
Software Requirements Specification

for

Research Publication Management System

Version 1.0 approved

Prepared by: MUHAMMAD ZAFAR(21I1534) ,ALYAN
SHAHID(21I1537), AHMED AWAIZ(21I1539)

FAST National University of Computer and Emerging Sciences,
Islamabad Campus

12 April 2024

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction.....	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions.....	1
1.4 Product Scope.....	2
1.5 References	3
2. Overall Description	4
2.1 Product Perspective	4
2.2 Product Functions.....	5
2.3 User Classes and Characteristics	6
2.4 Operating Environment	6
2.5 Design and Implementation Constraints.....	7
2.6 User Documentation	10
2.7 Assumptions and Dependencies	10
3. External Interface Requirements	11
3.1 User Interfaces	12
3.2 Hardware Interfaces.....	12
3.3 Software Interfaces	13
3.4 Communications Interfaces	14
4. System Features	17
4.1 System Feature 1	Error! Bookmark not defined.
4.2 System Feature 2 (and so on)	Error! Bookmark not defined.
5. Other Nonfunctional Requirements.....	18
5.1 Performance Requirements.....	18
5.2 Safety Requirements.....	18
5.3 Security Requirements.....	18
5.4 Software Quality Attributes.....	19
5.5 Business Rules.....	19
6. Diagrams.....	19
6.1 Use Case Diagram	19
6.2 Activity Diagram	21
6.3 Sequence Diagram.....	25
Appendix A: Glossary.....	31
Appendix B: Analysis Models.....	31
Appendix C: To Be Determined List.....	32

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to outline the software requirements for the Research Publication Management System, hereinafter referred to as the RPM System. This SRS document identifies the product to be developed and provides a detailed description of its scope.

The RPM System aims to facilitate the management of research publications within a university or research institution. It will provide researchers with a platform to submit their publications, which will then undergo review, categorization, and subsequent accessibility to other researchers within the institution. The system is designed to streamline the publication management process, enhance collaboration among researchers, and improve access to scholarly work.

This document serves as a comprehensive guide for the development team, stakeholders, and end-users, detailing the functional and non-functional requirements of the RPM System. It outlines the features, functionalities, and constraints of the system to ensure that it meets the needs and expectations of its users.

1.2 Document Conventions

This SRS document follows the following conventions to ensure clarity and consistency in presenting the software requirements for the Research Publication Management System:

1. Font Styles/Line Spacing:

- Bold: Indicates section headings and titles.
- Line Spacing: 1.15
- Text Style: Times New Roman/Arial
- Headings Size: 18
- Subheadings Size: 14
- Text Size: 11

2. Priority Levels:

- Each requirement statement is assigned a priority level to indicate its importance in the development process.
- Priority levels are represented using the following scale:
 - Must-have (MUST): Essential requirements that must be implemented for the system to be considered functional.
 - Should-have (SHOULD): Important requirements that are highly desirable but not critical for the initial release.

- Could-have (COULD): Optional requirements that may enhance the system's functionality if time and resources permit.
- Won't-have (WON'T): Requirements that are explicitly excluded from the current scope of the project.

3. Numbering Scheme:

- Requirements and subsections are numbered hierarchically to facilitate easy reference and navigation within the document.

4. Terminology:

- Consistent terminology is used throughout the document to avoid ambiguity and ensure clarity.

5. Acronyms and Abbreviations:

- Acronyms and abbreviations are spelled out upon their first occurrence in the document, followed by the acronym or abbreviation in parentheses. Subsequently, the acronym or abbreviation may be used.

These conventions are adopted to maintain uniformity and readability across the SRS document, enabling effective communication among stakeholders, developers, and end-users.

1.3 Intended Audience and Reading Suggestions

Intended Audience:

The SRS document is intended for a diverse audience involved in the development, management, and usage of the Research Publication Management System (RPM System). This includes but is not limited to:

- 1. Developers:** Individuals responsible for designing, implementing, and testing the RPM System.
- 2. Project Managers:** Leaders overseeing the development process, resource allocation, and project milestones.
- 3. Quality Assurance/Testers:** Personnel responsible for verifying and validating the functionality and performance of the RPM System.
- 4. Stakeholders:** Individuals or groups with a vested interest in the success of the RPM System, such as university administrators, researchers, and funding agencies.
- 5. Documentation Writers:** Writers tasked with producing user manuals, technical guides, and other documentation related to the RPM System.

Reading Suggestions:

The SRS document is organized to provide comprehensive information about the requirements and scope of the RPM System. Readers are encouraged to follow the suggested sequence for optimal understanding:

1. Overview Sections: Begin by reading the introductory sections (1. Introduction) to gain a high-level understanding of the purpose and scope of the RPM System.

2. Functional Requirements: Developers and project managers should focus on section 2 (Functional Requirements) to understand the specific features and functionalities expected from the RPM System.

3. Non-Functional Requirements: Testers and quality assurance personnel should review section 3 (Non-Functional Requirements) to identify performance, security, and usability constraints.

4. System Models and Use Cases: Developers and testers may refer to section 4 (System Models and Use Cases) for detailed diagrams and scenarios depicting system behavior and interactions.

5. External Interface Requirements: Developers and stakeholders should pay attention to section 5 (External Interface Requirements) to understand integration points with external systems and services.

By following this suggested reading sequence, readers can gain a comprehensive understanding of the RPM System's requirements and make informed decisions throughout the development lifecycle.

1.4 Product Scope

The Research Publication Management System (RPM System) is a software solution designed to streamline the management of research publications within a university or research institution. The RPM System provides researchers with a centralized platform to submit their publications, which undergo review, categorization, and subsequent accessibility to other researchers within the institution.

Purpose:

The primary purpose of the RPM System is to enhance collaboration and knowledge sharing among researchers by facilitating the submission, review, and dissemination of research publications. By centralizing the publication management process, the system aims to improve efficiency, visibility, and accessibility of scholarly work within the institution.

Benefits, Objectives, and Goals:

- **Efficient Publication Management:** The RPM System automates and simplifies the process of submitting, reviewing, and categorizing research publications, saving time and effort for researchers and administrators.
- **Enhanced Collaboration:** By providing a centralized platform for publication submission and access, the system fosters collaboration and knowledge exchange among researchers within the institution.
- **Improved Visibility:** Researchers can easily search and access a wide range of research publications, increasing the visibility of their work and promoting academic recognition.
- **Compliance and Governance:** The system ensures compliance with institutional policies and standards for publication management, promoting transparency and accountability.

- Integration with Academic Databases: Integration with academic databases and repositories enhances the discoverability and reach of research publications, further advancing the institution's reputation and impact in the academic community.

Relation to Corporate Goals/Business Strategies:

The RPM System aligns with the corporate goals and business strategies of the university or research institution by:

- Enhancing the institution's research capabilities and reputation through improved publication management and collaboration.
- Supporting the institution's mission to advance knowledge and innovation by providing researchers with tools to effectively share and disseminate their research findings.
- Promoting efficiency and productivity by streamlining the publication management process and reducing administrative burden on researchers and administrators.

1.5 References

- https://www.researchgate.net/figure/Example-of-an-research-data-management-systems-RDMS-in-research-institutions_fig1_345133344

2. Overall Description

2.1 Product Perspective

The Research Publication Management System (RPM System) is a standalone product developed as a new, self-contained solution to address the specific needs of managing research publications within a university or research institution. It is not part of an existing product family or a replacement for any existing systems but rather serves as an independent platform dedicated to publication management.

Context and Origin:

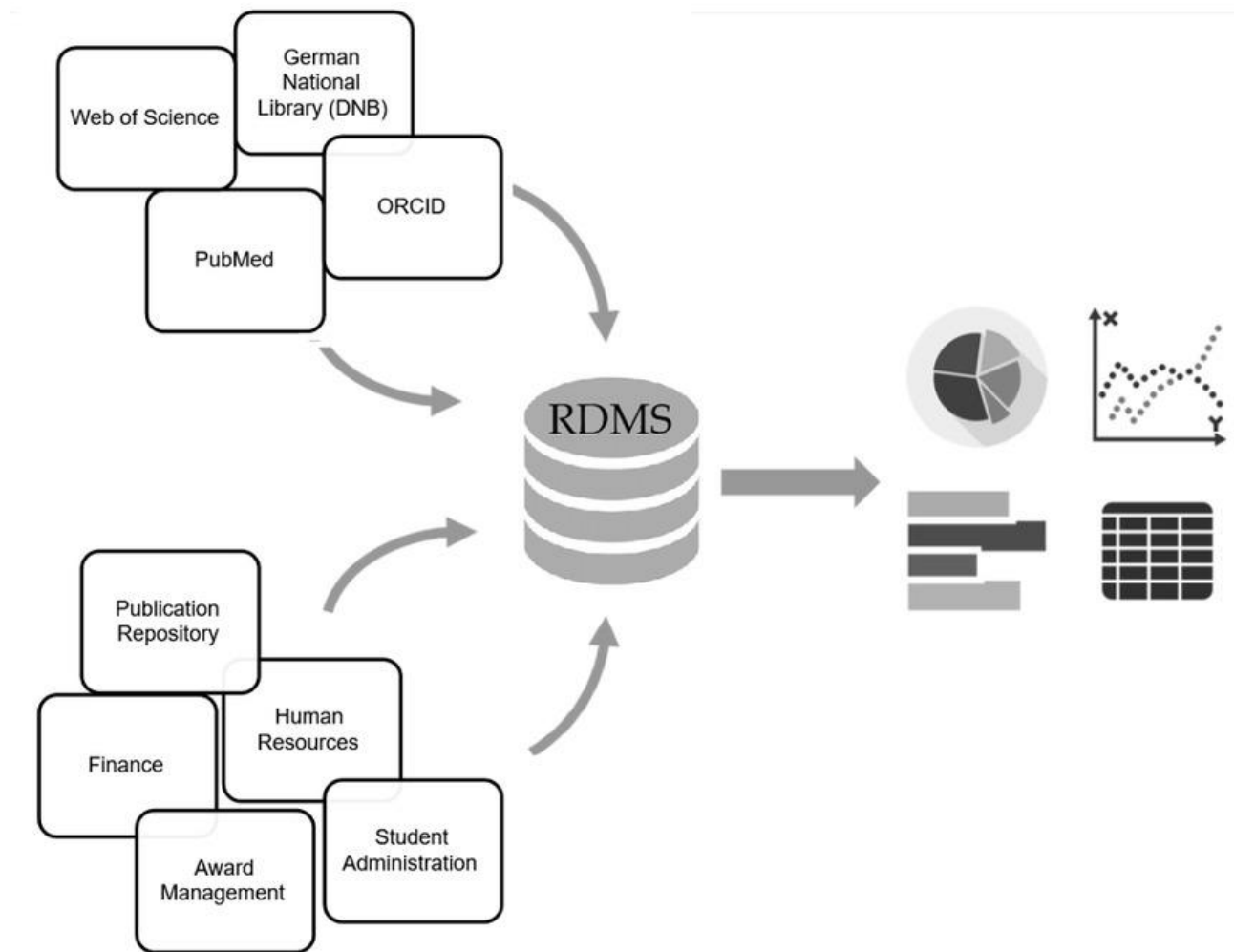
The RPM System has its origins in the growing demand for efficient and centralized tools to manage the increasing volume of research publications generated within academic and research environments. Traditional methods of publication management, such as manual submission and review processes, have become cumbersome and time-consuming. The RPM System emerges as a solution to streamline these processes, enhance collaboration among researchers, and improve the visibility and accessibility of scholarly work.

Relation to Larger System:

While the RPM System is designed as a self-contained product, it may interface with other systems or components within the larger institutional infrastructure. For example, it may integrate with existing academic databases, institutional repositories, or authentication systems to facilitate seamless access to research publications and ensure data consistency. However, the RPM System

itself operates independently and does not rely on any specific external systems for its core functionality.

DIAGRAM:



2.2 Product Functions

The Research Publication Management System (RPM System) is designed to perform the following major functions:

- **User Registration and Authentication:**

Allow users (researchers, administrators, etc.) to register accounts and authenticate themselves securely.

- **Publication Submission:**

Enable researchers to submit their research publications to the system, providing necessary metadata and files.

- **Publication Review and Approval:**

Facilitate the review and approval process for submitted publications by designated reviewers or administrators.

- **Publication Categorization and Tagging:**

Provide functionality for categorizing and tagging publications based on various criteria (e.g., topic, author, publication type).

- **Search and Discovery:**

Allow users to search and discover research publications based on specified criteria, such as keywords, authors, or publication dates.

- **Citation Tracking:**

Support tracking citations and references within publications, enabling users to navigate related works effectively.

- **Integration with Academic Databases:**

Integrate with external academic databases and repositories to enhance the discoverability and reach of research publications.

- **User Management and Access Control:**

Manage user roles and permissions to control access to system features and functionalities.

- **Reporting and Analytics:**

Provide reporting and analytics capabilities to track usage metrics, publication trends, and user activity.

- **System Administration:**

Offer administrative tools for managing system configuration, maintenance tasks, and user support.

2.3 User Classes and Characteristics

User Classes

1. Researchers:

- Frequency of Use: High, as they will be regularly submitting and accessing research publications.
- Subset of Product Functions Used: Mainly involved in publication submission, search and discovery, and citation tracking.
- Technical Expertise: Varied, ranging from novice to expert users.

- Security or Privilege Levels: Typically have standard user privileges, with access restricted to their own submissions and publications.
- Educational Level/Experience: Typically possess advanced degrees in their respective fields and have experience with academic publishing.

2. Administrators:

- Frequency of Use: Moderate to high, as they will be managing the system and overseeing publication processes.
- Subset of Product Functions Used: Involved in system management, publication review and approval, reporting and analytics, and system administration.
- Technical Expertise: Generally high, with knowledge of system administration and management.
- Security or Privilege Levels: Have elevated privileges to manage users, configure system settings, and perform administrative tasks.
- Educational Level/Experience: Typically possess advanced degrees and have experience in academic administration or IT management.

3. Reviewers (if applicable):

- Frequency of Use: Moderate, as they will review assigned publications periodically.
- Subset of Product Functions Used: Primarily involved in reviewing assigned publications and providing feedback.
- Technical Expertise: Moderate to high, depending on the field of expertise.
- Security or Privilege Levels: Have access to assigned publications for review purposes.
- Educational Level/Experience: Typically possess advanced degrees and have experience in peer review processes.

Distinguishing User Classes

- **Researchers** and **Administrators** are the most important user classes for this product, as they play key roles in submission, management, and oversight of research publications within the institution.
- **Reviewers** are less important but still necessary for the peer review process, if applicable.

2.4 Operating Environment

Hardware Platform

The Research Publication Management System (RPM System) is designed to operate on standard hardware commonly found in academic and research institutions. The hardware platform should meet the following minimum requirements:

Processor: Any modern multi-core processor capable of running the chosen operating system and database software efficiently.

Memory (RAM): At least 4 GB of RAM, with higher amounts recommended for better performance, especially when handling large datasets or concurrent user access.

Storage: Sufficient disk space for storing the application binaries, database files, and any uploaded publications. At least 100 GB of available disk space is recommended.

Network Connectivity: Reliable internet connectivity is required for accessing external academic databases, repositories, and for user authentication and communication.

Operating System and Versions

The RPM System is designed to be platform-independent and should be compatible with the following operating systems:

Linux: Any modern distribution such as Ubuntu, CentOS, or Debian, with recommended versions including Ubuntu 20.04 LTS or CentOS 8.

Windows: Windows 10 or later versions are supported.

macOS: macOS 10.15 (Catalina) or later versions are supported.

Software Components and Applications

The RPM System relies on several software components and applications to operate effectively. These include:

Web Server: The system requires web server software such as Apache HTTP Server or Nginx to serve the web-based user interface and handle HTTP requests.

Database Management System (DBMS): A relational database management system is required to store and manage publication metadata, user information, and system configurations. Recommended DBMS include MySQL, PostgreSQL, or SQLite.

Programming Languages: The system is primarily developed using programming languages such as Python for backend logic and JavaScript for frontend development. Ensure compatibility with the latest stable versions of Python (3.x) and JavaScript (ECMAScript 6).

Web Browser: Users should access the system using modern web browsers such as Google Chrome, Mozilla Firefox, Safari, or Microsoft Edge for optimal performance and compatibility.

Other Requirements

Security: The operating environment should adhere to standard security practices, including regular software updates, firewall configuration, and data encryption to protect sensitive information stored within the system.

Scalability: The system should be capable of scaling horizontally and vertically to accommodate increasing user loads and data volumes over time.

2.5 Design and Implementation Constraints

Corporate or Regulatory Policies

Data Privacy and Security Regulations: The system must comply with relevant data privacy and security regulations, such as GDPR (General Data Protection Regulation) or HIPAA (Health Insurance Portability and Accountability Act), to ensure the protection of sensitive user data.

Intellectual Property Policies: The system should respect intellectual property rights, including copyright and licensing agreements, when handling research publications and related materials.

Hardware Limitations

Resource Constraints: The system should be designed to operate within the specified hardware platform constraints, including memory, storage, and processing capabilities, to ensure optimal performance and scalability.

Interfaces to Other Applications

Integration Requirements: The system may need to interface with external academic databases, repositories, or authentication systems. Developers should ensure compatibility and seamless integration with these external applications.

Specific Technologies, Tools, and Databases

Programming Languages and Frameworks: The system must be developed using specific programming languages and frameworks, such as Python for backend development and JavaScript for frontend development. Additionally, adherence to coding standards and best practices is required to maintain code quality and readability.

Relational Database Management System (RDBMS): The system should utilize a specific RDBMS, such as MySQL or PostgreSQL, for storing and managing publication data.

Web Server Software: The system must be deployed on web server software, such as Apache HTTP Server or Nginx, to serve the web-based user interface and handle HTTP requests.

Security Considerations

Authentication and Authorization: The system should implement robust authentication and authorization mechanisms to control access to sensitive functionality and data.

Data Encryption: Transmission of sensitive information should be encrypted using secure protocols (e.g., HTTPS) to prevent unauthorized access or interception.

Vulnerability Management: Regular security assessments and vulnerability scans should be conducted to identify and mitigate potential security risks.

Design Conventions and Programming Standards

Coding Standards: Developers should adhere to established coding standards and programming conventions to ensure consistency and maintainability of the codebase.

Documentation Requirements: Comprehensive documentation should be provided for the system architecture, codebase, and APIs to facilitate future maintenance and enhancements.

Maintenance Responsibilities

Customer Organization Maintenance: The customer's organization may be responsible for maintaining the delivered software, including software updates, bug fixes, and user support. Clear guidelines and procedures should be established for ongoing maintenance and support activities.

2.6 User Documentation

User Documentation Components

The following user documentation components will be delivered along with the software:

User Manual:

A comprehensive user manual providing detailed instructions on how to use the Research Publication Management System (RPM System). It will cover topics such as user registration, publication submission, search functionality, citation tracking, and system administration tasks.

Online Help:

Interactive online help accessible within the RPM System interface. It will provide contextual guidance and assistance to users as they navigate through different features and functionalities of the system. This online help will be accessible via a help icon or a dedicated help section within the application.

Delivery Formats or Standards

The user documentation will be delivered in various formats to accommodate different user preferences and usage scenarios:

PDF Format: The user manual will be provided in PDF format for easy printing and offline reference.

Online Help System: The online help will be integrated into the RPM System interface and will be accessible through a web browser.

Known User Documentation Standards

The user documentation will adhere to industry best practices and standards for technical documentation, including:

Clarity and Conciseness: The documentation will be written in clear and concise language, avoiding technical jargon and ambiguous terminology.

User-Centric Approach: The documentation will focus on addressing the needs and expectations of the target user audience, providing relevant and actionable information.

Consistency and Formatting: Consistent formatting and styling will be applied throughout the documentation to enhance readability and user experience.

Accessibility: The documentation will be designed to be accessible to users with diverse backgrounds and abilities, ensuring compliance with accessibility standards (e.g., WCAG).

2.7 Assumptions and Dependencies

Assumed Factors

1. Availability of Third-Party Components:

- **Assumption:** It is assumed that third-party or commercial components, such as web server software (e.g., Apache HTTP Server) and database management systems (e.g., MySQL or PostgreSQL), will be available and compatible with the RPM System.
- **Implication:** If the availability or compatibility of these third-party components changes, it may impact the development and deployment of the RPM System, potentially requiring adjustments to the software architecture or dependencies.

2. Stable Operating Environment:

- **Assumption:** It is assumed that the operating environment (e.g., hardware platform, operating system) will remain stable and consistent throughout the development and deployment phases of the RPM System.
- **Implication:** Any changes or instability in the operating environment could affect the performance, reliability, and compatibility of the RPM System, potentially leading to delays or disruptions in the project timeline.

3. User Availability for Testing and Feedback:

- **Assumption:** It is assumed that users, including researchers, administrators, and reviewers, will be available and willing to participate in testing and providing feedback on the RPM System.
- **Implication:** If user availability is limited or if feedback is not provided in a timely manner, it may hinder the iterative development and refinement of the RPM System, potentially resulting in suboptimal user experience or functionality.

Dependencies

1. External APIs and Integration Points:

- **Dependency:** The RPM System depends on external APIs and integration points with academic databases, repositories, and authentication systems for features such as publication search, citation tracking, and user authentication.
- **Implication:** Any changes or disruptions to these external APIs or integration points may impact the functionality and performance of the RPM System, requiring adjustments or updates to maintain compatibility.

2. Software Development Frameworks and Libraries:

- **Dependency:** The RPM System relies on specific software development frameworks and libraries, such as Django for backend development and React.js for frontend development.
- **Implication:** Changes or updates to these frameworks and libraries may introduce compatibility issues or require modifications to the RPM System codebase, potentially affecting development timelines and deliverables.

3. External Interface Requirements

3.1 User Interfaces

The Research Publication Management System (RPM System) will include the following user interfaces to facilitate interaction between the software product and its users:

1. User Registration and Authentication Interface:

- **Characteristics:** This interface allows new users to register for an account and existing users to log in securely. It will include fields for entering user credentials (e.g., username, password) and options for password recovery or account management.
- **GUI Standards:** The user registration and authentication interface will adhere to standard security best practices for password hashing, encryption, and protection against common security threats such as SQL injection and cross-site scripting (XSS).

2. Publication Submission Interface:

- **Characteristics:** This interface enables researchers to submit their research publications to the RPM System. It will include forms for entering metadata (e.g., title, authors, abstract) and uploading publication files in various formats (e.g., PDF, DOCX).
- **GUI Standards:** The publication submission interface will follow standard web form design principles, including validation of user input, error handling, and feedback messages for successful submission.

3. Search and Discovery Interface:

- **Characteristics:** This interface allows users to search for and discover research publications within the RPM System. It will include search filters (e.g., keywords, authors, publication dates) and display search results in a user-friendly format.
- **GUI Standards:** The search and discovery interface will prioritize usability and accessibility, with intuitive search functionality and clear presentation of search results.

4. Publication Management Interface (For Administrators):

- **Characteristics:** This interface is accessible to administrators for managing publications, users, and system configurations. It will include options for reviewing submitted publications, assigning reviewers, and configuring system settings.
- **GUI Standards:** The publication management interface will incorporate role-based access control (RBAC) to enforce appropriate permissions and privileges for administrators.

3.2 Hardware Interfaces

The Research Publication Management System (RPM System) interacts with various hardware components of the system. Below are the logical and physical characteristics of each interface between the software product and the hardware components:

1. Server Hardware Interface:

Supported Device Types: The RPM System runs on server hardware infrastructure, including physical or virtual servers hosted in data centers or cloud environments.

- **Data and Control Interactions:** The software communicates with the server hardware to perform operations such as data storage, retrieval, processing, and network communication.
- **Communication Protocols:** Communication between the software and server hardware may occur via standard network protocols such as TCP/IP, HTTP, and HTTPS.

2. Client Devices Interface:

- **Supported Device Types:** Users access the RPM System from various client devices, including desktop computers, laptops, tablets, and smartphones.
- **Data and Control Interactions:** Client devices interact with the software to send requests, receive responses, and display information through web browsers or dedicated client applications.
- **Communication Protocols:** Communication between client devices and the software occurs over standard web protocols such as HTTP and HTTPS for web-based interfaces.

3. Storage Hardware Interface:

- **Supported Device Types:** The RPM System stores data on storage hardware devices, including hard disk drives (HDDs), solid-state drives (SSDs), and network-attached storage (NAS) systems.
- **Data and Control Interactions:** The software interacts with storage hardware to store and retrieve data files, documents, and metadata related to research publications.
- **Communication Protocols:** Communication between the software and storage hardware may utilize file system protocols (e.g., NFS, SMB) or database communication protocols (e.g., SQL).

4. Networking Hardware Interface:

- **Supported Device Types:** Networking hardware components such as routers, switches, firewalls, and load balancers are used to facilitate communication between the RPM System and client devices.
- **Data and Control Interactions:** The software communicates with networking hardware to establish network connections, route data packets, enforce security policies, and ensure high availability and scalability.

- **Communication Protocols:** Communication between the software and networking hardware occurs over standard network protocols such as TCP/IP, Ethernet, and IPsec.

Note:

The RPM System is designed to be hardware-agnostic, allowing it to run on a wide range of hardware configurations and environments.

3.3 Software Interfaces

The Research Publication Management System (RPM System) interacts with various software components, including databases, operating systems, tools, libraries, and integrated commercial components. Below are the connections between the RPM System and other specific software components:

1. Database Management System (DBMS):

- **Name and Version:** PostgreSQL 12.0
- **Data Items/Messages:**
 - Incoming:** User registration/authentication data, publication metadata, submission records, user preferences/settings.
 - Outgoing:** Search results, publication details, user authentication responses, system configuration data.
- **Purpose:** The RPM System uses PostgreSQL as the primary database to store and manage various data related to research publications, user accounts, and system configurations. It facilitates data retrieval, storage, and manipulation operations.
- **Services Needed:** CRUD (Create, Read, Update, Delete) operations for managing database records, transaction management, query execution, and indexing.
- **Nature of Communications:** The RPM System communicates with PostgreSQL using SQL queries over TCP/IP connections.

2. Web Server Software:

- **Name and Version:** Apache HTTP Server 2.4.41
- **Data Items/Messages:**
 - Incoming:** HTTP requests from client devices, including requests for web pages, API endpoints, and static resources (e.g., images, CSS files).
 - Outgoing:** HTTP responses containing requested resources, including HTML content, JSON data, and file downloads.
- **Purpose:** Apache HTTP Server serves as the web server for the RPM System, handling incoming HTTP requests from client devices and delivering appropriate responses. It provides the infrastructure for hosting the web-based user interface and supporting web services.
- **Services Needed:** HTTP request processing, request routing, request/response handling, content negotiation, caching, and logging.

- **Nature of Communications:** The RPM System communicates with Apache HTTP Server using the HTTP protocol over TCP/IP connections.

3. Programming Languages and Frameworks:

- **Python:** Version 3.8.5
- **Django:** Version 3.1.2
- **JavaScript/React.js:** ECMAScript 6
- **Data Items/Messages:** Business logic, user interface components, and application state data.
- **Purpose:** Python with Django is used for backend development, including data processing, authentication, and API development. JavaScript with React.js is used for frontend development, providing interactive user interfaces and dynamic content rendering.
- **Services Needed:** Routing, request handling, middleware, data serialization/deserialization, template rendering, and state management.
- **Nature of Communications:** The RPM System communicates internally between backend and frontend components using function calls, RESTful API requests/responses, and WebSocket connections for real-time updates.

3.4 Communications Interfaces

The Research Publication Management System (RPM System) requires various communications functions to facilitate interaction with users, external systems, and network services. Below are the requirements associated with communications functions required by the RPM System:

1. Web Browser Communication:

- **Requirements:** The RPM System interacts with web browsers used by users to access the web-based user interface. It utilizes HTTP and HTTPS protocols for communication between the client (web browser) and the server (RPM System).
- **Message Formatting:** HTTP request and response messages adhere to standard HTTP protocol formats, including headers and body content in appropriate MIME types (e.g., JSON, HTML).
- **Communication Standards:** HTTP/1.1 and HTTP/2 protocols are supported for web browser communication.
- **Communication Security:** Communication over HTTPS is enforced to ensure data confidentiality and integrity. SSL/TLS encryption is used to secure HTTP communication channels.
- **Data Transfer Rates:** Data transfer rates depend on network conditions and bandwidth availability, with optimizations for efficient data transmission (e.g., compression, caching).

- **Synchronization Mechanisms:** Asynchronous communication is supported for non-blocking operations, while synchronous communication is used for request-response interactions.

2. Email Communication:

- **Requirements:** The RPM System may send email notifications, alerts, and communication messages to users and administrators. It utilizes SMTP (Simple Mail Transfer Protocol) for outgoing email communication.
- **Message Formatting:** Email messages adhere to standard email formatting conventions, including subject, body, sender, recipient, and optional attachments.
- **Communication Standards:** SMTP protocol is used for sending outgoing emails. POP3 or IMAP protocols may be used for receiving incoming emails if email integration features are implemented.
- **Communication Security:** SMTP communication may be secured using STARTTLS or SSL/TLS encryption to protect email content and transmission.
- **Data Transfer Rates:** Email delivery rates depend on email server configuration, network latency, and recipient email server responsiveness.
- **Synchronization Mechanisms:** Email communication is typically asynchronous, with email delivery and processing occurring independently of user interactions within the RPM System.

3. Network Server Communication:

- **Requirements:** The RPM System may communicate with external network servers, APIs, or services for functionalities such as integration with academic databases, repositories, or authentication systems.
- **Message Formatting:** API requests and responses adhere to defined API protocols and data formats (e.g., JSON, XML).
- **Communication Standards:** RESTful APIs over HTTP/HTTPS are commonly used for network server communication. Other protocols such as FTP, LDAP, or OAuth may be used for specific integrations.
- **Communication Security:** Communication with external network servers is secured using HTTPS with SSL/TLS encryption to protect data in transit.
- **Data Transfer Rates:** Data transfer rates depend on network conditions, server load, and API response times.
- **Synchronization Mechanisms:** Synchronization mechanisms, such as polling or webhook notifications, may be implemented to ensure timely data synchronization between the RPM System and external servers.

Note:

Communication interfaces and protocols will be implemented and configured according to industry best practices and standards for security, performance, and reliability.

4. System Features

4.1 User Registration and Authentication

4.1.1 Description and Priority

- *Allow users to register accounts and authenticate themselves securely.*
- **Priority:** High

4.1.2 Stimulus/Response Sequences

- **Stimulus:** User navigates to the registration page.
- **Response:** System presents a form for entering user credentials.
- **Stimulus:** User submits registration information.
- **Response:** System validates the information and creates a new user account if it meets the criteria.
- **Stimulus:** User attempts to log in.
- **Response:** System verifies the user's credentials and grants access if they are valid.

4.1.3 Functional Requirements

- **REQ-1:** The system shall provide a user registration form with fields for username, email, password, etc.
- **REQ-2:** The system shall validate user input during registration to ensure data integrity.
- **REQ-3:** Upon successful registration, the system shall send a confirmation email to the user's registered email address.
- **REQ-4:** The system shall provide a login form with fields for username and password.
- **REQ-5:** The system shall authenticate user credentials during login and grant access if they are valid.
- **REQ-6:** The system shall implement password hashing to securely store user passwords.
- **REQ-7:** The system shall support password recovery mechanisms, such as email verification or security questions.

4.2 Publication Submission

4.1.1 Description and Priority

- *Enable researchers to submit their research publications to the system, providing necessary metadata and files.*

- **Priority:** High

4.1.2 Stimulus/Response Sequences

- **Stimulus:** User navigates to the submission page.
- **Response:** System presents a form for entering publication metadata and uploading files.
- **Stimulus:** User submits publication information.
- **Response:** System validates the information and stores the publication in the database.

4.1.3 Functional Requirements

- **REQ-1:** The system shall provide a publication submission form with fields for title, authors, abstract, etc.
- **REQ-2:** The system shall support uploading publication files in various formats, such as PDF or DOCX.
- **REQ-3:** The system shall validate publication metadata and file uploads to ensure data integrity.
- **REQ-4:** The system shall assign a unique identifier to each submitted publication for tracking purposes.
- **REQ-5:** The system shall notify users upon successful submission of their publications.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- **Response Time:** The system shall respond to user interactions within 2 seconds for common operations such as publication search and submission.
- **Throughput:** The system shall support a minimum of 100 concurrent users without degradation in performance, measured as transactions per second.
- **Scalability:** The system architecture shall support horizontal scaling to accommodate a 50% increase in user base within 6 months of deployment.

5.2 Safety Requirements

- **Data Integrity:** The system shall implement measures to ensure the integrity of user data, preventing unauthorized access, modification, or deletion.
- **Backup and Recovery:** The system shall regularly backup all data and configurations, with a recovery time objective (RTO) of less than 4 hours in the event of a failure.
- **Audit Trail:** The system shall maintain an audit trail of user actions, including publication submissions, edits, and deletions, to ensure accountability and traceability.

5.3 Security Requirements

- **User Authentication:** The system shall require users to authenticate using strong passwords, with support for multi-factor authentication for administrative accounts.

- **Data Encryption:** All sensitive data transmitted over the network, including user credentials and publication content, shall be encrypted using industry-standard encryption algorithms.
- **Access Control:** The system shall enforce role-based access control (RBAC), limiting access to sensitive features and data based on user roles and permissions.

5.4 Software Quality Attributes

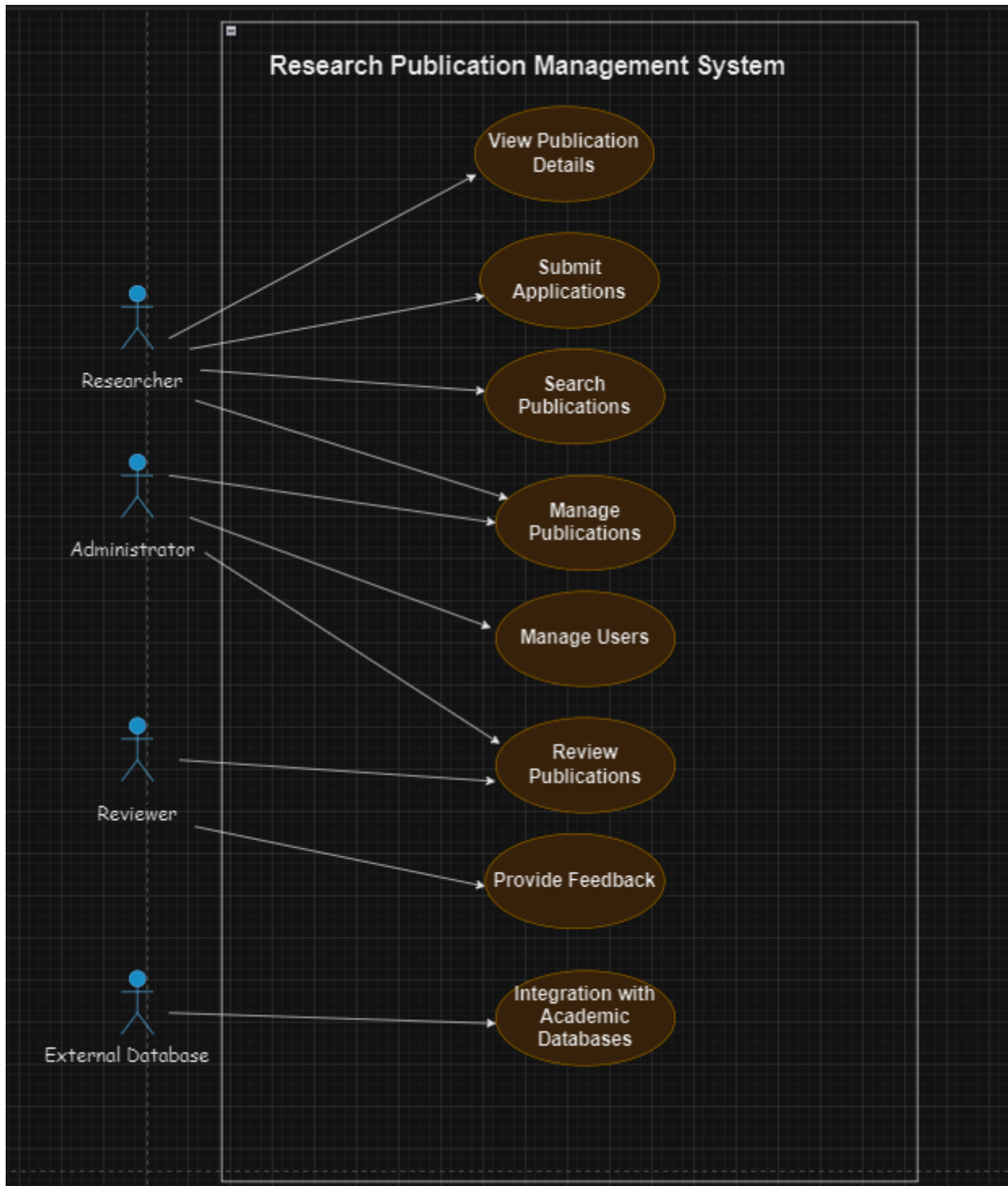
- **Maintainability:** The system shall adhere to modular design principles, allowing for easy maintenance and future enhancements without impacting core functionality.
- **Usability:** The system shall provide a user-friendly interface with intuitive navigation and clear instructions, targeting a learnability score of 80% for new users within the first hour of use.
- **Reliability:** The system shall achieve a mean time between failures (MTBF) of at least 30 days under normal operating conditions.

5.5 Business Rules

- **Publication Review:** Only designated administrators shall have the authority to approve or reject submitted publications, ensuring adherence to institutional publication guidelines.
- **User Registration:** Users must agree to the system's terms of service and privacy policy during registration, with an age restriction of 18 years or older for access.

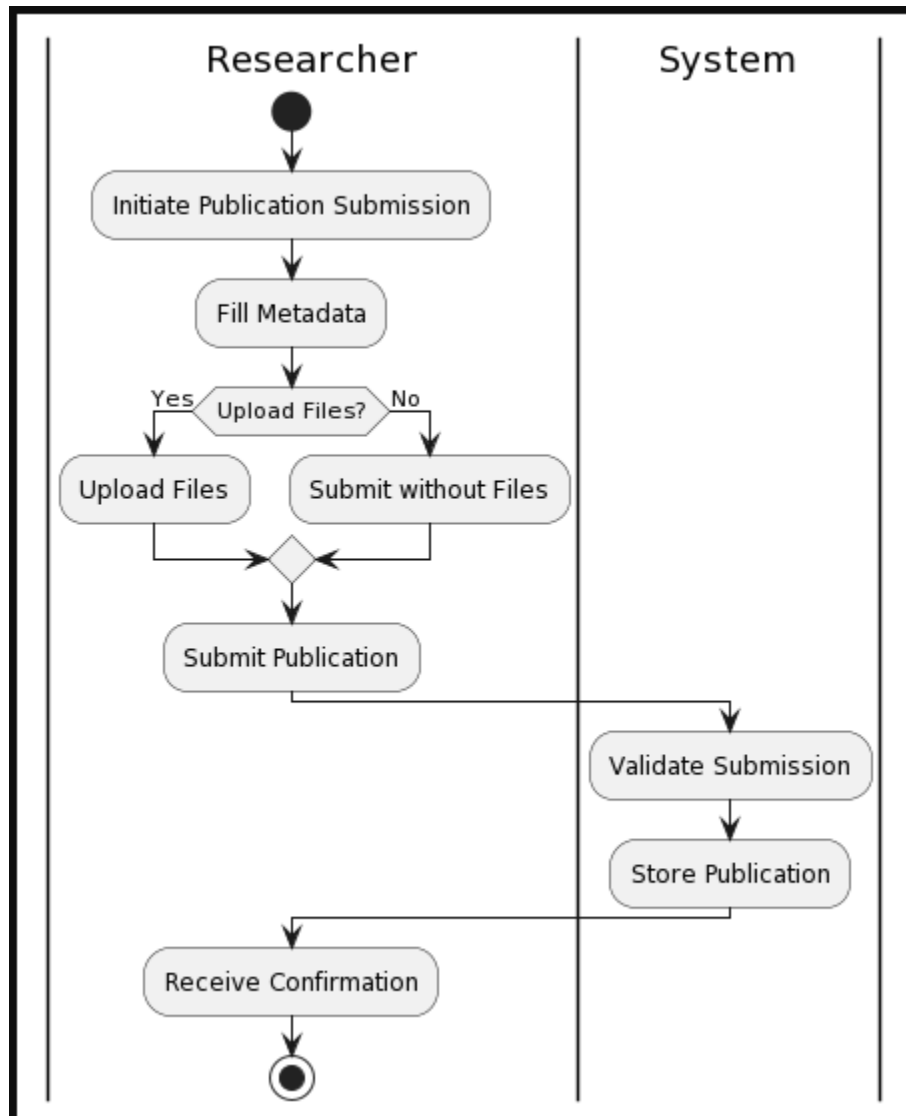
6. Diagrams

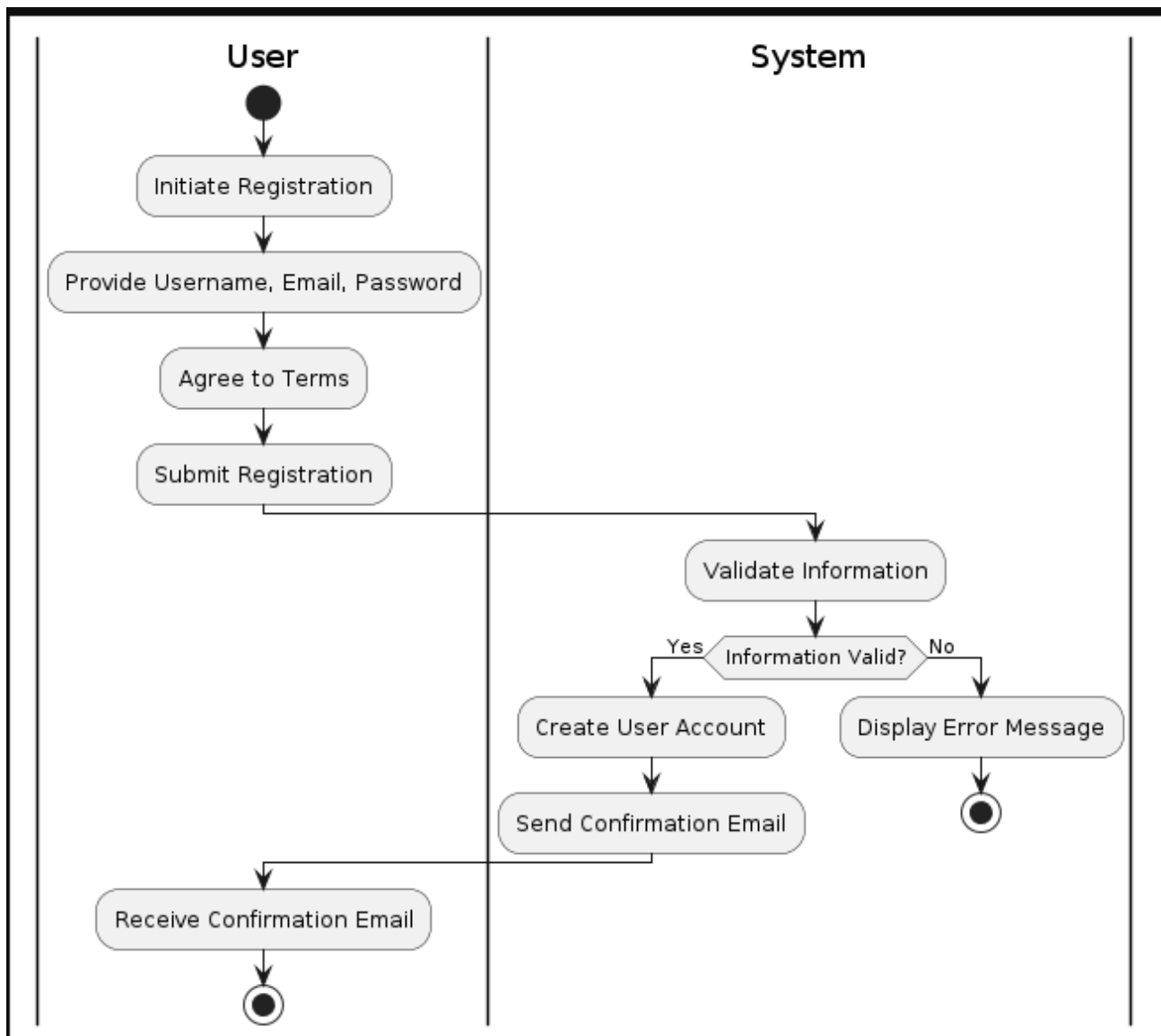
6.1 Use Case Diagram

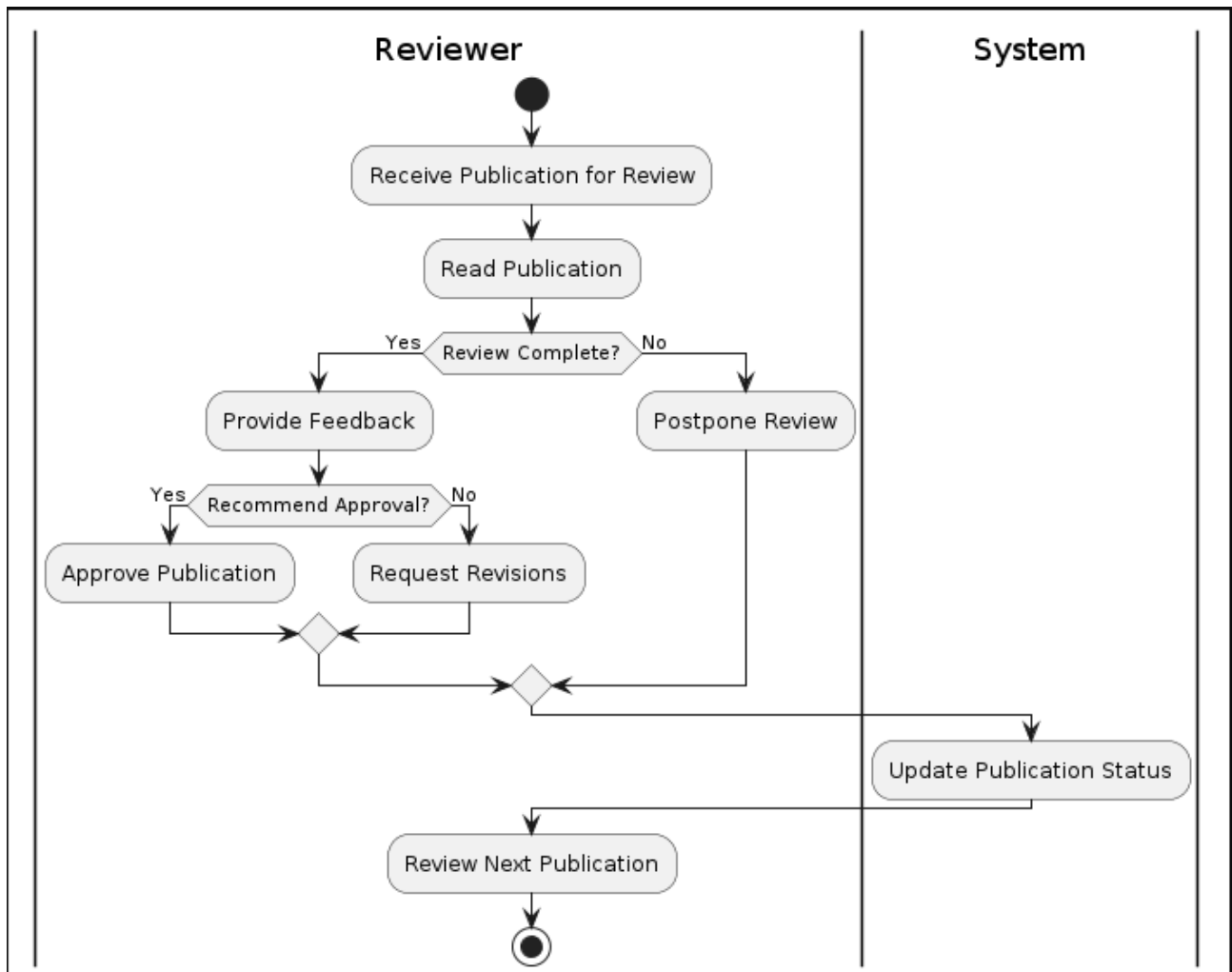


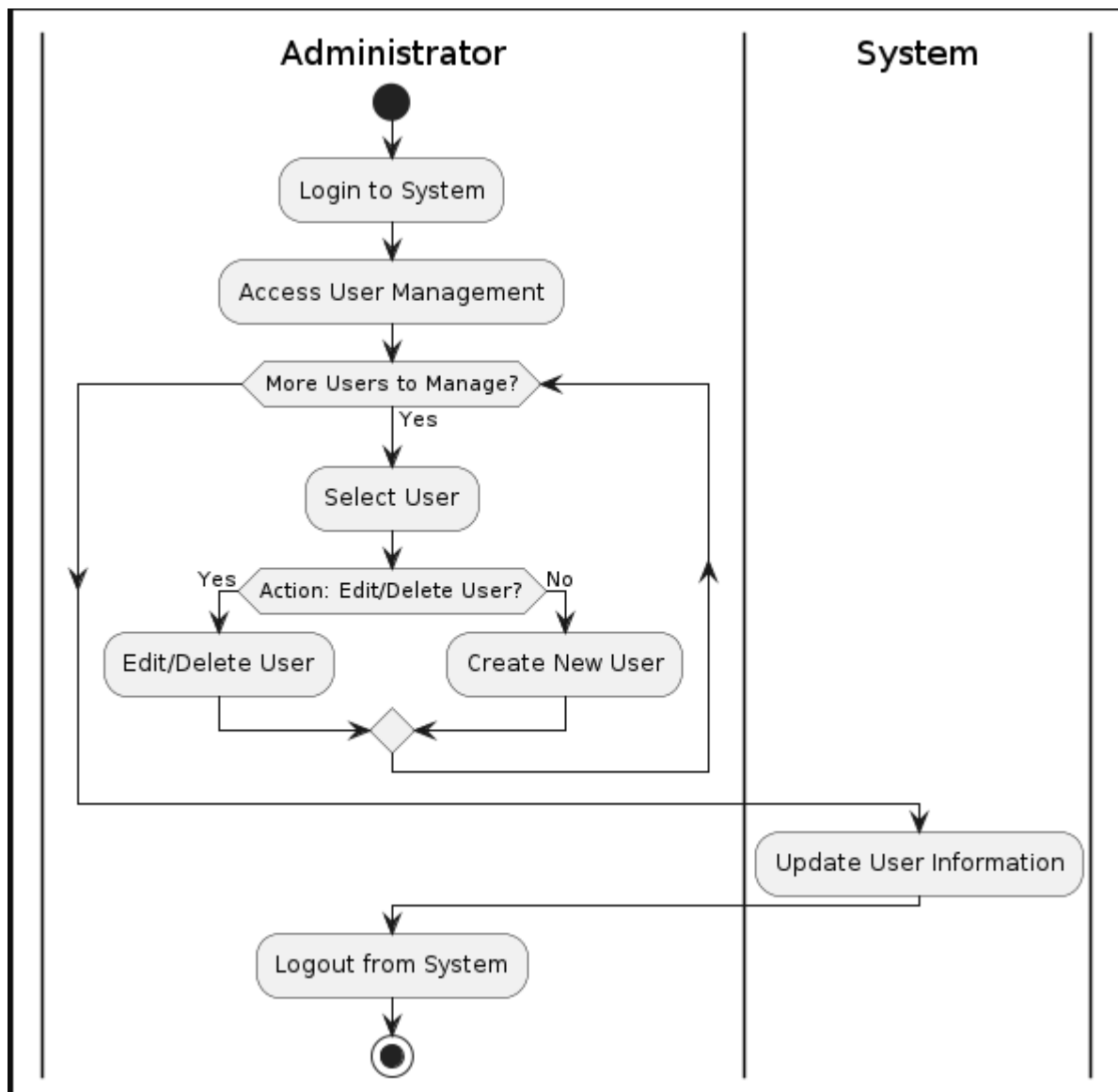
6.2 Activity Diagram

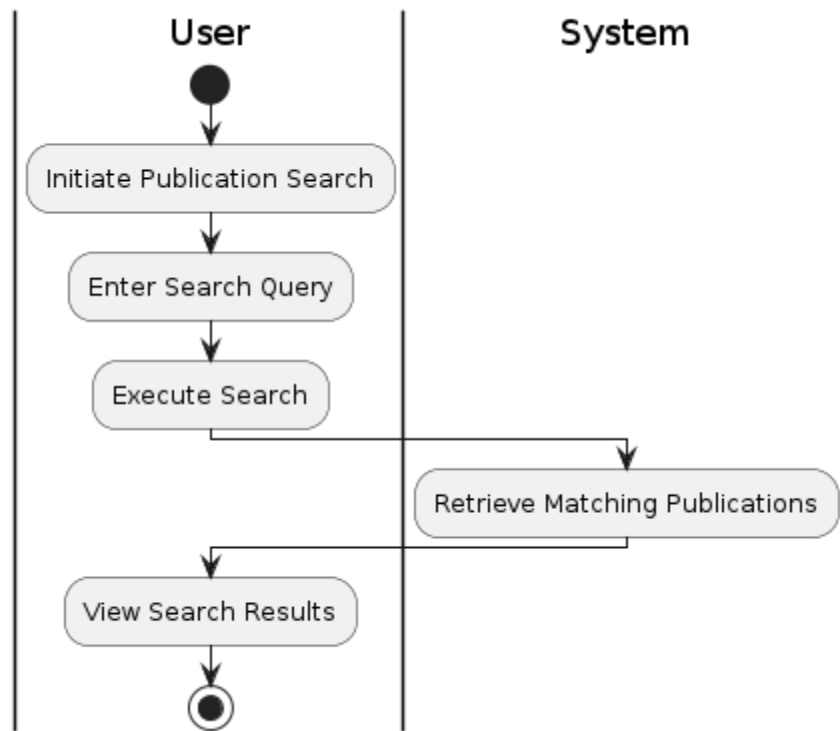
Activity Diagram for Publication Submission:



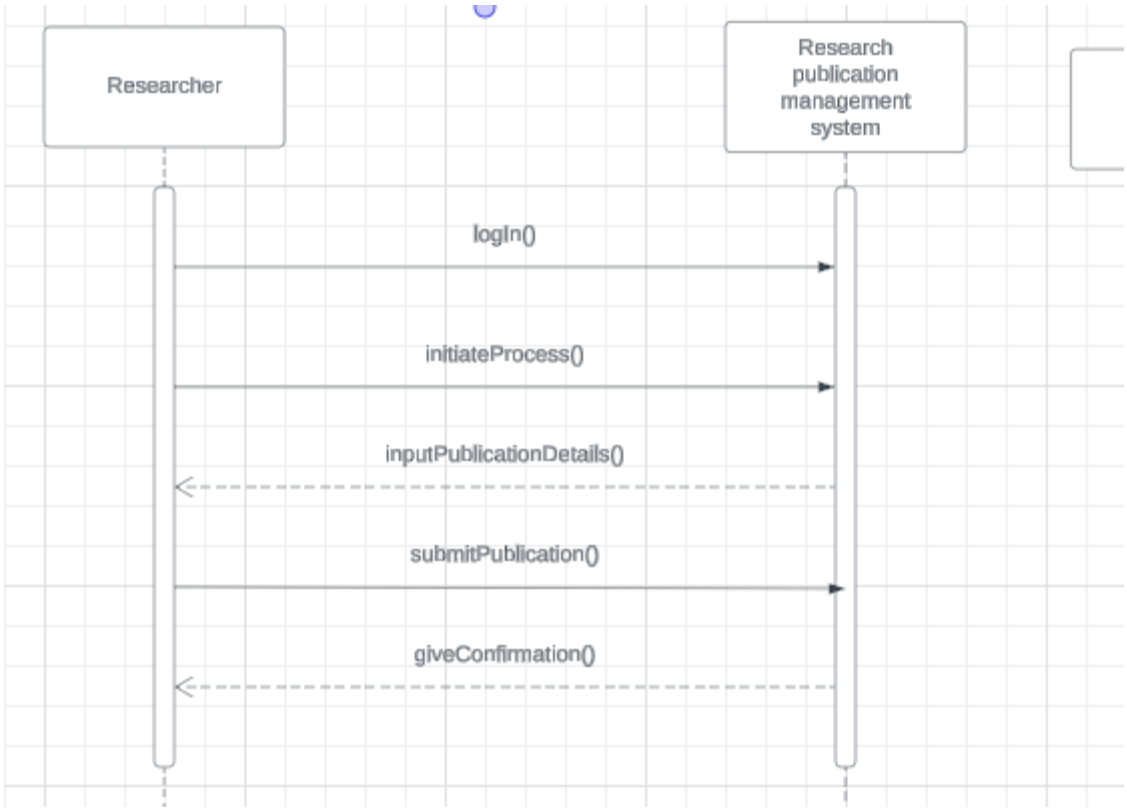
Activity Diagram for User Registration:

Activity Diagram for Publication Review:

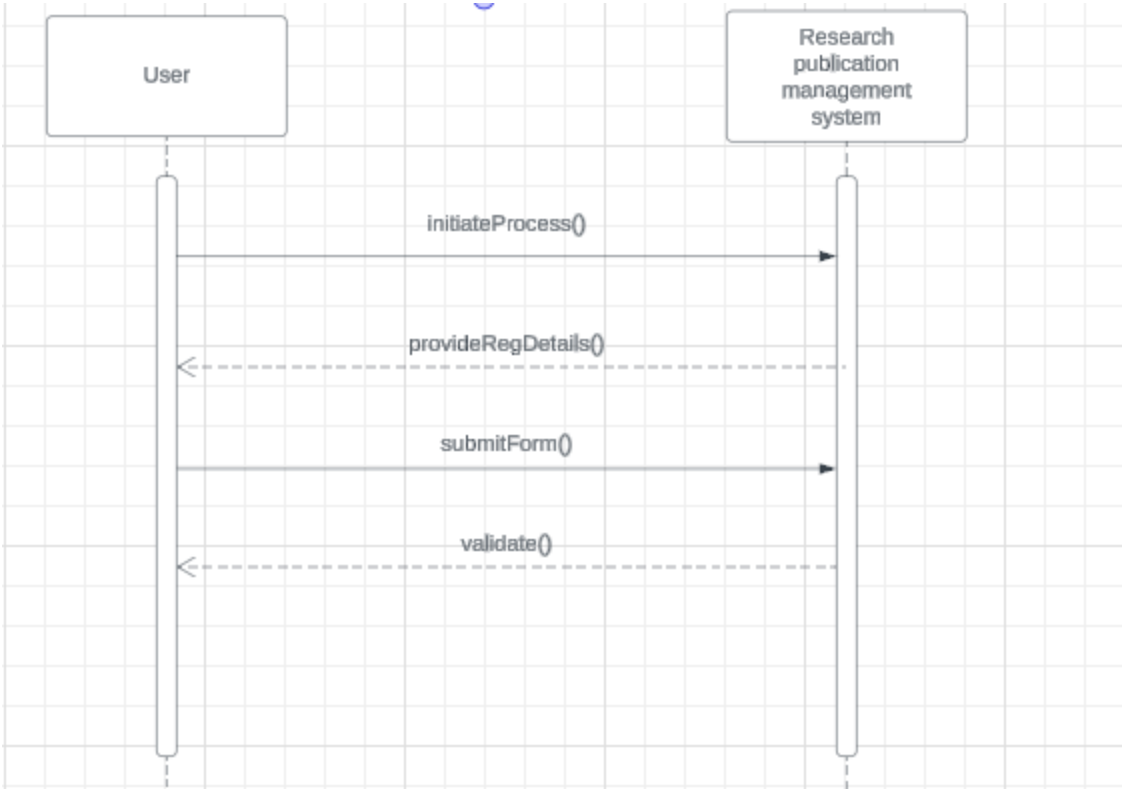
Activity Diagram for User Management:

Activity Diagram for Search Publications:**6.3 Sequence Diagram**

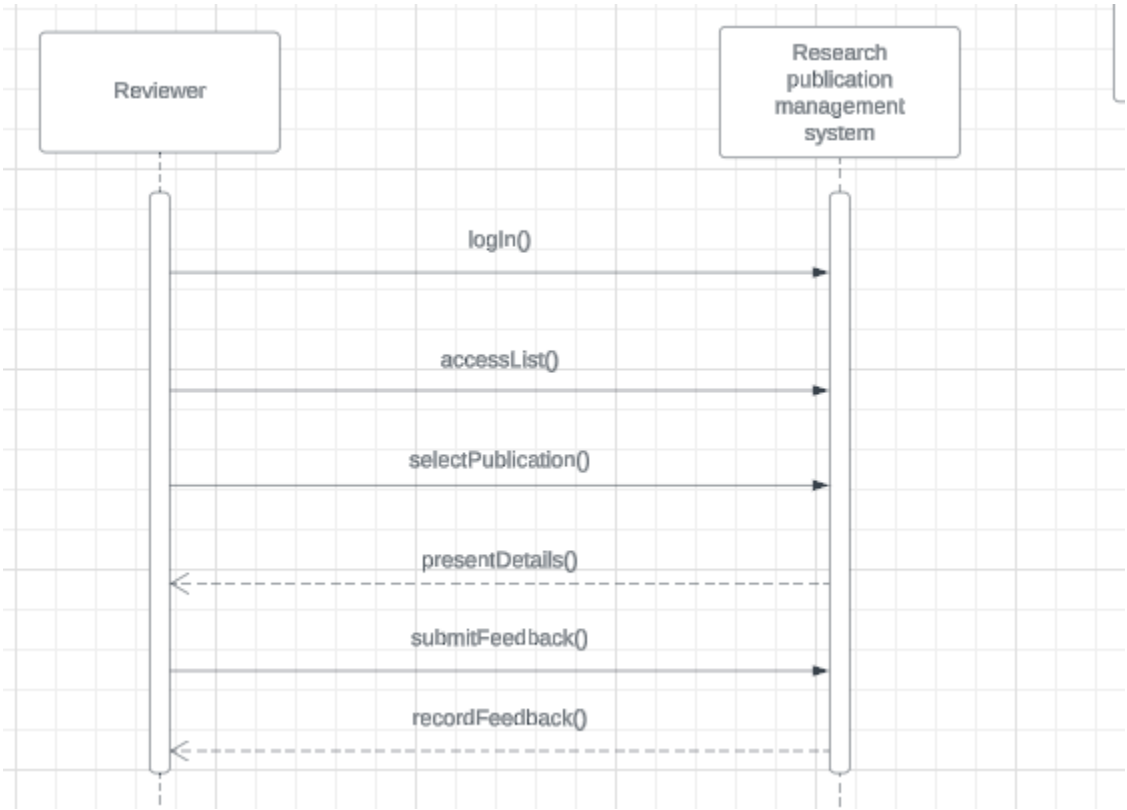
Publication Submission:



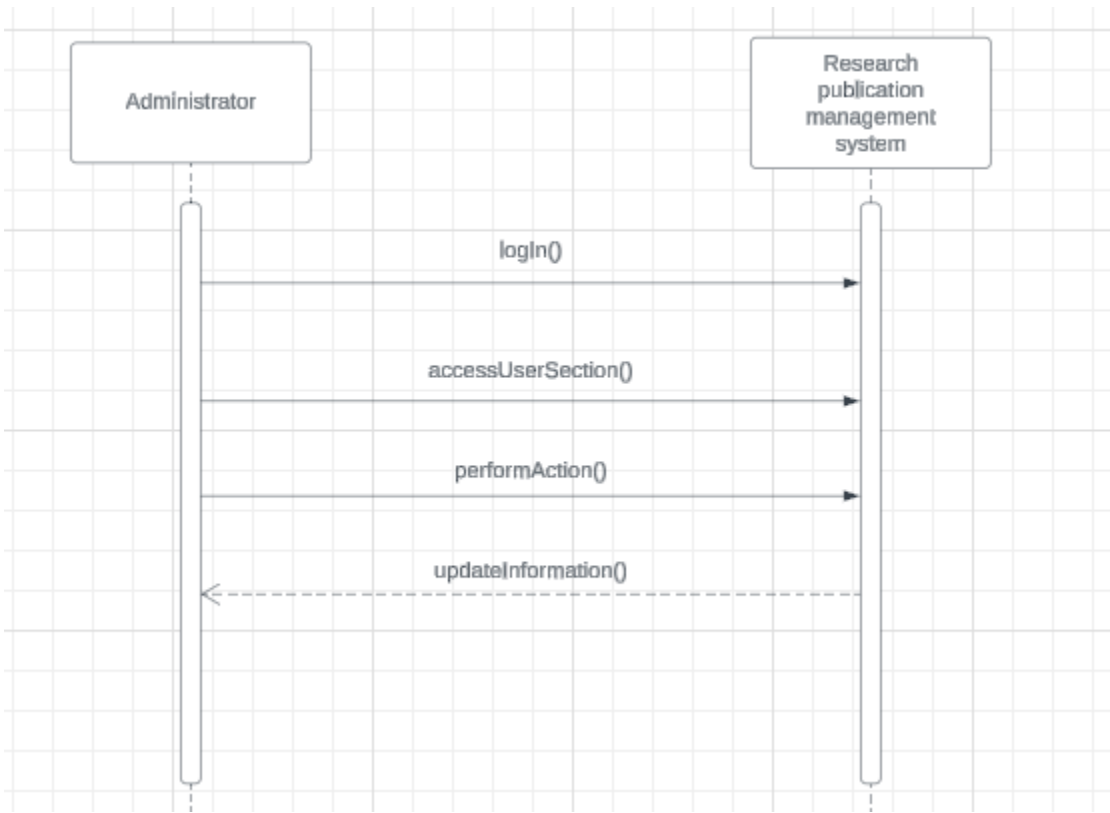
User Registration:



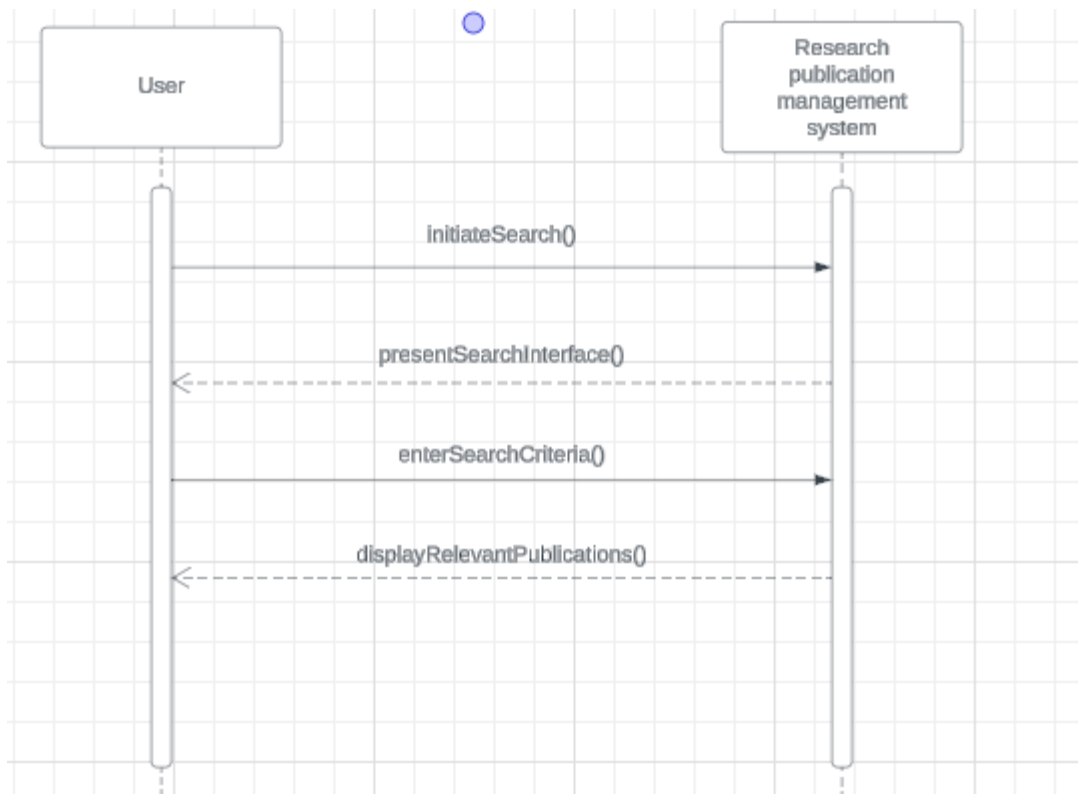
Publication Review:



User Management:



Search Publications:

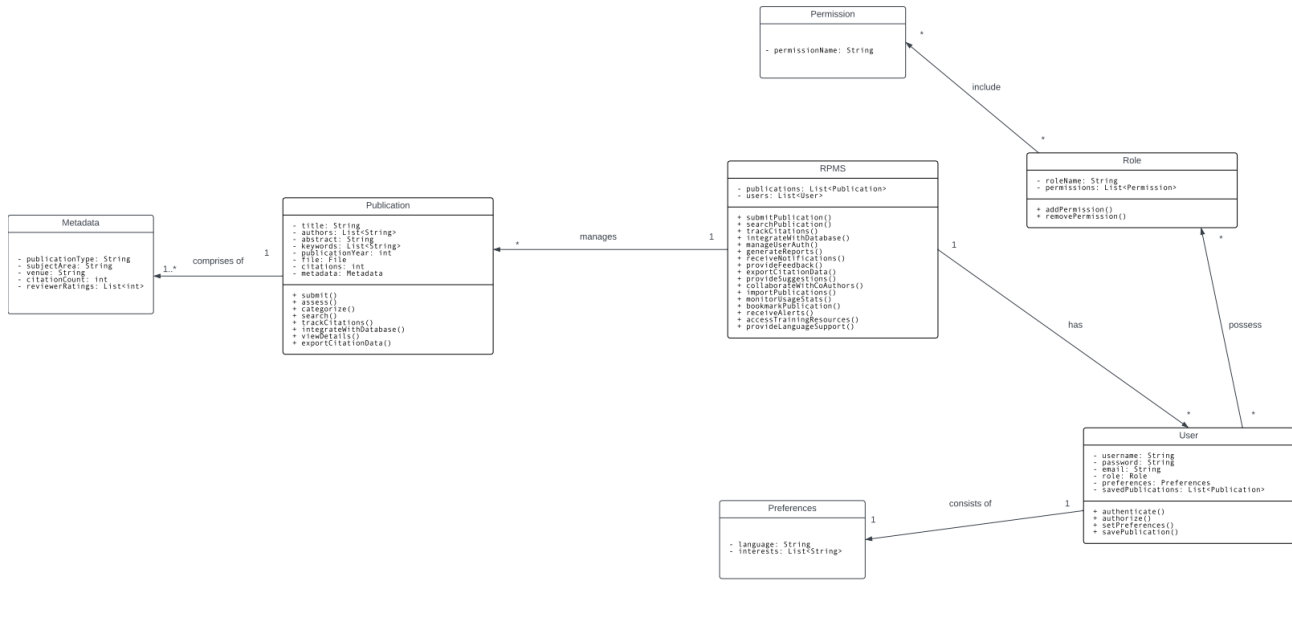


Appendix A: Glossary

- **RPM System:** Abbreviation for Research Publication Management System, the software product described in the document.
- **DBMS:** Acronym for Database Management System, software used to store and manage publication metadata, user information, and system configurations.
- **HTTP:** Acronym for Hypertext Transfer Protocol, a protocol used for transmitting data over the internet.
- **HTTPS:** Acronym for Hypertext Transfer Protocol Secure, an extension of HTTP for secure communication over a computer network.
- **SQL:** Acronym for Structured Query Language, a domain-specific language used in programming and designed for managing data held in a relational database management system.
- **API:** Acronym for Application Programming Interface, a set of rules and protocols for building and interacting with software applications.
- **SMTP:** Acronym for Simple Mail Transfer Protocol, a protocol used for sending email messages between servers.
- **POP3:** Acronym for Post Office Protocol version 3, a protocol used for retrieving email from a server.
- **IMAP:** Acronym for Internet Message Access Protocol, a protocol used for retrieving email from a server.
- **SSL/TLS:** Acronym for Secure Sockets Layer/Transport Layer Security, cryptographic protocols designed to provide communications security over a computer network.

Appendix B: Analysis Models

CLASS DIAGRAM:



Appendix C: To Be Determined List

- **TBD-1:** Clarify the process for user authentication and password recovery.
- **TBD-2:** Define the specific metrics and parameters for reporting and analytics.
- **TBD-3:** Determine the frequency and scheduling of system maintenance tasks.
- **TBD-4:** Specify the error handling mechanisms for invalid user inputs during publication submission.
- **TBD-5:** Define the backup and recovery procedures for the RPM System database.
- **TBD-6:** Determine the criteria for user role assignment and access control policies.
- **TBD-7:** Clarify the workflow for assigning reviewers to submitted publications.
- **TBD-8:** Define the process for integrating with external academic databases and repositories.
- **TBD-9:** Specify the data retention policy for archived publications and user data.
- **TBD-10:** Determine the criteria for prioritizing search results in the discovery interface.