and prompt the system admin to retry. | |

Table 10 Use Case Full Description Manage User

|  |  |  |
| --- | --- | --- |
| Use Case Name: | Generate Report | |
| Scenario: | Generate reports in the system | |
| Triggering Event: | Generate report | |
| Brief  Description: | When a basic user, project manager, project admin, library admin, or system admin wants to obtain a specific report from the system, they will choose a type of report allowed to them and set the criteria for the report’s basis. and request the system to generate a report for them to download. | |
| Actors: | Basic user, project manager, project admin, library admin, system admin | |
| Related Use Cases: |  | |
| Stakeholders: | Student: to access and download reports.  Project manager: to review and analyze reports and to monitor academic performance.  Project admin: to track project-based learning outcomes;  to make informed strategic decisions, assess effectiveness, and support curriculum enhancements;  to review department-specific reports and analyze data for program improvement.  System admin: to manage system permissions and report generation. | |
| Preconditions: | User is logged in.  User has the permission to generate the report they want. | |
| Postconditions: | A report is created.  The report is made available for download. | |
| Flow of Activities: | Actor | System |
| 1. User opens the reports page. 2. User selects the report type, sets the criteria, and generates the report. 3. User downloads the generated report. | 1.1. Display report type options according to their permissions.  2.1 Generate the requested report based on the type selected.  2.2 Display confirmation message that report generation is successful.  3.1 Report file download starts. |
| Exception Conditions: | 2.1 If the user does not have permission to the type of report or the criteria they set, the report cannot be generated and an error message will appear.  2.1. If the provided details by the user are invalid or incomplete, the system will not fulfill the request.  2.2 If the system encounters an issue during generation, the system will display an error message and prompt the student to retry. | |

Table 11 Use Case Full Description Generates Reports

## System Analysis and Design

### Context Diagram

This is the level 0 of the DFD (Data Flow Diagram) as it demonstrates how each of the entities interact with the system, and what the system’s function can provide to those said entities, but the system ensures that some classified information such as user information is not shared to some entities, as there are going to be limitations on how information will be shared.

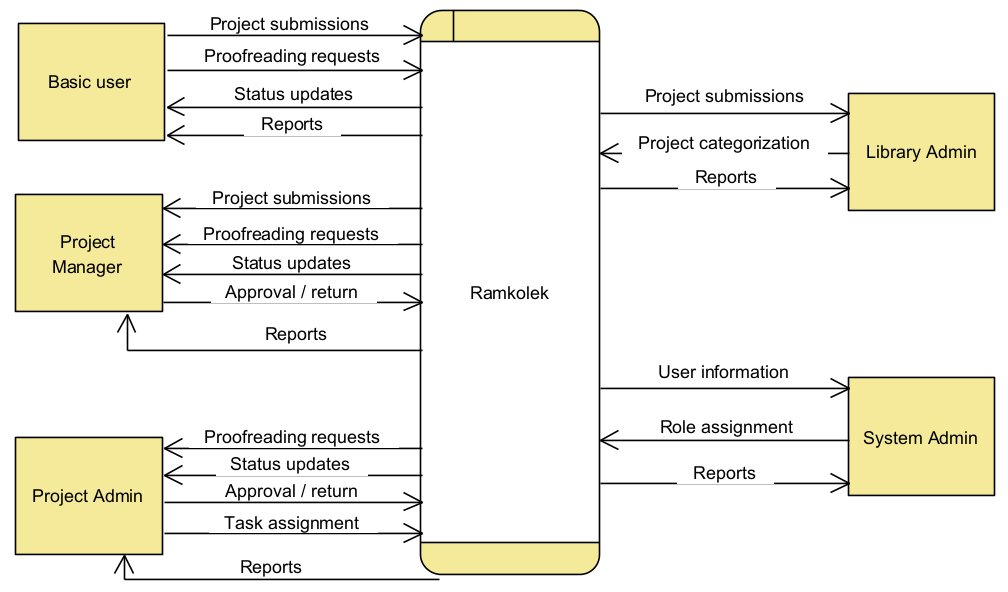


Figure 5 Context diagram

### Data Flow Diagrams

As shown in this diagram is the main system’s functions and interactions to different entities with each of the data process also interacting with the stored data, but some processes cannot function without the stored data being involved.

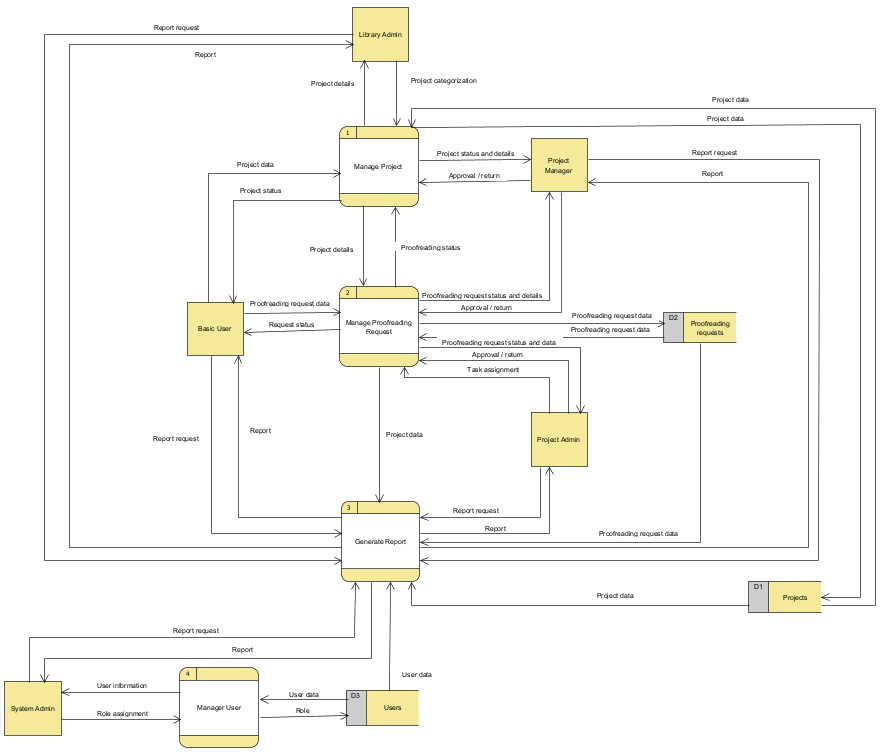


Figure 6 Data Flow Diagram Level 1

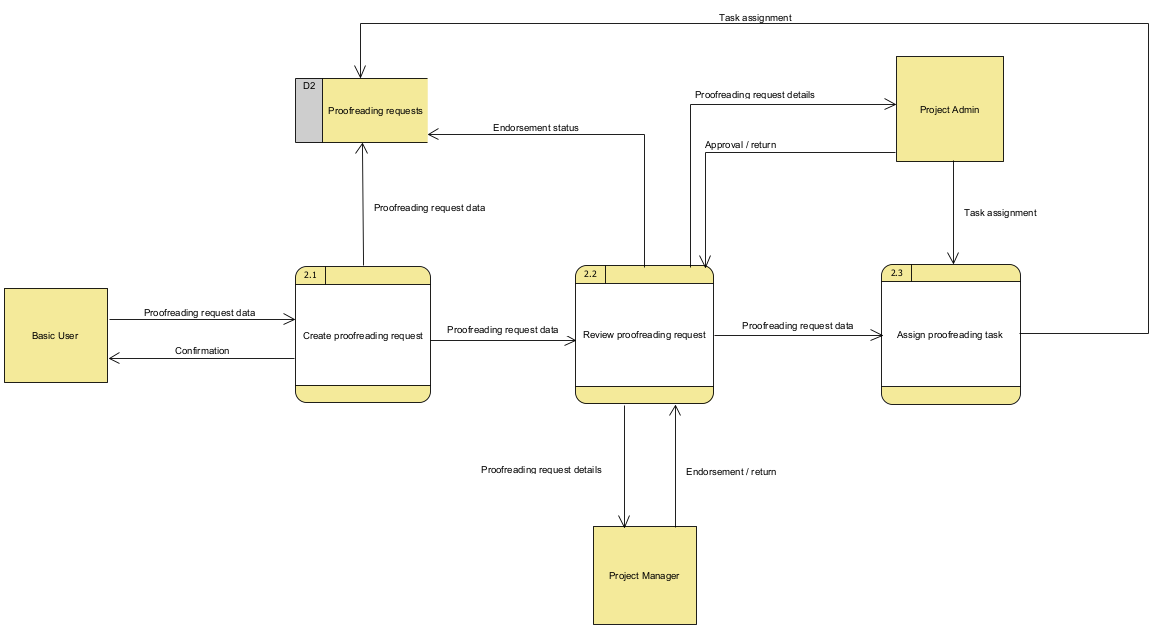


Figure 7 Data Flow Diagram Level 2 Manage Request

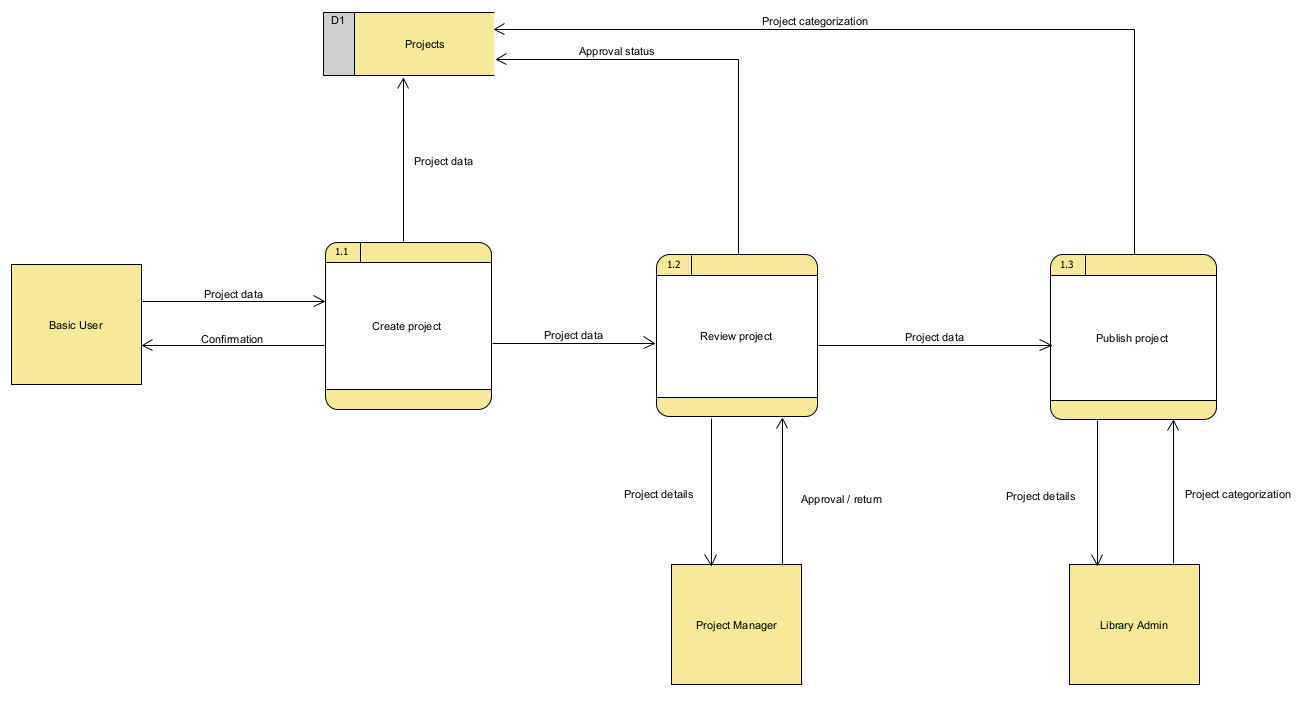


Figure 8 Data Flow Diagram Level 2 Manage Project

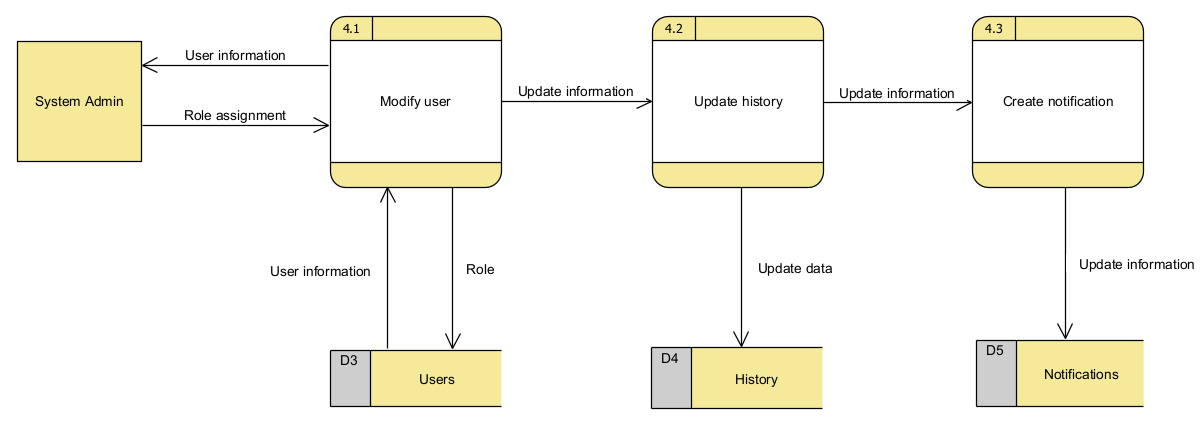


Figure 9 Data Flow Diagram Level 2 manage User

### Entity-Relationship Diagrams

This diagram shows the graphical overview of the system’s information on different entities, as the cardinality represents some users interacting with different entities.

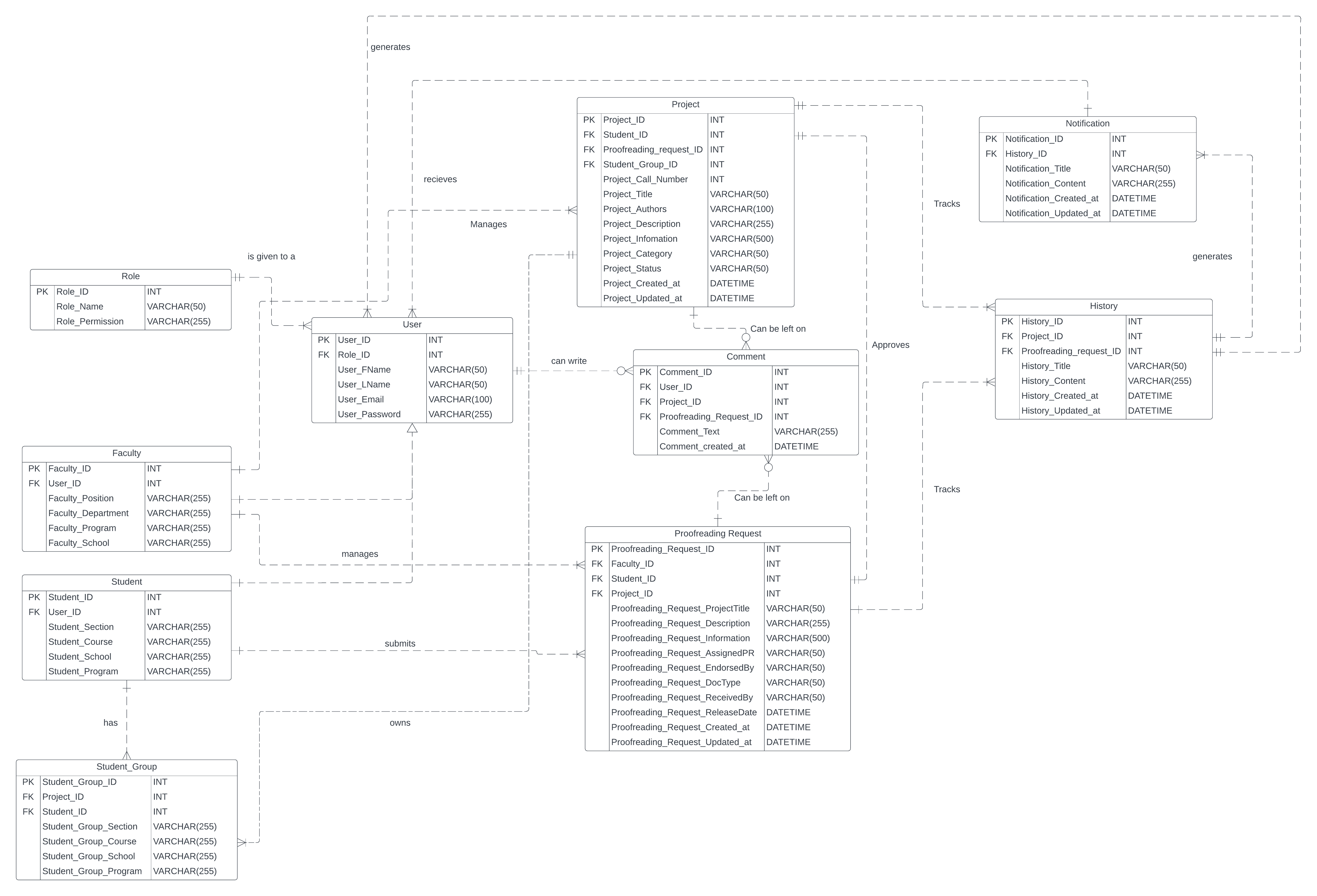


Figure 10 Entity Relationship Diagram

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Table Name | Attribute Name | Label | Type | Description | Format | Required | PK/FK |
| User | User\_ID | User id | INT | Unique identifier for each user | XXXXXXXX | Y | PK |
| Role\_ID | Role id | INT | Unique identifier for each role | XXXXXXXX | Y | FK |
| User\_FName | User first name | VARCHAR(50) | First name of the user | XXXXXXXXXX... | Y |  |
| User\_LName | User last name | VARCHAR(50) | Last name of the user | XXXXXXXXXX... | Y |  |
| User\_Email | User email | VARCHAR(100) | Email of the user | XXXXXXXX...@X.COM | Y |  |
| User\_Password | User password | VARCHAR(255) | Password of the user | XXXXXXXXXX... | Y |  |
|  |  |  |  |  |  |  |  |
| Faculty | Faculty\_ID | Faculty id | INT | Unique identifier for each Faculty | XXXXXXXX | Y | PK |
| User\_ID | User id | INT | Unique identifier for each user | XXXXXXXX | Y | FK |
| Faculty\_Position | Faculty position | VARCHAR(255) | The position of the faculty user | XXXXXXXXXX... | Y |  |
| Faculty\_Department | Faculty department | VARCHAR(255) | The faculty user's department | XXXXXXXXXX... | Y |  |
| Faculty\_Program | Faculty program | VARCHAR(255) | The faculty user's program | XXXXXXXXXX... | Y |  |
| Faculty\_School | Faculty school | VARCHAR(255) | The faculty's school | XXXXXXXXXX... | Y |  |
|  |  |  |  |  |  |  |  |
| Student | Student\_ID | Student id | INT | Unique identifier for each student | XXXXXXXX | Y | PK |
| User\_ID | User id | INT | Unique identifier for each user | XXXXXXXX | Y | FK |
| Student\_Section | Student section | VARCHAR(255) | The Student's section | XXXXXXXXXX... | Y |  |
| Student\_Course | Student course | VARCHAR(255) | The student's course | XXXXXXXXXX... | Y |  |
| Student\_School | Student school | VARCHAR(255) | The student's school | XXXXXXXXXX... | Y |  |
| Student\_Program | Student program | VARCHAR(255) | The student's program | XXXXXXXXXX... | Y |  |
|  |  |  |  |  |  |  |  |
| Project | Project\_ID | Project id | INT | Unique identifier for each Project | XXXXXXXX | Y | PK |
| Student\_ID | Student id | INT | Unique identifier for each student | XXXXXXXX | Y | FK |
| Proofreading\_Request\_ID | Proofreading request id | INT | Unique identifier for each proofreading request | XXXXXXXX | Y | FK |
| Student\_Group\_ID | Student group id | INT | Unique identifier for each student group | XXXXXXXX | Y | FK |
| Project\_Call\_Number | Project call number | INT | The project's call number | XXXXXXXX | Y |  |
| Project\_Title | Project title | VARCHAR(50) | The project's title | XXXXXXXXXX... | Y |  |
| Project\_Authors | Project authors | VARCHAR(100) | The project's authors | XXXXXXXXXX... | Y |  |
| Project\_Description | Project description | VARCHAR(255) | The project's description | XXXXXXXXXX... | Y |  |
| Project\_Information | Project information | VARCHAR(500) | The project's information | XXXXXXXXXX... | Y |  |
| Project\_Status | Project status | VARCHAR(50) | The project's status | XXXXXXXXXX... | Y |  |
| Project\_Created\_at | Project created at | DATETIME | The project's date and time of creation | YYYY-MM-DD hh:mm:ss | N |  |
| Project\_Updated\_at | Project updated at | DATETIME | The project's date and time of update | YYYY-MM-DD hh:mm:ss | N |  |
| Project\_Category | Project category | VARCHAR(50) | The project's category | XXXXXXXXXX... |  |  |
|  |  |  |  |  |  |  |  |
| Proofreading Request | Proofreading\_Request\_ID | Proofreading request id | INT | Unique identifier for each proofreading request | XXXXXXXX | Y | PK |
| Project\_ID | Project id | INT | Unique identifier for each Project | XXXXXXXX | Y | FK |
| Faculty\_ID | Faculty id | INT | Unique identifier for each Faculty | XXXXXXXX | Y | FK |
| Student\_ID | Student id | INT | Unique identifier for each student | XXXXXXXX | Y | FK |
| Comment\_ID | Comment id | INT | Unique identifier for each comment | XXXXXXXX | Y | FK |
| Proofreading\_Request\_  Description | Proofreading request description | VARCHAR(255) | The description of the proofreading request | XXXXXXXXXX... | Y |  |
| Proofreading\_Request\_  Information | Proofreading request information | VARCHAR(500) | The information about proofreading request | XXXXXXXXXX... | Y |  |
| Proofreading\_Request\_  Created\_at | Proofreading request created at | DATETIME | The proofreading request's date and time of creation | YYYY-MM-DD hh:mm:ss | N |  |
| Proofreading\_Request\_  Updated\_at | Proofreading request updated at | DATETIME | The proofreading request's date and time of update | YYYY-MM-DD hh:mm:ss | N |  |
| Proofreading\_Request\_  ProjectTitle | Proofreading request project title | VARCHAR(50) | The project title within the proofreading request | XXXXXXXXXX... | Y |  |
| Proofreading\_Request\_  AssignedPR | Proofreading request assigned proofreader | VARCHAR(50) | The assigned proofreader in the proofreading request | XXXXXXXXXX... | Y |  |
| Proofreading\_Request\_  EndorsedBy | Proofreading request endorsed by | VARCHAR(50) | The one who endorsed the project within the proofreading request | XXXXXXXXXX... | Y |  |
| Proofreading\_Request\_  DocType | Proofreading request document type | VARCHAR(50) | The document type within the proofreading request | XXXXXXXXXX... | Y |  |
| Proofreading\_Request\_  ReceivedBy | Proofreading request recieved by | VARCHAR(50) | The one who recieves the proofreading request | XXXXXXXXXX... | Y |  |
| Proofreading\_Request\_  ReleaseDate | Proofreading request release date | DATETIME | The proofreading request's date and time of release | YYYY-MM-DD hh:mm:ss | N |  |
|  |  |  |  |  |  |  |  |
| Role | Role\_ID | Role id | INT | Unique identifier for the role | XXXXXXXX | Y | PK |
| Role\_Name | Role name | VARCHAR(255) | The name of the role | XXXXXXXX | Y |  |
| Role\_Permission | Role permission | VARCHAR(50) | The permission of the role | XXXXXXXX | Y |  |
|  |  |  |  |  |  |  |  |
| Comment | Comment\_ID | Comment id | INT | Unique identifier for each comment | XXXXXXXX | Y | PK |
| User\_ID | User id | INT | Unique identifier for each user | XXXXXXXX | Y | FK |
| Project\_ID | Project id | INT | Unique identifier for each Project | XXXXXXXX | Y | FK |
| Proofreading\_Request\_ID | Proofreading request id | INT | Unique identifier for each proofreading request | XXXXXXXX | Y | FK |
| Comment\_Text | Comment text | VARCHAR(255) | The comment's text | XXXXXXXX | Y |  |
| Comment\_Created\_at | Comment created at | DATETIME | The date and time of the comment created | YYYY-MM-DD hh:mm:ss | N |  |
|  |  |  |  |  |  |  |  |
| History | History\_ID | History id | INT | Unique identifier of the history | XXXXXXXX | Y | PK |
| Project\_ID | Project id | INT | Unique identifier for each Project | XXXXXXXX | Y | FK |
| Proofreading\_Request\_ID | Proofreading request id | INT | Unique identifier for each proofreading request | XXXXXXXX | Y | FK |
| History\_Title | History title | VARCHAR(50) | The title of the history | XXXXXXXX | N |  |
| History\_Content | History content | VARCHAR(255) | the content of the history | XXXXXXXX | Y |  |
| History\_Created\_at | History created at | DATETIME | The date and time of the history created | YYYY-MM-DD hh:mm:ss | N |  |
| History\_Updated\_at | History updated at | DATETIME | The date and time of the history update | YYYY-MM-DD hh:mm:ss | N |  |
|  |  |  |  |  |  |  |  |
| Notification | Notification\_ID | Notification id | INT | Unique identifier of the notification | XXXXXXXX | Y | PK |
| History\_ID | History id | INT | Unique identifier of the history | XXXXXXXX | Y | FK |
| Notification\_Title | Notification title | VARCHAR(50) | The title of the notification | XXXXXXXX | N |  |
| Notification\_Content | Notification content | VARCHAR(255) | The content of the notification | XXXXXXXX | Y |  |
| Notification\_Created\_at | Notification created at | DATETIME | The date and time of the notification created | YYYY-MM-DD hh:mm:ss | N |  |
| Notification\_Updated\_at | Notification updated at | DATETIME | The date and time of the notification update | YYYY-MM-DD hh:mm:ss | N |  |

Table 12 Data Dictionary

### Activity Diagrams

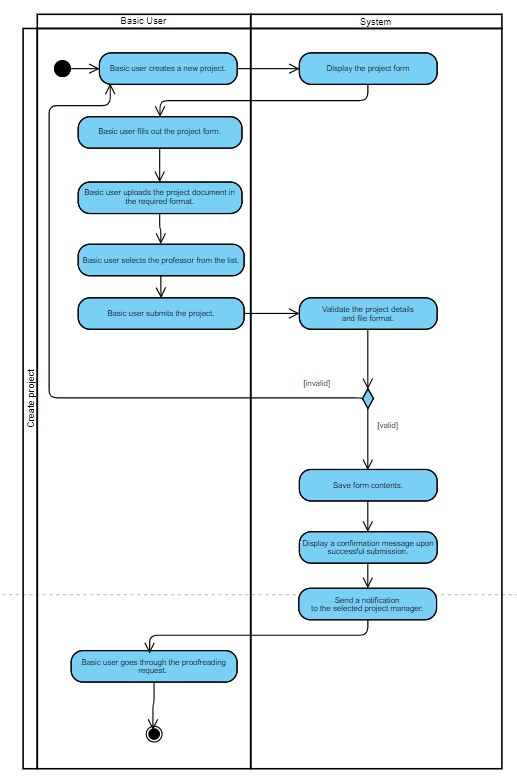


Figure 11 Activity Diagram Manage Project Create

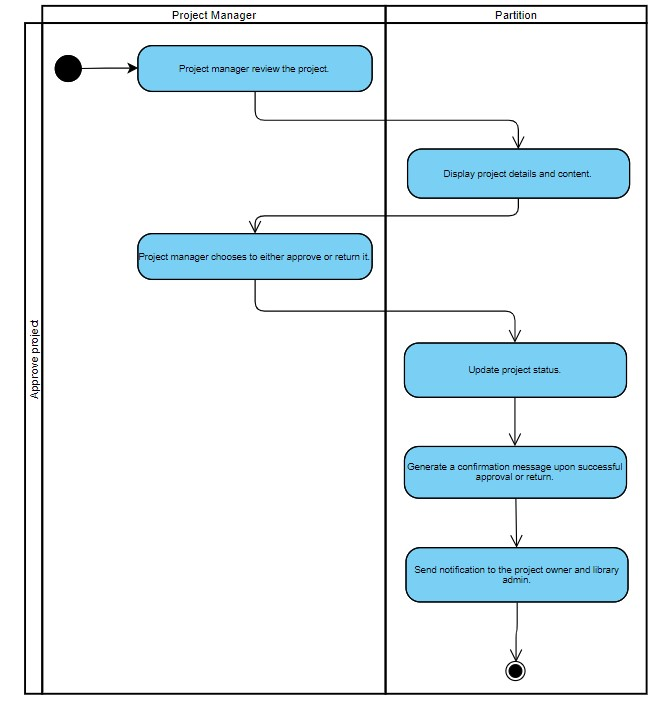


Figure 12 Activity Diagram Manage Project Approve

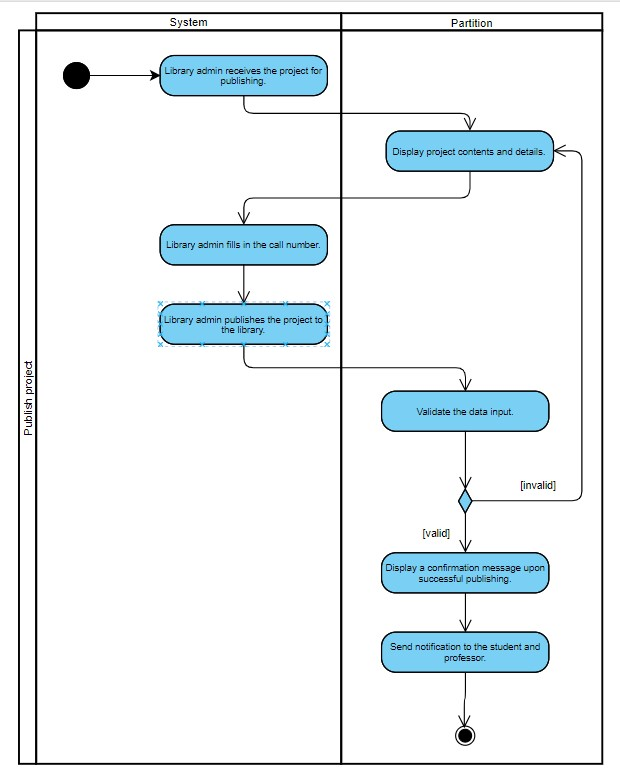


Figure 13 Activity Diagram Manage Project Published

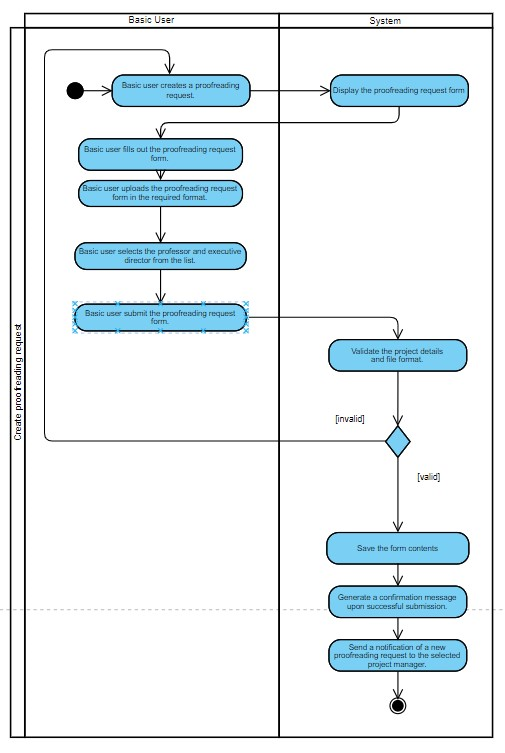


Figure 14 Activity Diagram Manage Proofreading Create

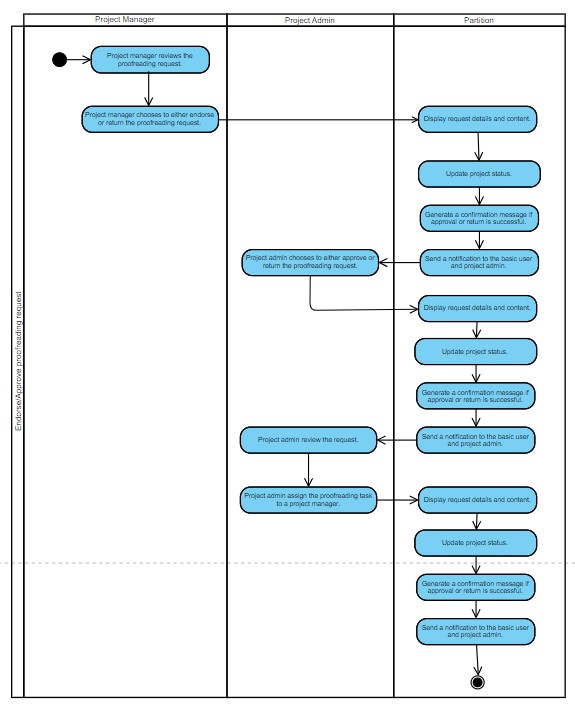


Figure 15 Activity Diagram Managing Proofreading Request Endorse/Approval

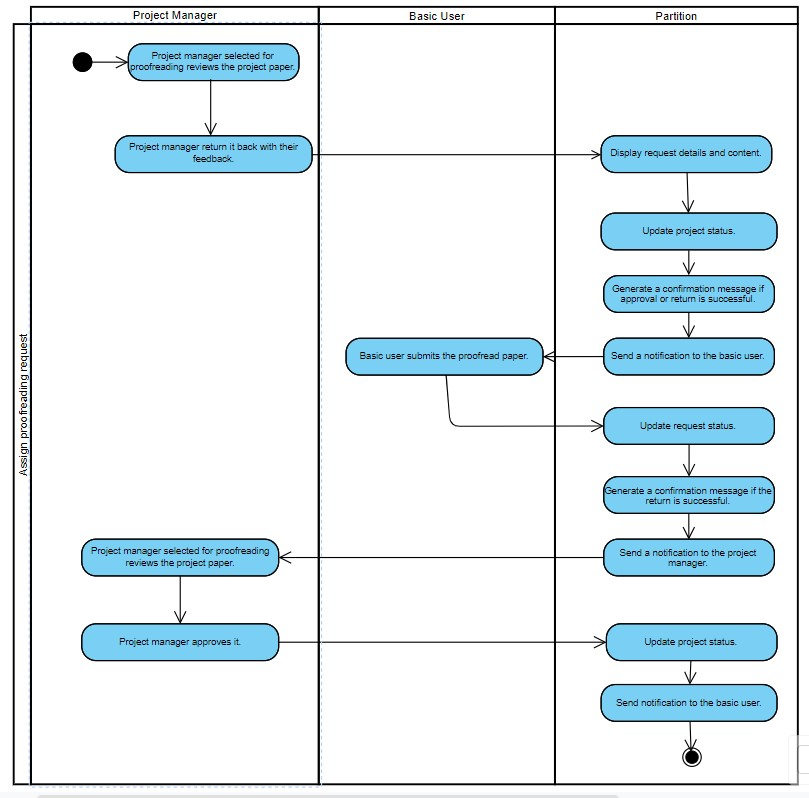


Figure 16 Activity Diagram Managing Proofreading Request Assign

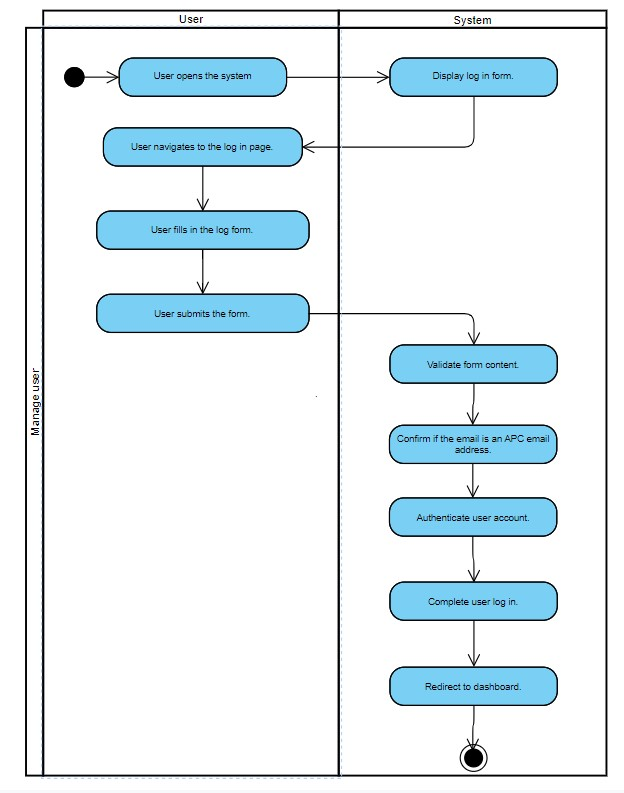


Figure 17 Activity Diagram Access System

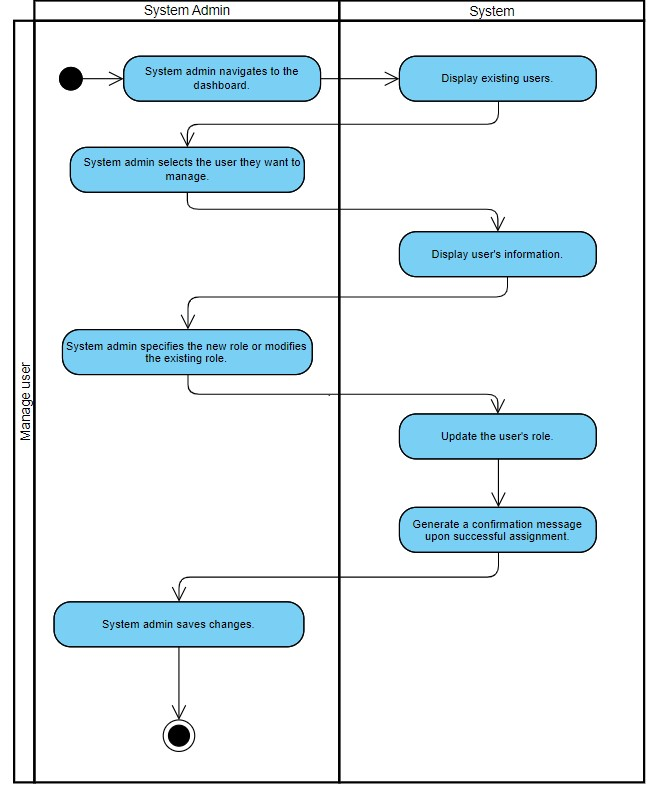


Figure 18 Activity Diagram Manage User

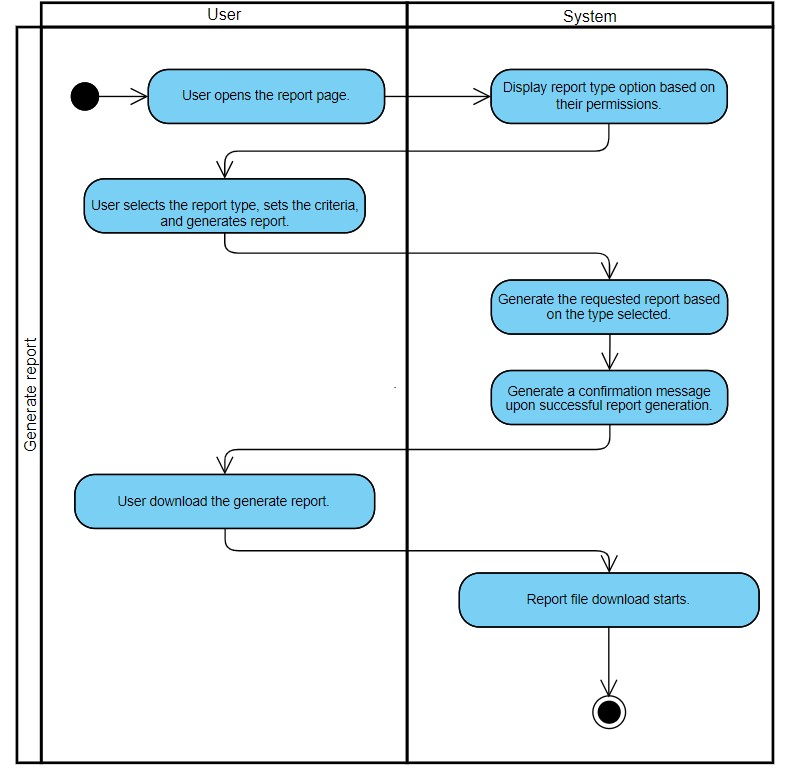


Figure 19 Activity Diagram Generate Report

### Class Diagram

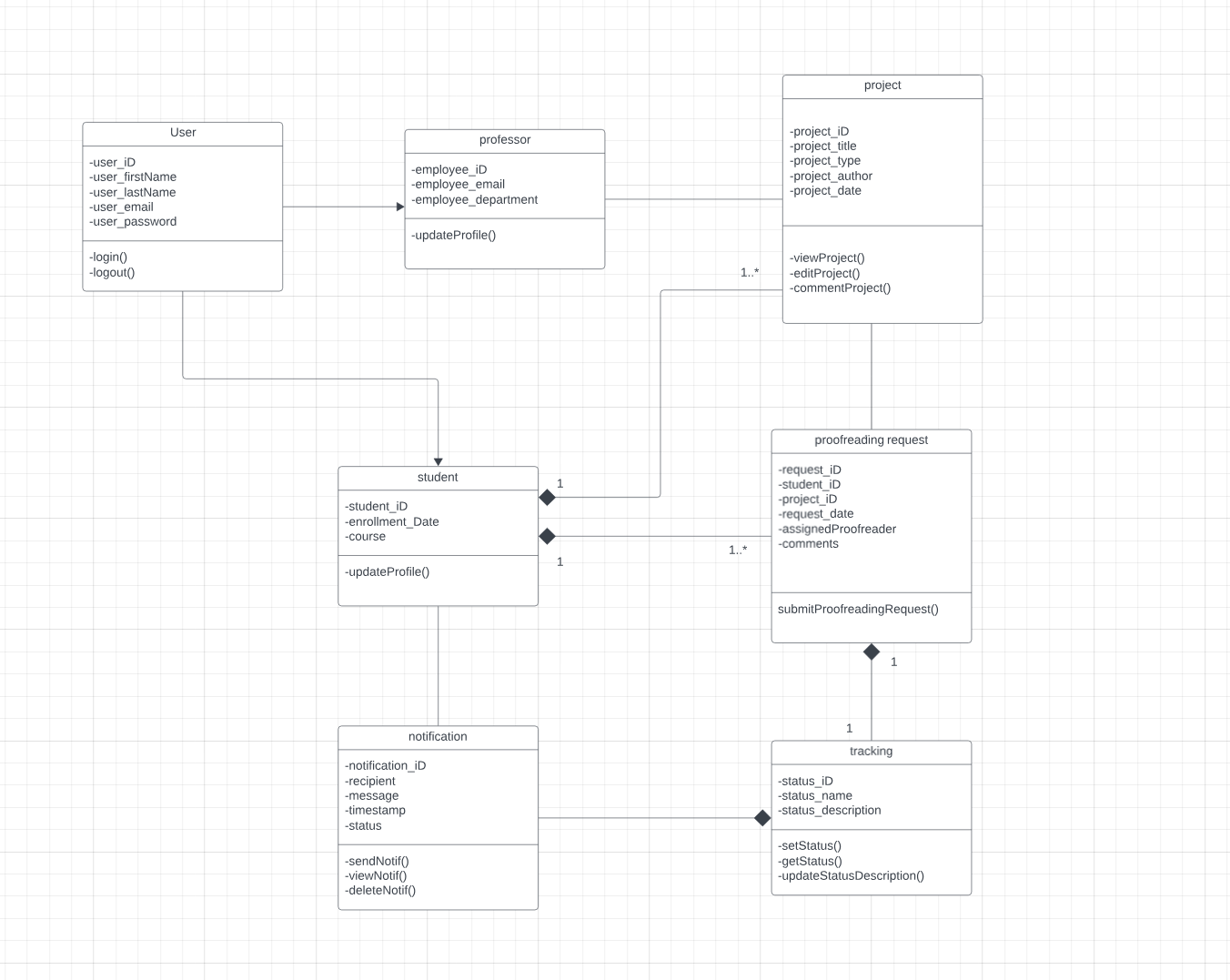


Figure 20 Class Diagram

### Sequence Diagrams

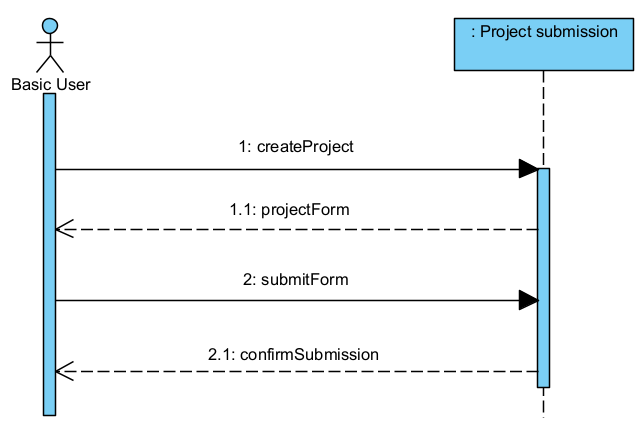


Figure 21 System Sequence Diagram Manage Project Create

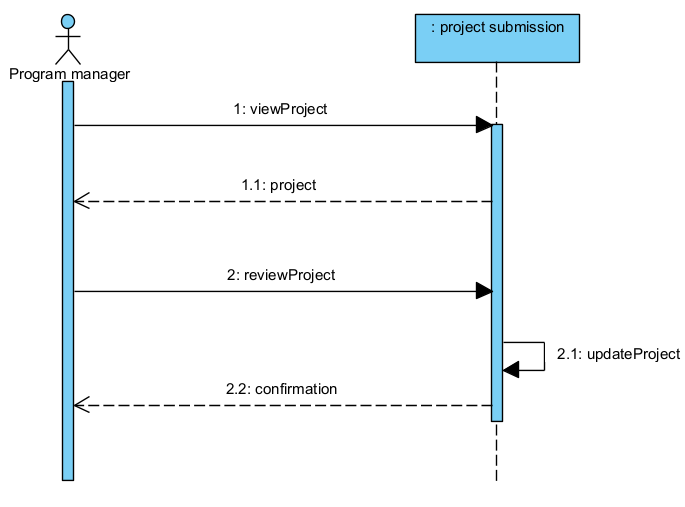


Figure 22 System Sequence Diagram Manage Project Approve

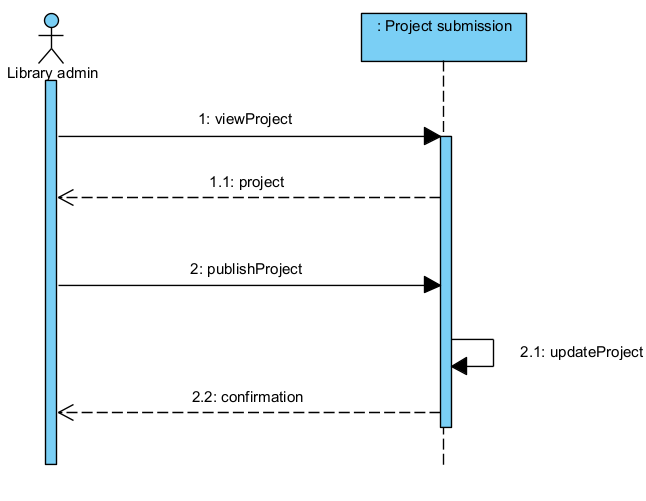


Figure 23 System Sequence Diagram Manage Project Publish

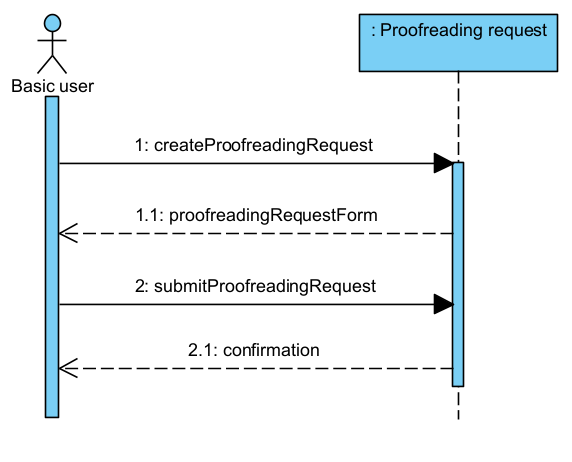
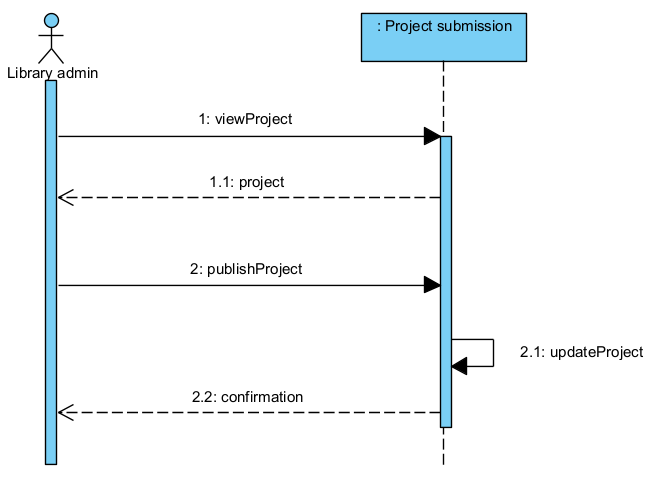


Figure 24 System Sequence Diagram Manage Proofreading Request Create

A diagram of a project manager

Description automatically generated

Figure 25 System Sequence Diagram Manage Proofreading Request Approve



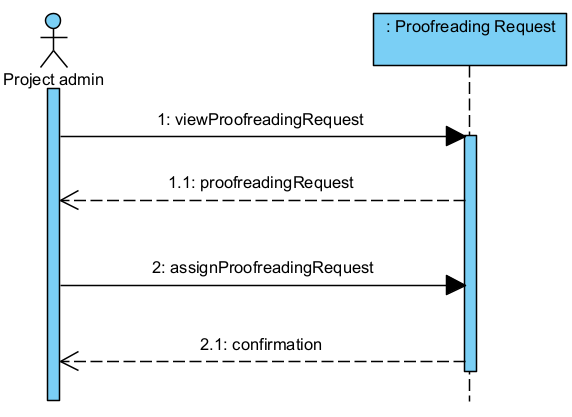


Figure 26 System Sequence Diagram Manage Proofreading Request Assign

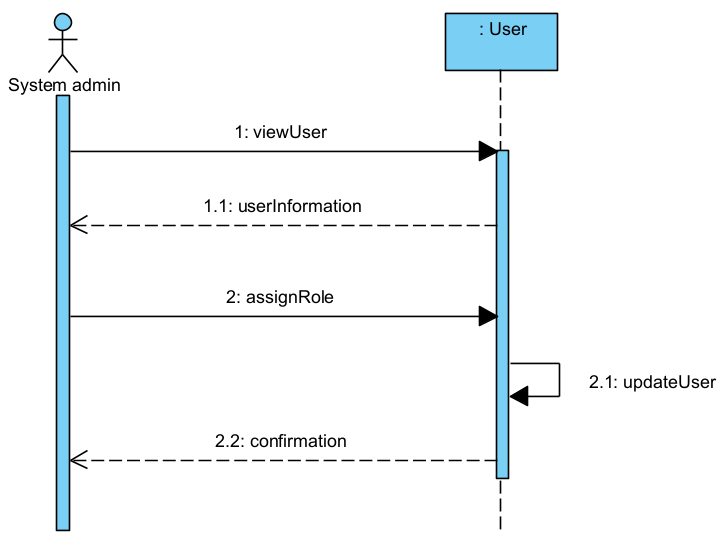


Figure 27 System Sequence Diagram Manage User

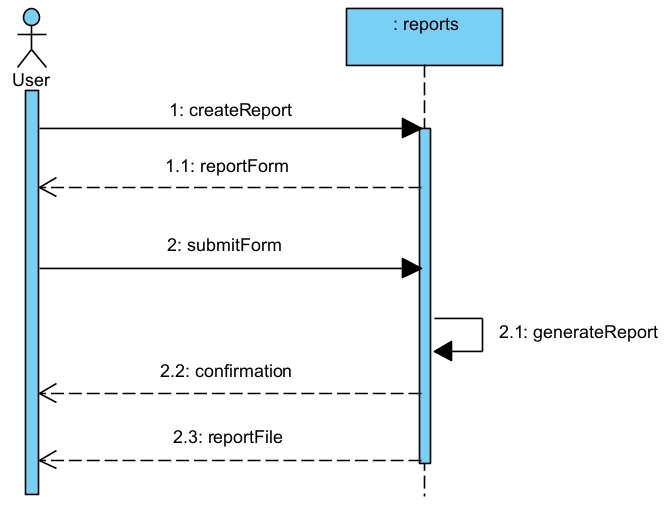


Figure 28 System Sequence Diagram Generate Report

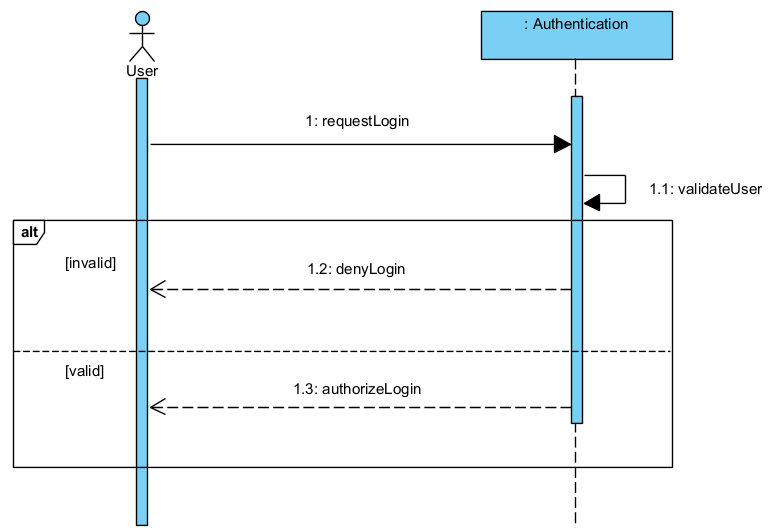


Figure 29 System Sequence Diagram Access System

State Machine Diagrams

A diagram of a company

Description automatically generated

Figure 30 State Machine Diagram Proofreading Approval

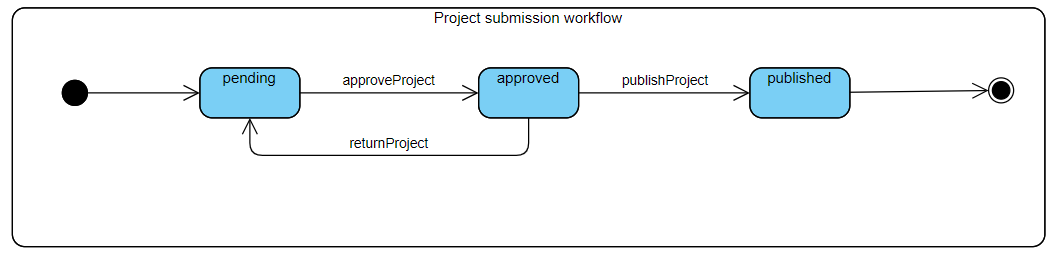
\

Figure 31 State Machine Diagram Project Approval

### Package Diagram

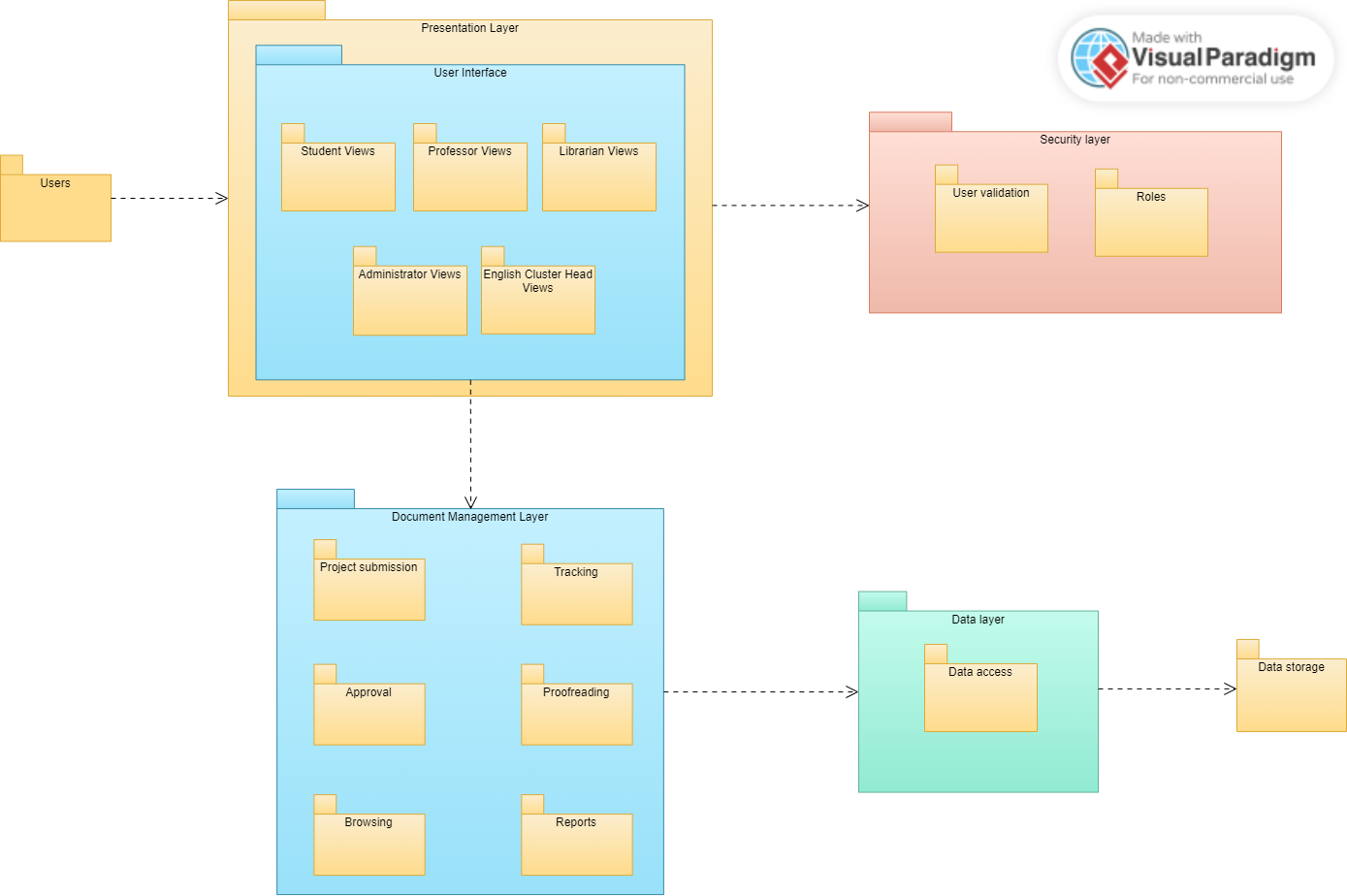


Figure 32 Package diagram

### Component Diagram

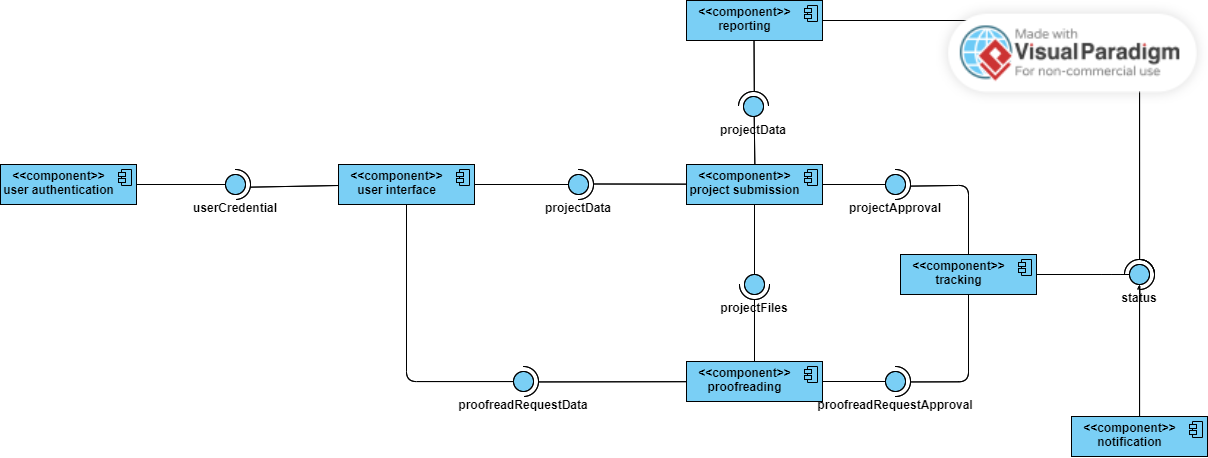


Figure 33 Component Diagram

### Deployment Diagram

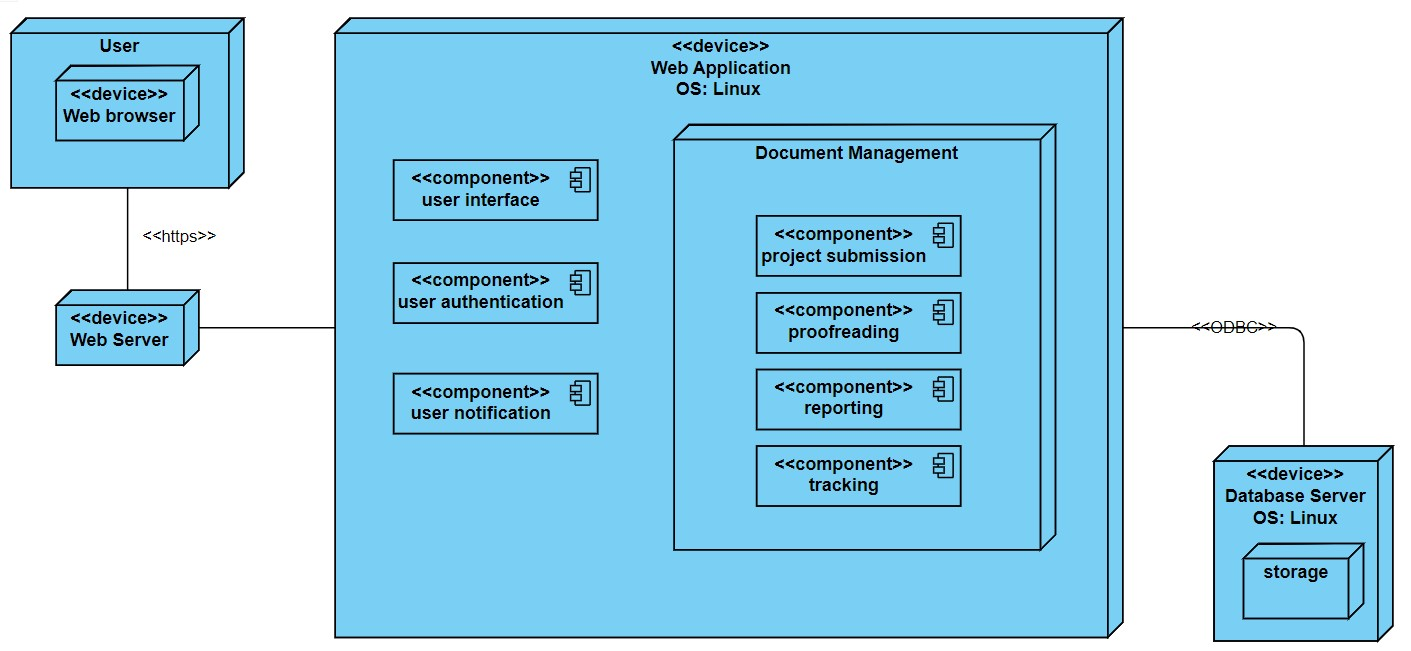


Figure 34 Deployment Diagram

# Conclusion

For this subject, the team has created various diagrams in the system analysis and design. The requirements analysis has also been further defined compared to what was shown in the midterm presentation. The team managed to conceptualize a new system that will make submitting project files more convenient than ever before, as the team manages to improve better diagrams that will be easier to read and comprehend the system’s capabilities, as well as ensuring all diagrams must be connected to avoid any misunderstandings. For the next part of PBL, the plan is to begin the development of Ramkolek. The goal is for each sprint, an MVP will be created to demonstrate the system’s functionality to the client, and once the client has been satisfied with the functions it can finally be deployed to the public.

# References

|  |  |
| --- | --- |
| [1] | S. Siripipatthanaku, P. Kaewpuang, T. Sitthipon and P. Sriboonruang, "A Review of Educational Adaptation During the COVID-19 Pandemicvia Online Learning," *International Journalof Computing Sciences Research,* vol. 7, pp. 1514-1530, 2022. |
| [2] | "About APC," Asia Pacific College, 2019. [Online]. Available: https://apc.edu.ph/about-apc/. [Accessed 27 May 2023]. |
| [3] | O. D. P. Srirahayu, "Open data repositories: Current risks and opportunities," *College & Research Library News,* vol. 81, 2020. |
| [4] | I. Balu N, "Introduction Microsoft Teams," in *Introducing Microsoft Teams*, SpringerLink, 2018, pp. 1-42. |
| [5] | J. Poston, S. Apostel and K. Richardson, "Using Microsoft Teams to Enhance Engagement and Learning with Any Class: It’s Fun and Easy," 2019. |
| [6] | S. Mijailović, D. Simonović and D. Đokić, "Some examples of using One Drive in establishing more interactive cooperation of," 2016. |
| [7] | R. Mercurio and B. Merrill, "OneDrive," in *Beginning Microsoft 365 Collaboration Apps*, Apress, 2021. |
| [8] | R. S. F. W. a. J. L. Haiyang Wang, On the Impact of Virtualization on Dropbox-like Cloud File Storage/Synchronization Services, IEEE, 2012. |
| [9] | Github, "Github Docs," [Online]. Available: https://docs.github.com/en/get-started/quickstart/hello-world. [Accessed 8 November 2023]. |
| [10] | M.-A. S. A. Z. Joseph Feliciano, "Student Experiences Using GitHub in Software Engineering Courses: A Case Study," 2016. |
| [11] | Folderit, "Document Management System Folderit," [Online]. Available: https://www.folderit.com/. [Accessed 8 November 2023]. |
| [12] | "Download Microsoft Teams Desktop and Mobile Apps | Microsoft Teams," Microsoft, 2023. [Online]. Available: https://www.microsoft.com/en-us/microsoft-teams/download-app. [Accessed 9 June 2023]. |
| [13] | "Official Website of Koha Library Software," Koha Community, 2023. [Online]. Available: https://koha-community.org/. [Accessed 7 June 2023]. |
| [14] | F. Richter, "Big Three Dominate the Global Cloud Market," Statista, 28 April 2023. [Online]. Available: https://www.statista.com/chart/18819/worldwide-market-share-of-leading-cloud-infrastructure-service-providers/. [Accessed 6 June 2023]. |
| [15] | L. S. Vailshery, "Global use of cloud providers by organizations 2022, by vendor," Statista, 12 September 2022. [Online]. Available: https://www.statista.com/statistics/1224552/organization-use-cloud-provider-global/. [Accessed 6 June 2023]. |
| [16] | "Welcome to CodeIgniter4," CodeIgniter Foundation, 21 May 2023. [Online]. Available: https://www.codeigniter.com/user\_guide/intro/index.html. [Accessed 7 June 2023]. |
| [17] | P. K. T. S. a. P. S. S. Siripipatthanaku, "A Review of Educational Adaptation During the COVID-19 Pandemicvia Online Learning," *International Journalof Computing Sciences Research,* vol. 7, pp. 1514-1530, 2022. |
| [18] | S. A. G. S. G. a. N. T. N. S. Bashir, "Evolution of institutional repositories: Managing institutional research output to remove the gap of academic elitism," *Journal of Librarianship and Information Science,* vol. 54, no. 3, pp. 518-531, 2021. |
| [19] | M. S. a. A. Sobhan, "Authentication and Authorization: Security Issues for Institutional Digital Repositories," *ResearchGate,* 2010. |
| [20] | D. S. a. D. Đ. S. Mijailović, Some examples of using One Drive in establishing more interactive cooperation of, 2016. |
| [21] | C. Song, "Scalable Systems 101," educative, 31 May 2022. [Online]. Available: https://www.educative.io/blog/scalable-systems-101#scalability. [Accessed 14 February 2023]. |
| [22] | A. M. Satina, "Design and Development of an Online Repository System for Thesis and Special Problem Manuscripts," *IJODeL,* vol. 3, no. 1, 2017. |
| [23] | S. Bashir, S. Bashir, S. A. Ganaie, S. Gul and N. T. Nisa, "Evolution of institutional repositories: Managing institutional research output to remove the gap of academic elitism," *Journal of Librarianship and Information Science,* vol. 54, no. 3, pp. 518-531, 2021. |
| [24] | O. Altunyan, "Collaborate on research papers with GitHub," Dev.to, 5 May 2021. [Online]. Available: https://dev.to/azure/collaborate-on-research-papers-with-github-76e. [Accessed 10 June 2023]. |

# Appendices

## Appendix A: Product Roadmap

*Updated product roadmap file here. Make sure that content is readable.*

|  |  |  |
| --- | --- | --- |
| SNTSDEV | SSYADD1 | SCSPROJ |
| Research  Project Proposal  Product Vision  User Stories  Project Roadmap | User Case  Event Tables  GAP Analysis  Context Diagram  Data Flow Diagrams  Entity-Relationship Diagrams  Activity Diagrams  Object Diagrams  Class Diagrams  Sequence Diagrams  Package Diagram  Component Diagram  Deployment Diagram  Prototyping | Sprint 1 (Weeks 1 – 4)   * Login/Logout * Final Submission Form * Proofreading Request Form * Dashboard   Sprint 2 (Weeks 5 – 8)   * Notifications * Roles * Approve/Return   Sprint 3 (Weeks 9 – 13)   * Homepage * Search & Filter * Report Generation |

## Appendix B: Minutes of the Meetings

