UNIVERSITY OF BARISHAL
DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Software Requirements Specifications (SRS) on

Barishal University Journal

Course: CSE-3104
Software Engineering & Information System Design Lab

Submitted by:

Md Riajuddin Sikder 22 CSE 016	Khan Md Omar Faruk 22 CSE 018	Md Yeamin Talukder 22 CSE 020
Likhon Mandal 22 CSE 023	Abdullah Al Noman 22 CSE 038	Md Abdullah 22 CSE 048

Supervised by:
Md Samsuddoha
Assistant Professor, Dept. of CSE

Submission Date:
18 January 2026

Contents

Contributors

Name	Topic
Khan Md Omar Faruk, Md Abdullah	Inception (Finding Stakeholders, Requirements)
Md Yeamin Talukder, Abdullah Al Noman, Likhon Mandal	Collaboration Gathering, Quality Function Development, User Story & Description
Abdullah Al Noman, Md Yeamin Talukder, Md Abdullah	Use Case Scenario, Use Case Diagram
Md Abdullah, Abdullah Al Noman, Khan Md Omar Faruk	Activity Diagram
Md Abdullah, Likhon Mandal, Md Riajuddin Sikder, Abdullah Al Noman, Md Yeamin Talukder	Swimlane Diagram
Khan Md Omar Faruk	Database Modeling
Md Riajuddin Sikdar, Md Abdullah	Class-Based Modeling
Md Yeamin Talukder, Khan Md Omar Faruk	State Transition Diagram, Sequence Diagram
Md Riajudding Sikder, Likhon Mandal	Data Flow Diagram

Inception

Introduction

The University Research Journal Management System is a web-based platform designed to manage the submission, review, and publication of academic research papers within a university. The system aims to provide a structured and transparent environment where students and teachers can submit research work, participate in peer review, and access published papers across different faculties and departments.

The purpose of developing this system is to simplify the traditional journal management process by automating tasks such as paper submission, reviewer assignment, editorial decision-making, and publication. It also ensures proper role-based access so that different users can perform their responsibilities efficiently.

The goal of the inception phase for this system is to understand the scope of the project, identify the stakeholders involved, and recognize their needs and expectations. This phase helps in identifying potential conflicts, defining system boundaries, and establishing a clear understanding of how the system will operate.

Identifying Stakeholders

Based on the system analysis and stakeholder consultation, the following stakeholders have been identified for the BU Research Portal system. These stakeholders are individuals or groups who are directly or indirectly involved in the system or affected by its operation. Identifying these stakeholders is essential to understand their roles, responsibilities, and expectations, which helps ensure that the system meets academic, operational, and administrative requirements:

1. Admin
2. Author
3. Editor
4. Reviewer
5. Students
6. University

Viewpoints of Stakeholders

This section describes the concerns, expectations, and responsibilities of each stakeholder with respect to the BU Research Portal system.

1. Admin

The Admin represents the system authority responsible for governance and control of the platform.

- **System Oversight:** The Admin requires access to system-level information to monitor user registrations, role assignments, paper submissions, review progress, and overall platform activity.
- **Role Management:** The Admin needs the ability to approve or reject requests from Authors for promotion to Editor or Reviewer roles and to manage user accounts through role-based access control mechanisms.
- **Content monitor:** The Admin expects facilities to oversee submitted content, address conflicts or misuse, and ensure compliance with institutional research and publication policies.
- **Reporting and Support:** The Admin requires system-generated reports related to submissions, reviews, and publications, as well as tools to manage notifications and respond to user issues.

2. Author

The Author is a primary content contributor responsible for submitting and revising research papers.

- **Account and Role Requests:** The Author requires secure authentication and the ability to submit requests to the Admin for role promotion, with visibility into the approval status of such requests.
- **Paper Submission:** The Author needs facilities to submit research papers electronically, update submissions, and track the current status of each paper within the review workflow.
- **Revision and Feedback Handling:** The Author expects timely notifications regarding editorial decisions and access to consolidated reviewer feedback provided by the Editor for revision purposes.
- **Publication Tracking:** The Author requires access to historical information regarding submitted papers, revision cycles, and final publication outcomes.

3. Editor

The Editor is responsible for managing the peer review process and making publication decisions.

- **Paper Assignment:** The Editor requires tools to view submitted papers, assign multiple Reviewers to each paper, and monitor reviewer progress.
- **Review Coordination:** The Editor needs mechanisms to receive, analyze, and consolidate reviewer opinions and correction suggestions.
- **Editorial Decision Making:** The Editor expects system support to forward feedback to Authors for revision or to accept papers for publication based on reviewer evaluations.
- **Workflow Monitoring:** The Editor requires a dashboard or equivalent mechanism to track paper statuses, deadlines, and editorial workload.

4. Reviewer

The Reviewer is responsible for evaluating assigned research papers and providing expert feedback.

- **Assignment Notification:** The Reviewer requires notification of new paper assignments, including access details and submission deadlines.
- **Review Submission:** The Reviewer needs structured facilities to submit evaluations, comments, corrections, and recommendations to the Editor.
- **Secure Manuscript Access:** The Reviewer expects controlled and confidential access to assigned papers, limited strictly to their role and assignments.
- **Activity Tracking:** The Reviewer requires visibility into active and completed review assignments for reference and accountability.

5.Students

The Student is responsible for accessing and reading published research papers available on the BU Research Journal.

- Paper Browsing: Students can browse journals and published papers based on faculty, department, and research area.
- Search Functionality: Students can search for research papers using keywords, author names, or paper titles.
- Paper Access: Students are allowed to view and download published research papers for academic and learning purposes.

6.University

The University is responsible for overseeing and supporting the research activities conducted through the BU Research Portal.

- ❖ Research Management: The University manages and supports research publications across different faculties and departments.
- ❖ Academic Standards: The University ensures that published research follows academic and institutional guidelines.
- ❖ Research Access: The University provides a platform for sharing and accessing research work within the university community.

Common Requirements:

1. User-friendly and efficient system
2. Easy to operate for students, teachers, and administrators
3. The application can be accessed from any computer with Internet access

Conflicting Requirements:

1. Minimum system maintenance cost
2. Availability of required features within the allocated budget
3. Easy access for reading published papers
4. Clear and unambiguous system requirements

We finalized the following requirements for the system by categorizing and prioritizing the requirements.

Final Requirements:

1. Error-free system (Maximum 5% error may be considered acceptable).
2. Accessible via the Internet.
3. Allow valid users to register, log in, and log out of the system.
4. Restrict access to system functionalities based on assigned user roles (Admin, Author, Editor, Reviewer).
5. Allow administrators to manage journals, users, and role assignments.
6. Allow authors to submit research papers and upload revised versions.

7. Allow editors to assign reviewers and manage the review and publication process.
8. Allow reviewers to access assigned papers and submit reviews securely.
9. Allow students and teachers to browse, search, read, and download published research papers without logging in.
10. Provide notification services for submission status, review assignments, editorial decisions, and publication updates.

Elicitation

Introduction

Elicitation is the process of gathering requirements from stakeholders to understand what the system should do. The purpose of the elicitation phase is to collect, analyze, and clarify user needs and expectations. During this phase, several challenges may arise, such as unclear requirements, differences in user opinions, and scope limitations. To address these challenges, we conducted the elicitation process in an organized and systematic manner.

Eliciting Requirements

Unlike the inception phase where the focus is on identifying stakeholders and understanding the problem domain, the elicitation phase focuses on collecting detailed system requirements. This phase involves interaction with stakeholders to capture functional and non-functional requirements. To elicit requirements for the BU Research Portal system, we followed the activities described below.

The following elicitation works were carried out:

1. Collaborative Requirements Gathering
2. Quality Function Deployment
3. User Story
4. Usecase Description

1. Collaborative Requirements Gathering

Various approaches can be used for collaborative requirements gathering. In this project, we conducted structured discussions with key stakeholders involved in the research publication process.

The following steps were followed:

1. Meetings were conducted with faculty members involved in research and journal management.
2. Editors were consulted regarding the paper submission, review, and publication workflow.
3. Reviewers were asked about review criteria, deadlines, and feedback submission processes.
4. Students were consulted about accessing and reading published research papers.
5. Based on these discussions, the final list of system requirements was identified.

Usage Scenarios

Usage scenarios were created to understand how different users interact with the system.

1. Authors submit research papers and track submission status.
2. Editors assign reviewers and manage editorial decisions.
3. Reviewers review assigned papers and submit evaluations.
4. Students browse and read published research papers.
5. Administrators manage users, roles, and journals.

2. Quality Function Development

Normal Requirements

1. Admin account with full control over journals, users, and system operations
2. Separate role-based access for Admin, Author, Editor, and Reviewer
3. Multiple journals categorized by faculty and department
4. Interface for authors to submit research papers
5. Clear paper listings with title, authors, abstract, and publication details
6. Role-based dashboards for Admin, Author, Editor, and Reviewer
7. Paper browsing and searching options for all users
8. Notification service for submission, review, and publication updates

Expected Requirements

1. Responsive UI design for web (mobile and PC)
2. Secure document upload and download system
3. Reviewer assignment and review tracking system
4. Version control for revised paper submissions

Exciting Requirements

1. Blind or double-blind peer review option
2. Paper metrics (views and downloads)
3. Citation download option (APA, IEEE, BibTeX)
4. Comment or discussion section for published papers

3. User Story

User Registration, Login, and Role Management

This module will handle user registration, login, role assignment, and account management.

There will be four roles in the system: Admin, Author, Editor, and Reviewer.
Teachers can have multiple roles assigned at the same time.
Students will have only one role, which is Author.

Users must register before logging in.Registration will be done using a university email address.

Author Registration

Students and teachers who want to submit research papers will register as authors.
Authors will provide the following details:
Username, full name, university email, department, and password. Email verification is required before account activation.

Authors can submit research papers to selected journals. They can view submission status, upload revised versions, and view editorial decisions.
Authors will have a history of all submitted papers in their account.

Editor Module

The Editor module manages the review and publishing process of submitted papers.

Editors can view all submissions under their assigned journals. They can assign one or more reviewers to a paper and set review deadlines. Editors can make decisions such as accept, revise, or reject a paper. Editors control the final publication of accepted papers.

Reviewer Module

The Reviewer module allows reviewers to evaluate assigned research papers.

Reviewers will receive notifications when a paper is assigned to them.
They can download the paper and submit their review using a structured review form.
Reviewers can provide comments and recommendations for editors and authors.

Each reviewer will have a record of reviewed papers in their account.

Admin Module

The admin module is responsible for maintaining the overall system.

Admins can create and manage journals by faculty and department. Can assign roles to teachers.

Admins can monitor submission, review, and publication activities. Admins can manage users and system settings.

Reading and Browsing Papers

Students and teachers can browse and read published research papers.Users can search papers by title, author, journal, or department.
Published papers will be publicly accessible within the university system.

Notification Service

The Notification Service keeps users informed about important system events.

Authors will receive notifications for submission status and editorial decisions. Reviewers will receive notifications for review assignments and deadlines. Editors will receive notifications for new submissions and completed reviews. Admins will receive notifications for major system activities.

4. Use Case Descriptions

01. Authentication

i. Use case: Sign Up (Create Account)

Primary Actors: All Users
Goal in context: To create a new account in the BU Research Portal

Precondition:

1. System has registration interface
2. System has been programmed to store new user data in database
   Triggers: User needs to create an account
   Scenario:
3. Visit the Sign-Up page
4. Input required information (name, email, password, etc.)
5. System checks validity of inputs (email format, password rules)
6. System checks availability (email/username not already used)
7. System creates account and stores user data
8. System sends verification email (if required)
9. Confirmation message is displayed
   Exception:
10. Already exists: Email/username already registered
11. Ambiguous input: Missing/invalid fields
12. Registration failed: System/database error
    Priority: Essential, must be implemented
    When Available: First increment

ii. Use case: Sign In

Primary Actors: All Users
Goal in context: To enter the system securely
Precondition: Must be registered and active user
Triggers: User needs to log in

Scenario:

1. Visit the Sign In page
2. Input email/username and password
3. System validates credentials
4. System creates session and redirects user to dashboard/home
   Exception:
5. Unrecognized user: Account not found
6. Wrong password
7. User blocked/inactive
   Priority: Essential, must be implemented
   When Available: First increment

iii. Use case: Sign Out

Primary Actors: All Users
Goal in context: To exit from the system
Precondition: Must be logged in
Triggers: User wants to log out

Scenario:

1. Click the Sign Out button
2. System ends user session
3. User is redirected to public/home page
   Exception:
4. Session error: Logout fails due to system error
   Priority: Essential, must be implemented
   When Available: First increment

iv. Use case: Forgot Password

Primary Actors: All Users
Goal in context: To reset account password when forgotten
Precondition: User account must exist
Triggers: User cannot remember password

Scenario:

1. Visit the Forgot Password page
2. Enter registered email address
3. System validates email existence
4. System sends reset link/OTP to email
5. User opens link/enters OTP
6. User sets a new password
7. Confirmation message is displayed

   Exception:
8. Email not found
9. OTP/link expired
10. Weak password (fails rules)

    Priority: Essential, must be implemented

    When Available: First increment

v. Use case: Verify Email

Primary Actors: Author, Editor, Reviewer
Goal in context: To verify email for trusted submission/review access
Precondition:

1. User has registered account
2. System can send verification email

   Triggers: System requires email verification to enable role functions

   Scenario:
3. System sends verification email after registration/role assignment
4. User opens email and clicks verification link
5. System validates token/link
6. System marks email as verified
7. System shows confirmation message and enables access

   Exception:
8. Invalid/expired verification link
9. Email delivery failure

   Priority: Essential, must be implemented

   When Available: First increment

02. Repository & Search

i. Use case: Search Papers

Primary Actors: Public, All Users
Goal in context: To search papers in the repository
Precondition: System has searchable paper database/index
Triggers: User needs to find a paper

Scenario:

1. Visit Papers/Search page
2. Enter keywords/filters (title, author, year, topic)
3. Click Search button
4. System displays matching results list
5. User proceeds to view details or download (if desired)
   Exception:
6. No results found
7. Invalid filters/search query
   Priority: Essential, must be implemented
   When Available: First increment

ii. Use case: Show Paper

Primary Actors: Public, All Users
Goal in context: To view paper details and increase view count
Precondition: Paper exists in repository
Triggers: User selects a paper from results/list

Scenario:

1. User clicks a paper title
2. System opens paper details page (abstract, authors, keywords, journal, etc.)
3. System increases view count for the paper
4. User proceeds to download or save as favourite (if logged in)
   Exception:
5. Paper missing/unavailable

6. View counter update error
   Priority: Essential, must be implemented
   When Available: First increment

iii. Use case: Download Paper

Primary Actors: Public, All Users
Goal in context: To download paper file and increase download count
Precondition: Paper file exists and is accessible
Triggers: User clicks Download
Scenario:

1. User opens paper details page or result entry
2. Click Download button
3. System checks availability/permissions (if any)
4. System increases download count
5. System starts file download
   Exception:
6. File not found
7. Permission denied (restricted paper)
8. Download counter update error
   Priority: Essential, must be implemented
   When Available: First increment

iv. Use case: Search Journals

Primary Actors: Public, All Users
Goal in context: To find journals available in the portal
Precondition: Journal records exist in database
Triggers: User needs to search journal list

Scenario:

1. Visit Journals page
2. Enter journal name/keyword
3. Click Search
4. System displays matching journals
   Exception:
5. No journal found

6. Invalid search query
   Priority: Expected
   When Available: First increment

v. Use case: Show Journals

Primary Actors: Public, All Users
Goal in context: To view journal details and related papers
Precondition: Journal exists
Triggers: User selects a journal

Scenario:

1. User clicks a journal from the list
2. System shows journal details (scope, ISSN, papers list)
3. User proceeds to browse papers in that journal

Exception:

4. Journal missing/unavailable
   Priority: Expected
   When Available: First increment

vi. Use case: Sort Papers

Primary Actors: Public, All Users
Goal in context: To sort paper results for easier browsing
Precondition: Search results/list must be available
Triggers: User wants sorting (date, relevance, views, downloads)

Scenario:

1. User opens paper list/search results
2. Select sorting option (Newest, Most Viewed, etc.)
3. System reorders results accordingly
4. User continues browsing

   Exception:

5. Sorting option invalid

6. System fails to reload sorted list
   Priority: Expected
   When Available: First increment

vii. Use case: Save Paper as favourite

Primary Actors: Logged-in Users
Goal in context: To bookmark a paper for later access
Precondition:

1. User must be logged in
2. Paper exists
   Triggers: User wants to save a paper

   Scenario:
3. User opens paper details page
4. Click “Save as Favorite” button
5. System stores paper in user favorites list
6. Confirmation message displayed
   Exception:
7. Not logged in
8. Already saved: paper already in favorites
   Priority: Expected
   When Available: Second increment

03. Author Submission

i. Use case: Submit New Manuscript

Primary Actors: Author
Goal in context: To submit a new manuscript to the portal
Precondition: Author must be logged in and verified (if required)
Triggers: Author wants to submit a paper

Scenario:

1. Author logs in
2. Opens “New Submission” page
3. Enters manuscript metadata (title, abstract, keywords, journal, etc.)

4. Uploads manuscript file (or proceeds to upload step)
5. Confirms submission
6. System creates submission record and shows confirmation
   Exception:
7. Missing required fields
8. File missing/invalid format
9. System error while saving submission
   Priority: Essential, must be implemented
   When Available: First increment

ii. Use case: Upload Manuscript File

Primary Actors: Author
Goal in context: To upload the manuscript document for submission
Precondition: Author is in submission process and logged in
Triggers: Author needs to attach manuscript file

Scenario:

1. Author opens upload section
2. Chooses file (PDF/DOC as allowed)
3. System validates file type/size
4. System uploads and stores file securely
5. Upload success message displayed
   Exception:
6. Unsupported file type
7. File too large
8. Upload failed (network/server error)
   Priority: Essential, must be implemented
   When Available: First increment

iii. Use case: Submit Revision

Primary Actors: Author
Goal in context: To submit revised manuscript after review feedback
Precondition:

1. A previous submission exists
2. Revision is requested/open
   Triggers: Author receives revision request

Scenario:

3. Author logs in
4. Opens submission history
5. Selects the manuscript needing revision
6. Upload revised file and update notes/response
7. Click Submit Revision
8. System saves revision version and updates status

Exception:

9. Revision window closed
10. Invalid file upload
11. Submission not found

Priority: Essential, must be implemented
When Available: Second increment

iv. Use case: View Submission History

Primary Actors: Author
Goal in context: To view all submitted manuscripts and their details
Precondition: Author must be logged in
Triggers: Author wants to see past submissions

Scenario:

1. Author logs in
2. Opens “Submission History” page
3. System displays list of submissions with status and dates
4. Author selects one to view details

Exception:

5. No submissions found
6. System cannot load history

Priority: Expected
When Available: First increment

v. Use case: Track Paper Status

Primary Actors: Author
Goal in context: To track the current workflow status of a submitted paper
Precondition: Manuscript submission exists
Triggers: Author wants progress updates

Scenario:

1. Author logs in
2. Opens a submitted manuscript
3. System displays status (Under Review, Revision, Accepted, Rejected, Published)
4. System shows timeline/updates if available

   Exception:
5. Submission not found
6. Status update unavailable

   Priority: Expected

   When Available: First increment

04. Editorial Workflow

i. Use case: Assign Editor to Paper

Primary Actors: Admin

Goal in context: To assign an editor responsible for a submitted paper

Precondition:

1. Admin must be logged in
2. Paper submission exists
3. Editor account exists

   Triggers: New submission requires editorial handling

   Scenario:
4. Admin logs in
5. Opens pending submissions list
6. Selects a paper
7. Chooses an editor from editor list
8. Confirms assignment
9. System updates paper record and notifies editor

   Exception:
10. Paper not found
11. Editor not available/invalid
12. Assignment save failed

    Priority: Essential, must be implemented

    When Available: First increment

ii. Use case: Assign Reviewer to Paper

Primary Actors: Editor
Goal in context: To assign reviewers to evaluate a manuscript
Precondition:

1. Editor is logged in
2. Paper is assigned to editor
3. Reviewer accounts exist

Triggers: Paper enters review phase

Scenario:

4. Editor logs in
5. Opens assigned papers
6. Selects a paper
7. Searches/selects reviewer(s)
8. Confirms assignment
9. System notifies reviewer(s)

Exception:

10. Reviewer unavailable/conflict
11. Paper not eligible for review stage
12. Notification failure

Priority: Essential, must be implemented
When Available: First increment

iii. Use case: Set Review Deadlines

Primary Actors: Editor
Goal in context: To set deadlines for reviewers
Precondition: Paper has assigned reviewer(s)
Triggers: Editor wants time-bound review process

Scenario:

1. Editor opens paper review settings
2. Selects deadline date
3. Confirms deadline
4. System stores deadline and notifies reviewers

Exception:

5. Invalid date (past date)

6. Reviewers not assigned yet
   Priority: Expected
   When Available: Second increment

iv. Use case: Download Manuscript

Primary Actors: Editor, Reviewer
Goal in context: To download manuscript file for evaluation
Precondition:

1. User logged in as editor/reviewer
2. Paper assigned to them
3. File exists
   Triggers: Need to read manuscript offline

   Scenario:
4. User logs in
5. Opens assigned papers
6. Selects a paper
7. Clicks Download Manuscript
8. System checks permission and starts download
   Exception:
9. Not assigned: access denied
10. File missing
    Priority: Essential, must be implemented
    When Available: First increment

v. Use case: Submit Evaluation/Feedback

Primary Actors: Reviewer
Goal in context: To submit review comments and recommendation
Precondition:

1. Reviewer logged in
2. Paper assigned and within deadline (if enforced)
   Triggers: Reviewer completes review

Scenario:

3. Reviewer opens assigned paper
4. Enters comments/feedback (confidential + author comments if applicable)
5. Selects recommendation (Minor/Major revision, Accept, Reject)
6. Clicks Submit Feedback
7. System saves evaluation and notifies editor

Exception:

8. Deadline exceeded
9. Missing required feedback fields
10. Save failed

Priority: Essential, must be implemented
When Available: First increment

vi. Use case: Submit Paper Verdict

Primary Actors: Editor
Goal in context: To decide the final verdict based on reviews
Precondition: Review feedback exists
Triggers: Editor ready to make decision

Scenario:

1. Editor opens paper details
2. Reads reviewer evaluations
3. Selects verdict (Accept/Reject/Revision)
4. Submits verdict
5. System updates status and notifies author

Exception:

6. Reviews not completed
7. Invalid decision/state conflict

Priority: Essential, must be implemented
When Available: First increment

vii. Use case: Publish Accepted Paper

Primary Actors: Editor
Goal in context: To publish an accepted paper to the public repository
Precondition: Paper status must be “Accepted”
Triggers: Editor proceeds to publish

Scenario:

1. Editor opens accepted papers list
2. Selects paper
3. Confirms publication (journal issue/metadata)
4. System publishes paper and makes it visible publicly
5. System notifies author and updates repository index

   Exception:
6. Paper not accepted
7. Missing metadata for publication
8. Publication failed

   Priority: Essential, must be implemented
   When Available: Second increment

viii. Use case: View Assigned Papers

Primary Actors: Reviewer, Editor
Goal in context: To view papers assigned for review/editing
Precondition: User must be logged in
Triggers: User needs to access their workload

Scenario:

1. User logs in
2. Opens “Assigned Papers” page
3. System displays list with status and deadlines
4. User selects a paper to proceed

   Exception:
5. No assigned papers
6. Load error

   Priority: Expected
   When Available: First increment

05. Administration

i. Use case: Manage Journals (Create/Edit/Delete)

Primary Actors: Admin
Goal in context: To maintain journal list in the portal

Precondition: Admin must be logged in
Triggers: Admin needs to add/update/remove journals

Scenario:

1. Admin logs in
2. Opens journal management panel
3. Selects create/edit/delete action
4. Enters/updates journal details
5. Confirms changes
6. System updates database and shows confirmation

   Exception:
7. Duplicate journal (already exists)
8. Journal not found (for edit/delete)
9. Save/delete failed

   Priority: Essential, must be implemented
   When Available: First increment

ii. Use case: Manage User Accounts

Primary Actors: Admin
Goal in context: To create, update, block, or remove user accounts
Precondition: Admin must be logged in
Triggers: Admin needs to manage users

Scenario:

1. Admin logs in
2. Opens user management page
3. Searches/selects user
4. Performs action (edit info/block/delete)
5. Confirms action
6. System updates user record and shows message

   Exception:
7. User not found
8. Unauthorized action
9. Update failed

   Priority: Essential, must be implemented
   When Available: First increment

iii. Use case: Up/Down Role for User

Primary Actors: Admin
Goal in context: To change user role (promote/demote permissions)
Precondition:

1. Admin logged in
2. User exists

Triggers: Admin needs to assign/change role

Scenario:

3. Admin opens user account details
4. Selects role change option
5. Chooses new role (e.g., Author → Reviewer, Reviewer → Editor)
6. Confirms update
7. System updates permissions and notifies user (optional)

Exception:

8. Invalid role assignment
9. User not eligible for role
10. Update failed

Priority: Expected
When Available: Second increment

iv. Use case: Submit Report Form

Primary Actors: Admin
Goal in context: To submit a formal report (issue/summary/compliance)
Precondition: Admin must be logged in
Triggers: Admin needs to file a report

Scenario:

1. Admin opens Report Form page
2. Enters report details
3. Attaches evidence (optional)
4. Clicks Submit
5. System saves report and shows confirmation

Exception:

6. Missing required fields

7. Submission failed
   Priority: Expected
   When Available: Second increment

v. Use case: Generate System Reports

Primary Actors: Admin
Goal in context: To generate system-level reports (usage, submissions, downloads, etc.)
Precondition: Admin must be logged in and reporting module available
Triggers: Admin needs system analytics/reports

Scenario:

1. Admin opens Reports page
2. Selects report type and date range
3. Clicks Generate
4. System processes data and displays report
5. Admin exports/downloads report (optional)
   Exception:
6. Invalid date range
7. No data available
8. Report generation failed
   Priority: Expected
   When Available: Second increment

06. Communication

i. Use case: View Notifications

Primary Actors: All Users
Goal in context: To view system notifications (status updates, assignments, alerts)
Precondition: User must be registered (and logged in for personal notifications)
Triggers: User wants to check updates

Scenario:

1. User logs in (if required)
2. Opens notifications panel/page
3. System displays latest notifications

4. User opens a notification to view details
   Exception:
5. No notifications available
6. Load error
   Priority: Expected
   When Available: First increment

Use Case Scenario:

Level – 0	Level – 1	Level – 2	Actors
	Authentication	Sign Up (Create Account)	All Users
		Sign In	All Users
		Sign Out	All Users
		Forgot Password	All Users
		Verify Email	Author, Editor, Reviewer
	Repository & Search	Search Papers	Public, All Users
		Show Paper (View Details & Count)	Public, All Users
		Download Paper (Inc. DL Count)	Public, All Users
		Search Journals	Public, All Users
		Show Journals	Public, All Users
		Sort Papers	Public, All Users

BU Research Portal		Save Paper as Favorite	Logged-in Users
	Author Submission	Submit New Manuscript	Author
		Upload Manuscript File	Author
		Submit Revision	Author
		View Submission History	Author
		Track Paper Status	Author
	Editorial Workflow	Assign Editor to Paper	Admin
		Assign Reviewer to Paper	Editor
		Set Review Deadlines	Editor
		Download Manuscript (For Review)	Editor, Reviewer
		Submit Evaluation/Feedback	Reviewer
		Submit Paper Verdict (Accept/Reject)	Editor
		Publish Accepted Paper	Editor

		View Assigned Papers	Reviewer, Editor
	Administration	Manage Journals (Create/Edit/Delete)	Admin
		Manage User Accounts	All User
		Up/Down Role for User	Admin
		Submit Report Form	Author,Editor, Reviewer
		Generate System Reports	Admin
	Communication	View Notifications	All Users

Use Case Diagram

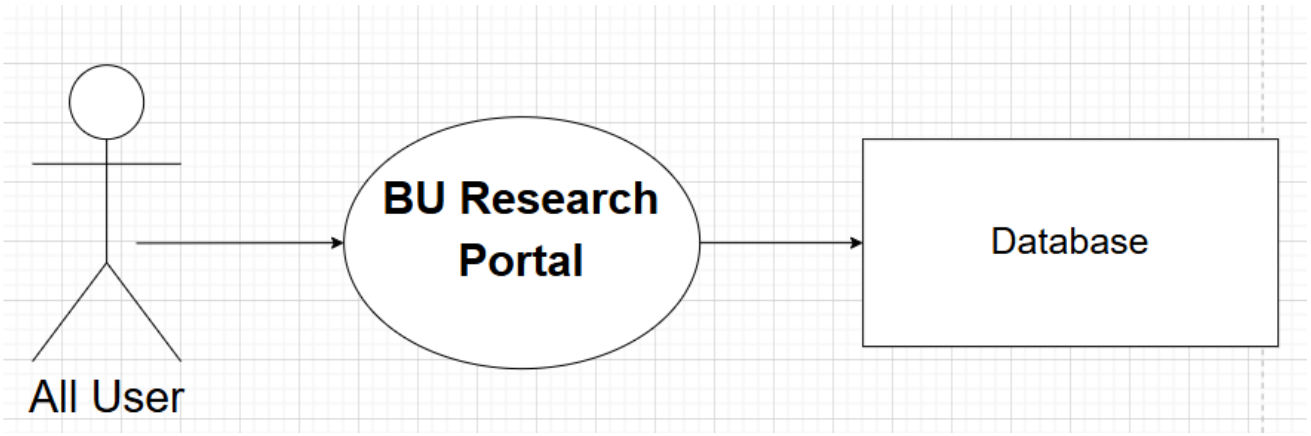


Fig : Level 0 for BU Research Portal

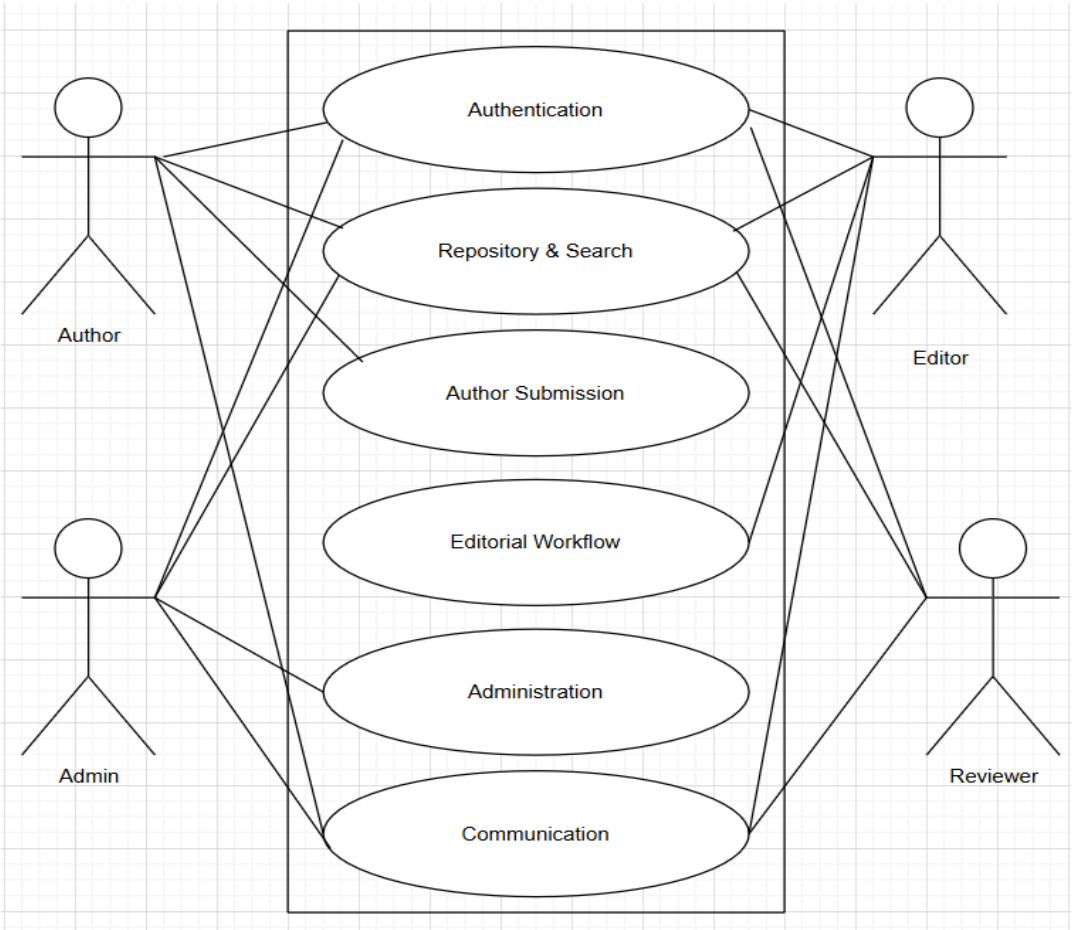


Fig: Level 1 for BU Research Portal

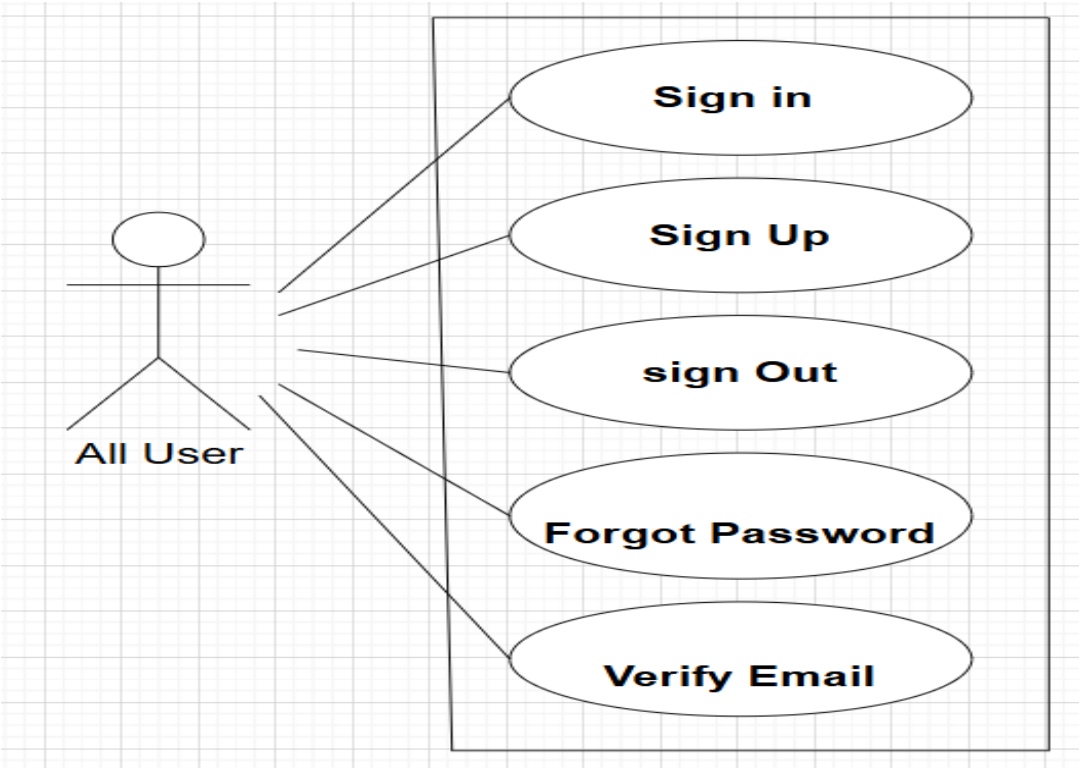


Fig : Level 2.1(Authentication) for BU Research Portal

This use case diagram represents an authentication system where an **All User** actor can perform actions such as Sign In, Sign Up, Sign Out, Forgot Password, and Verify Email. It illustrates the basic user interactions required to access, manage, and secure user accounts within the system.

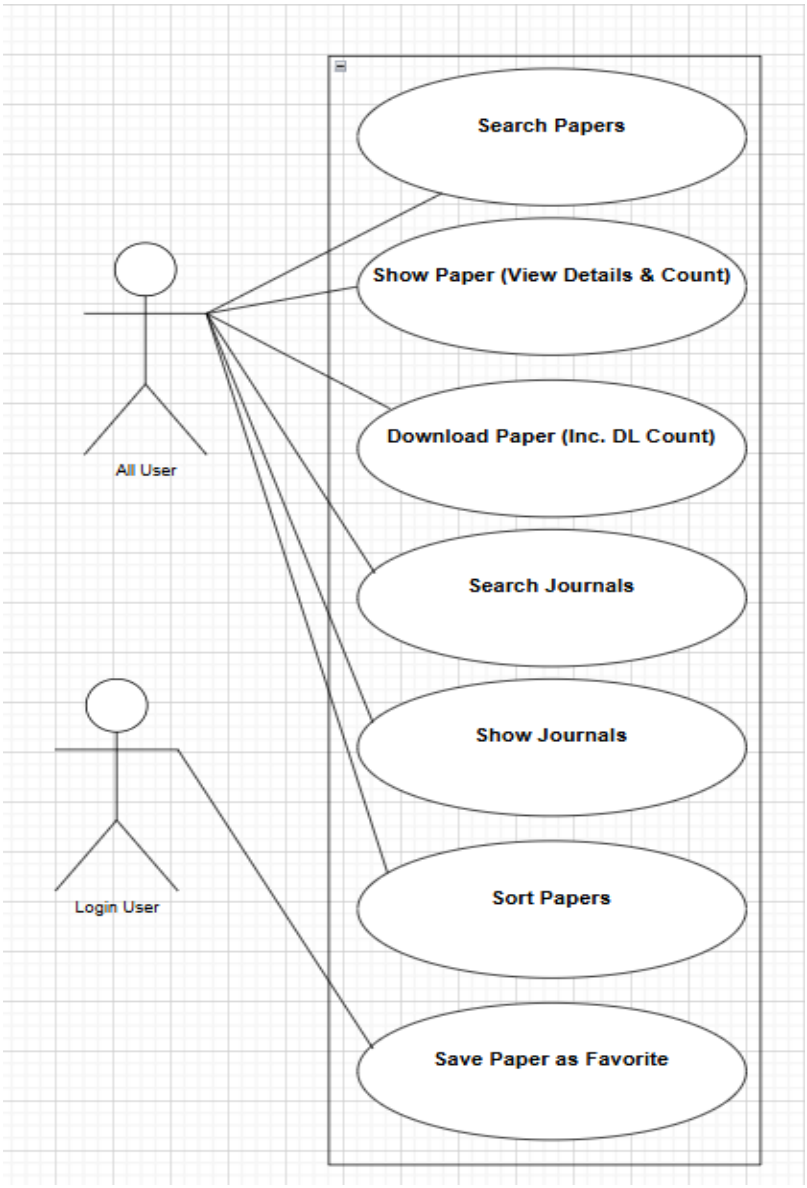


Fig : Level 2.2(Repository and Search) for BU Research Portal

This use case diagram represents an academic paper management system where all users can search, view, download, and browse papers and journals.Logged-in users have additional privileges such as sorting papers and saving papers as favorites.

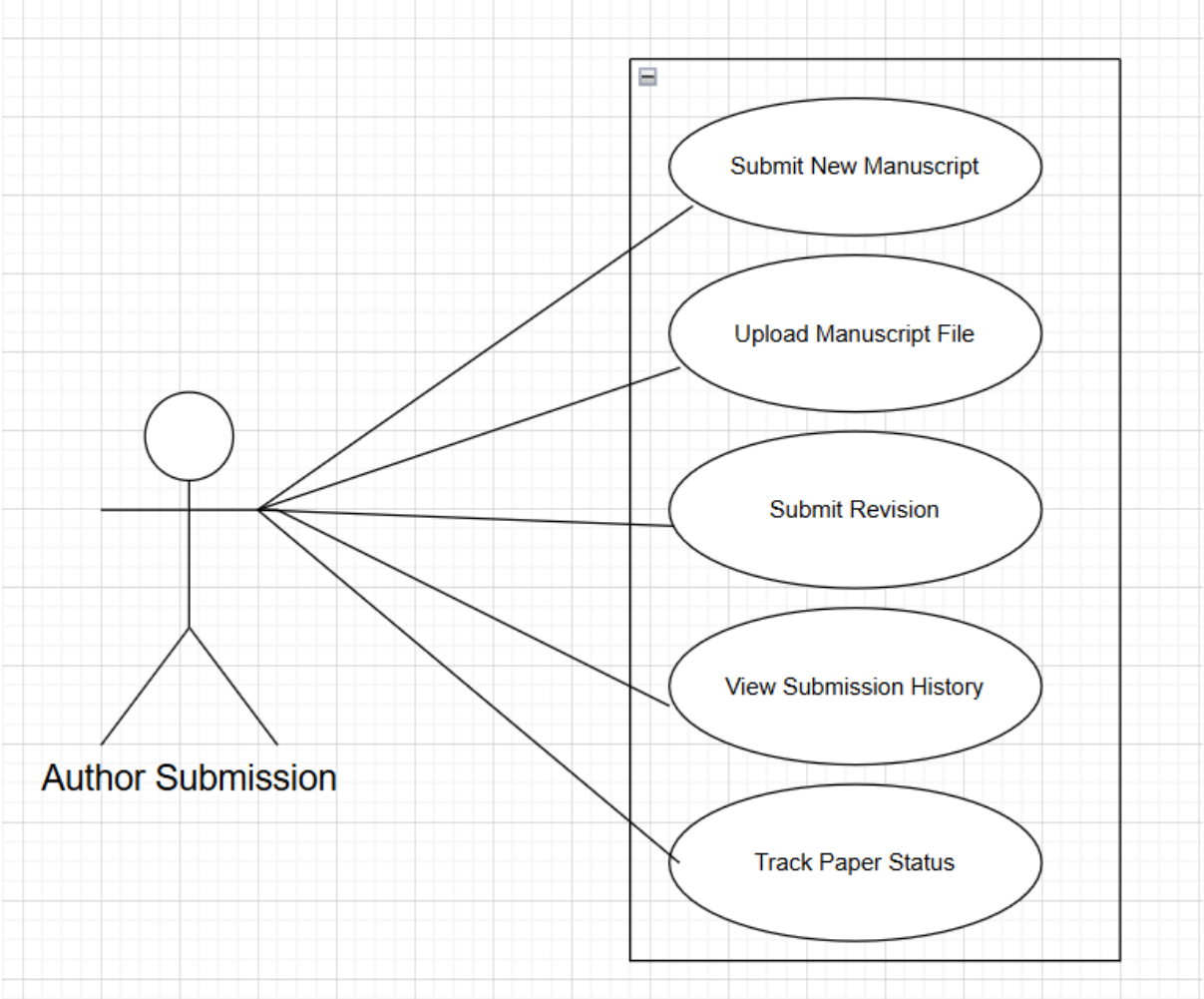


Fig : Level 2.3(Author Submission) for BU Research Portal

This use case diagram illustrates the author submission module of a journal system where authors can submit new manuscripts and upload files.Authors can also submit revisions, view their submission history, and track the current status of their papers.

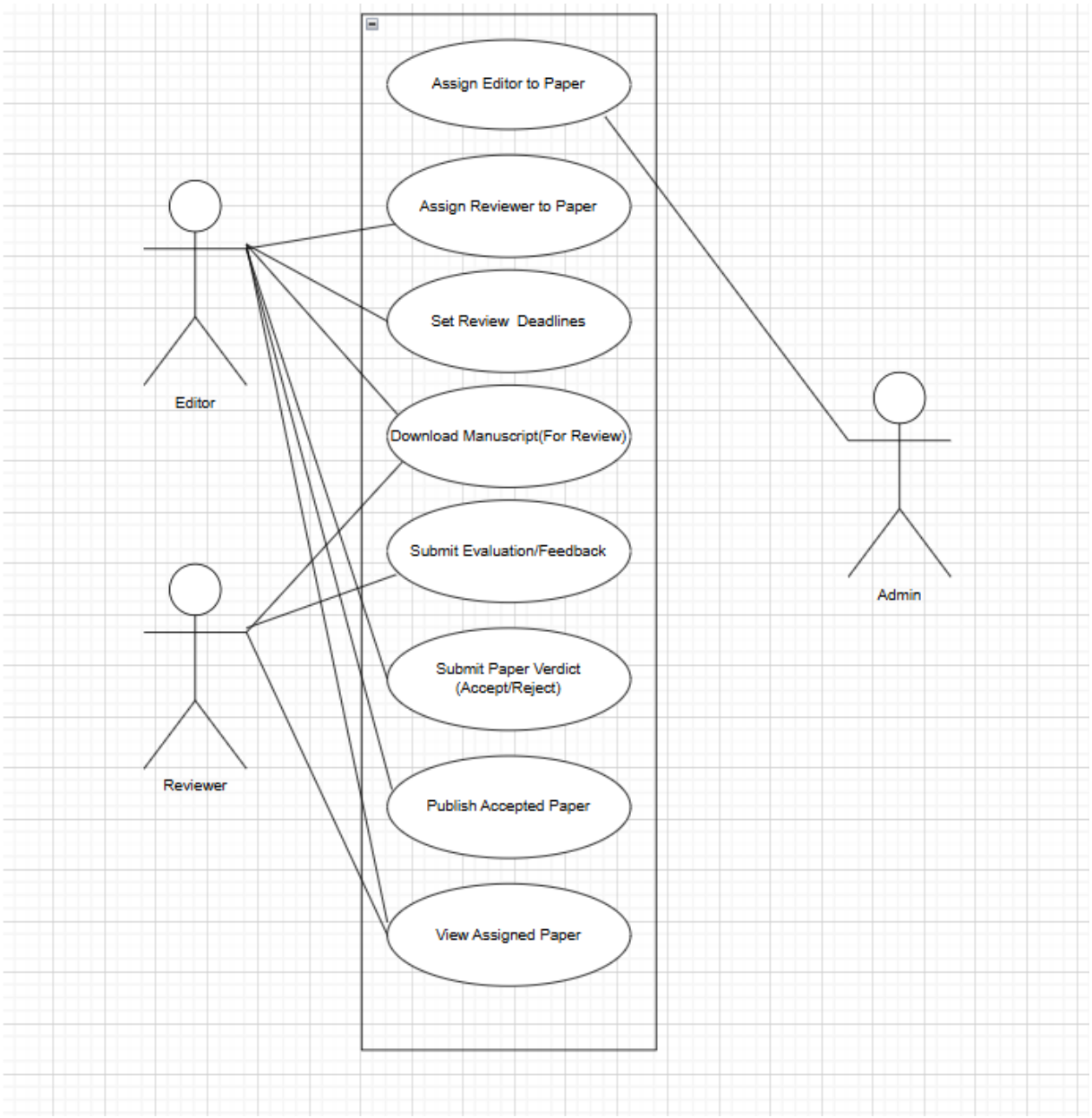


Fig : Level 2.4(Editorial Workflow) for BU Research Portal

This use case diagram shows the editorial workflow of a journal system involving editors, reviewers, and administrators.Editors and reviewers manage paper assignments, reviews, feedback, and verdicts, while admins oversee editor assignment and publication of accepted papers.

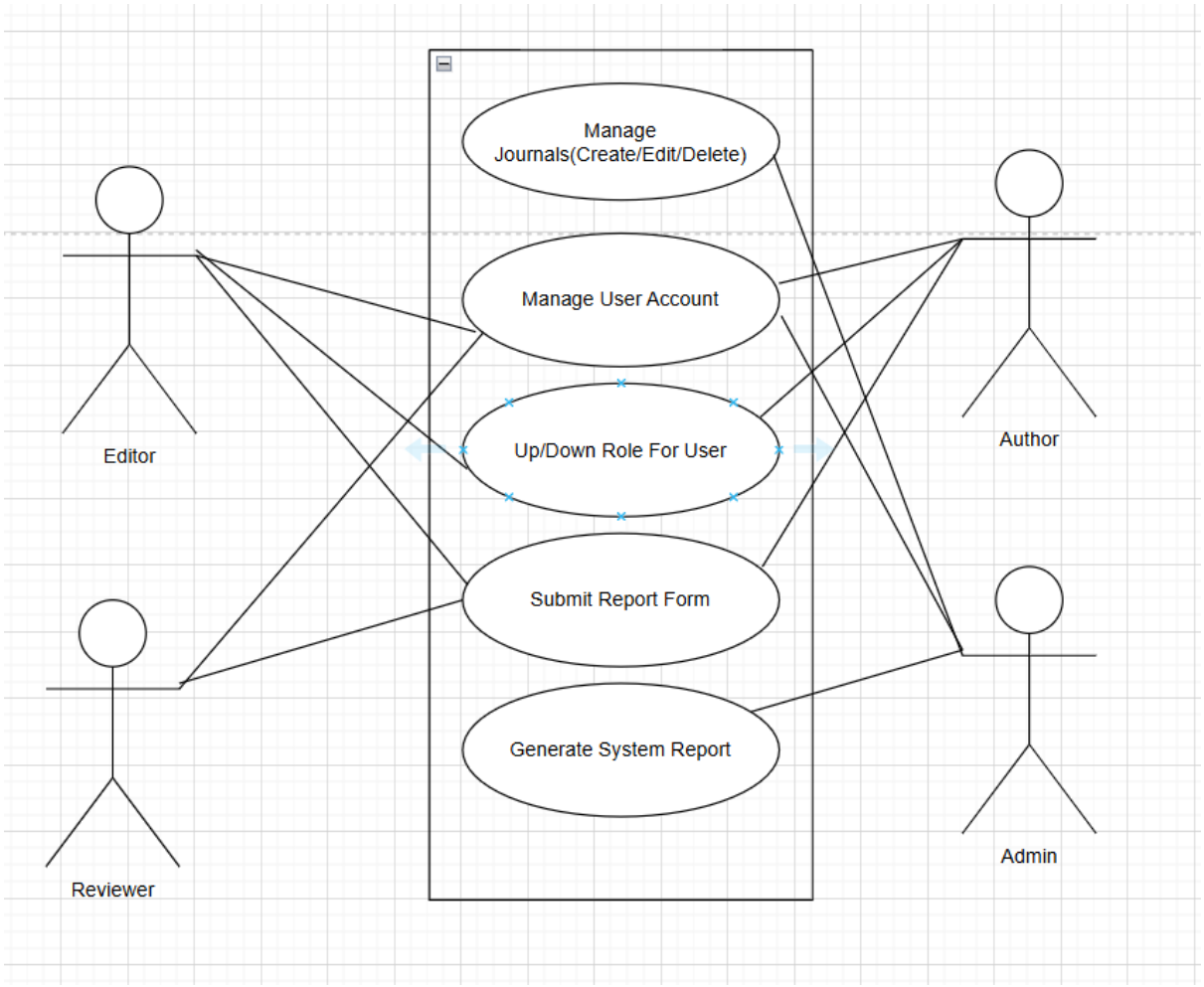


Fig : Level 2.5 (Administration) for BU Research Portal

This use case diagram represents the system administration module where editors, reviewers, authors, and admins interact with management functions.Administrators manage journals, user accounts, roles, and generate system reports, while other users can submit reports and access relevant account features.

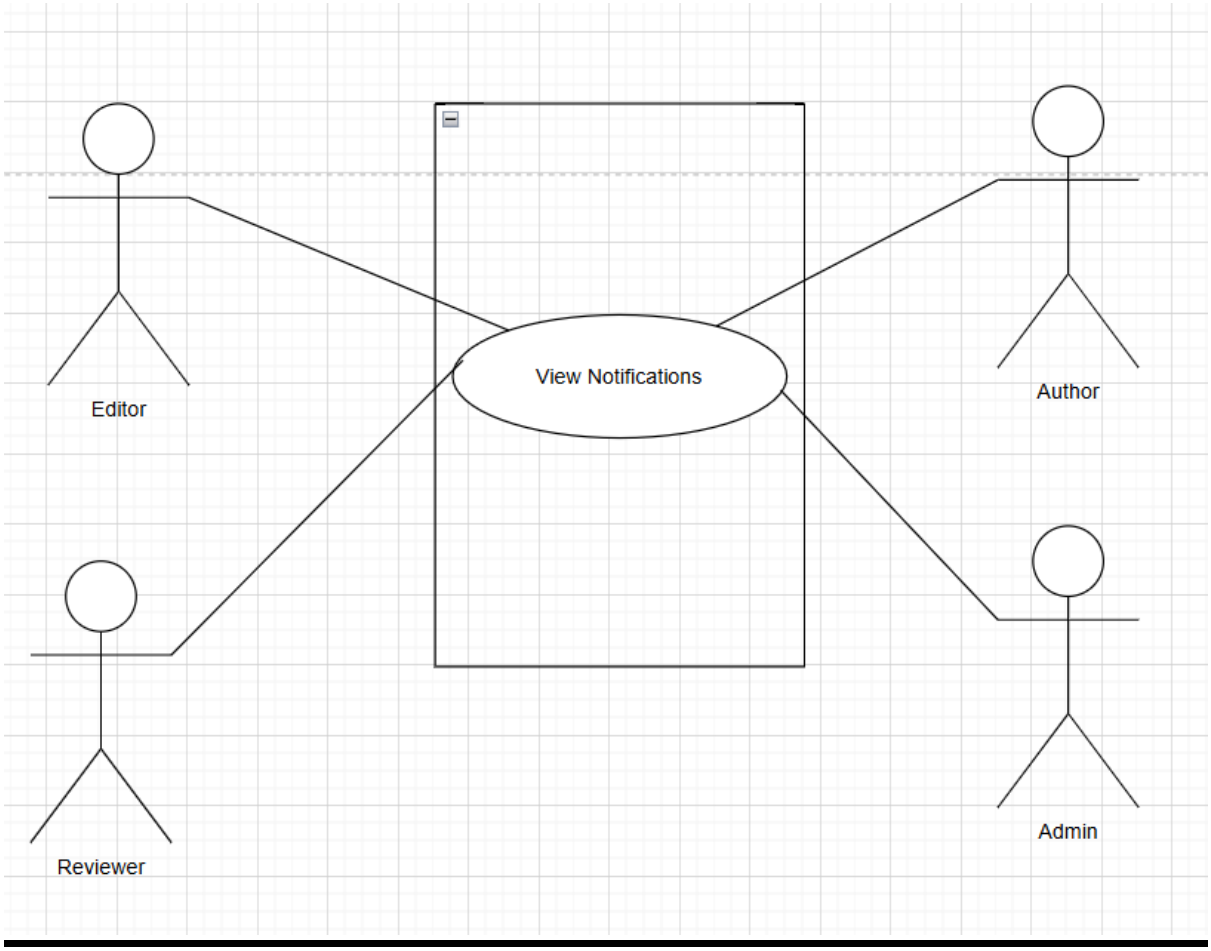


Fig : Level 2.6 (Communication) for BU Research Portal

Activity Diagram

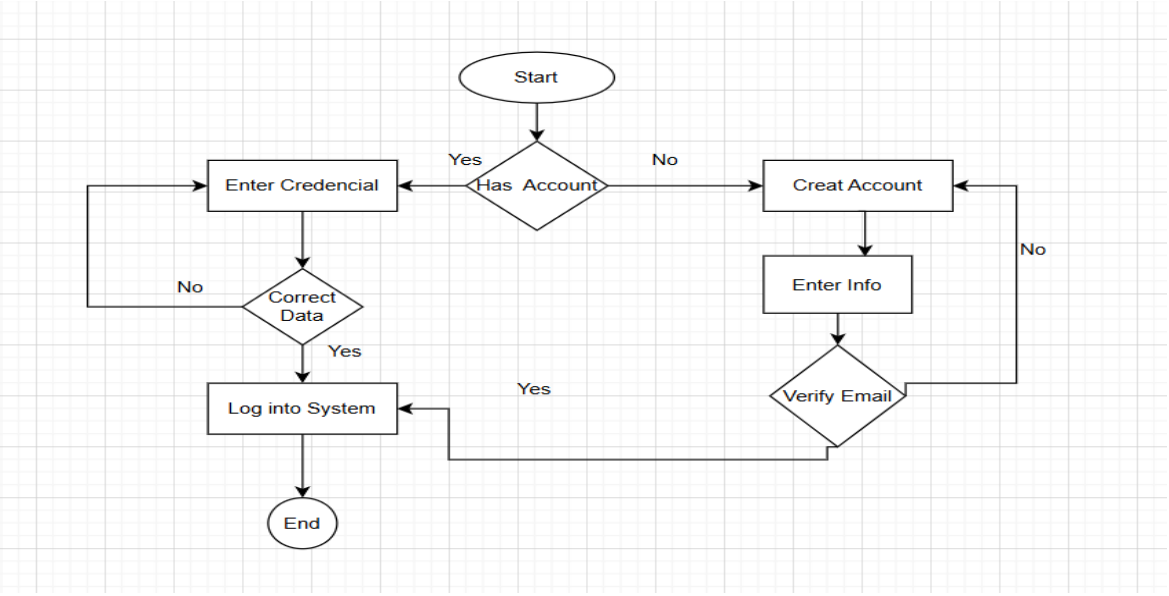


Fig : Login System

This activity diagram illustrates the user authentication process, starting from checking whether a user already has an account.Users either log in with valid credentials or create a new account by entering details and verifying their email before accessing the system.

Activity Diagram: Registration System

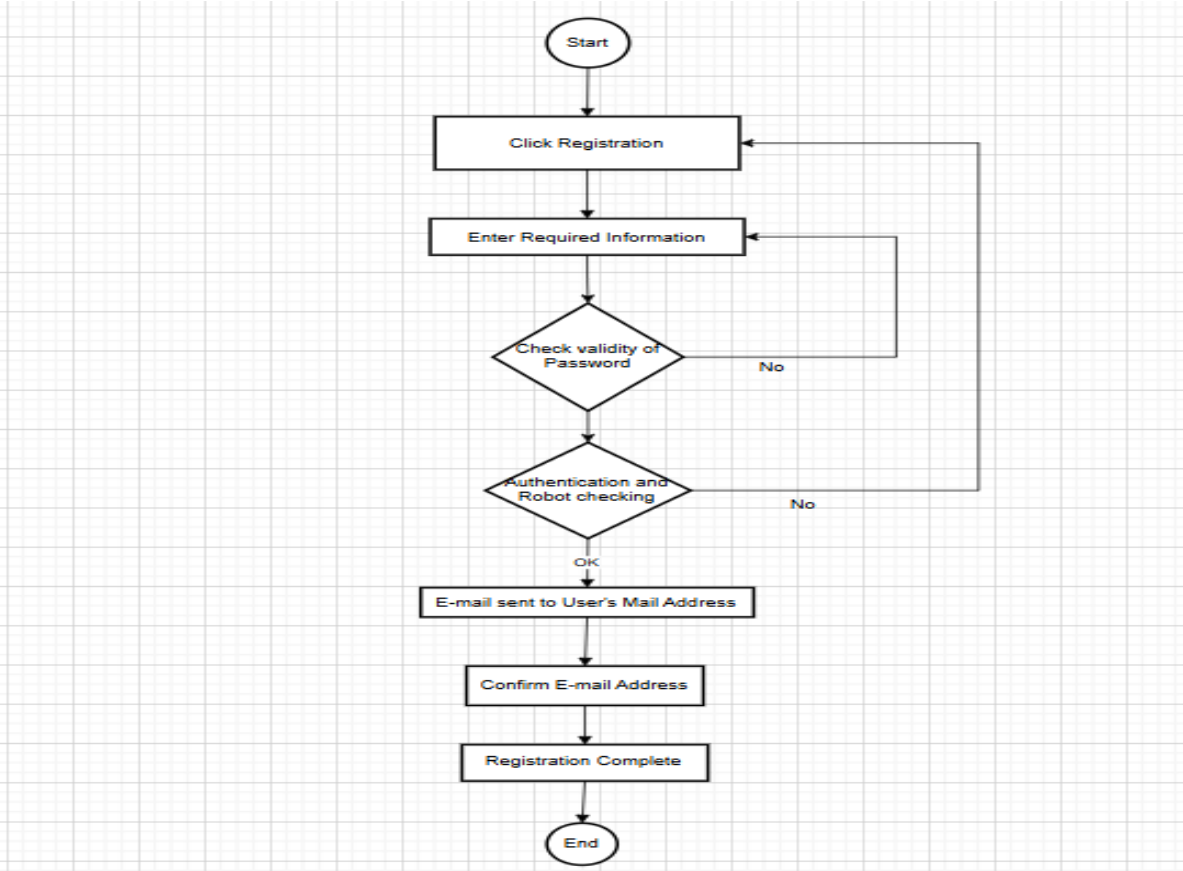


Fig : Registration System

This activity diagram depicts the user registration workflow, starting from initiating registration and entering required details.It includes password validation, authentication checks, email verification, and ends with successful account registration.

Swimlane Diagram : Registration System

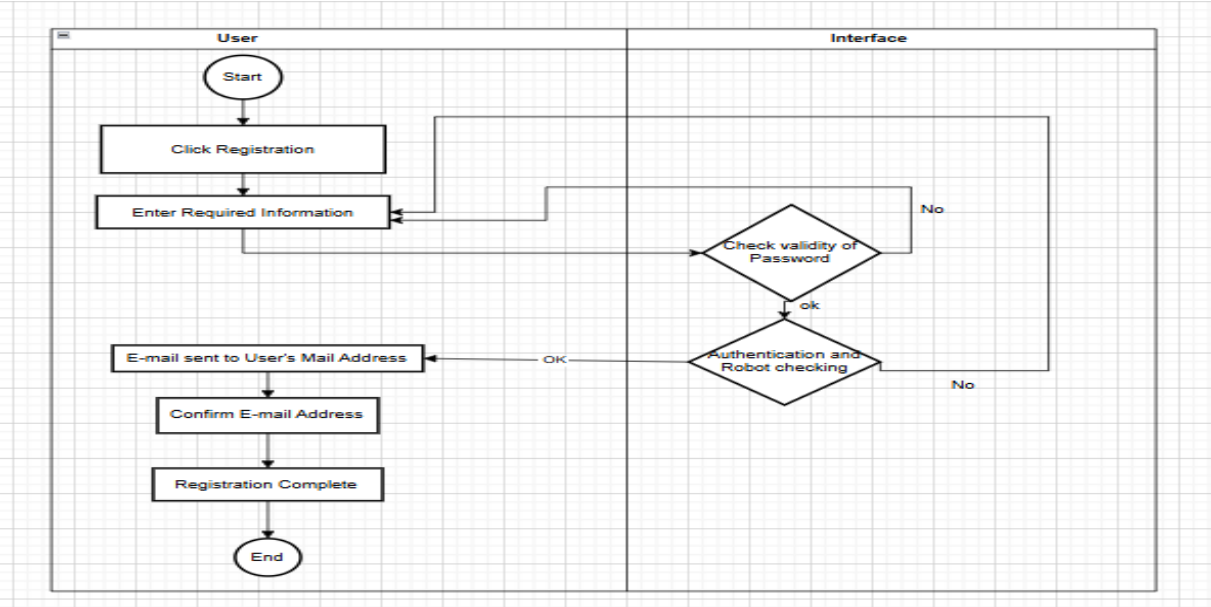


Fig: Registration System

This swimlane activity diagram shows the user registration process with interactions between the user and the system interface.It covers entering registration details, password validation, authentication checks, email confirmation, and successful account creation.

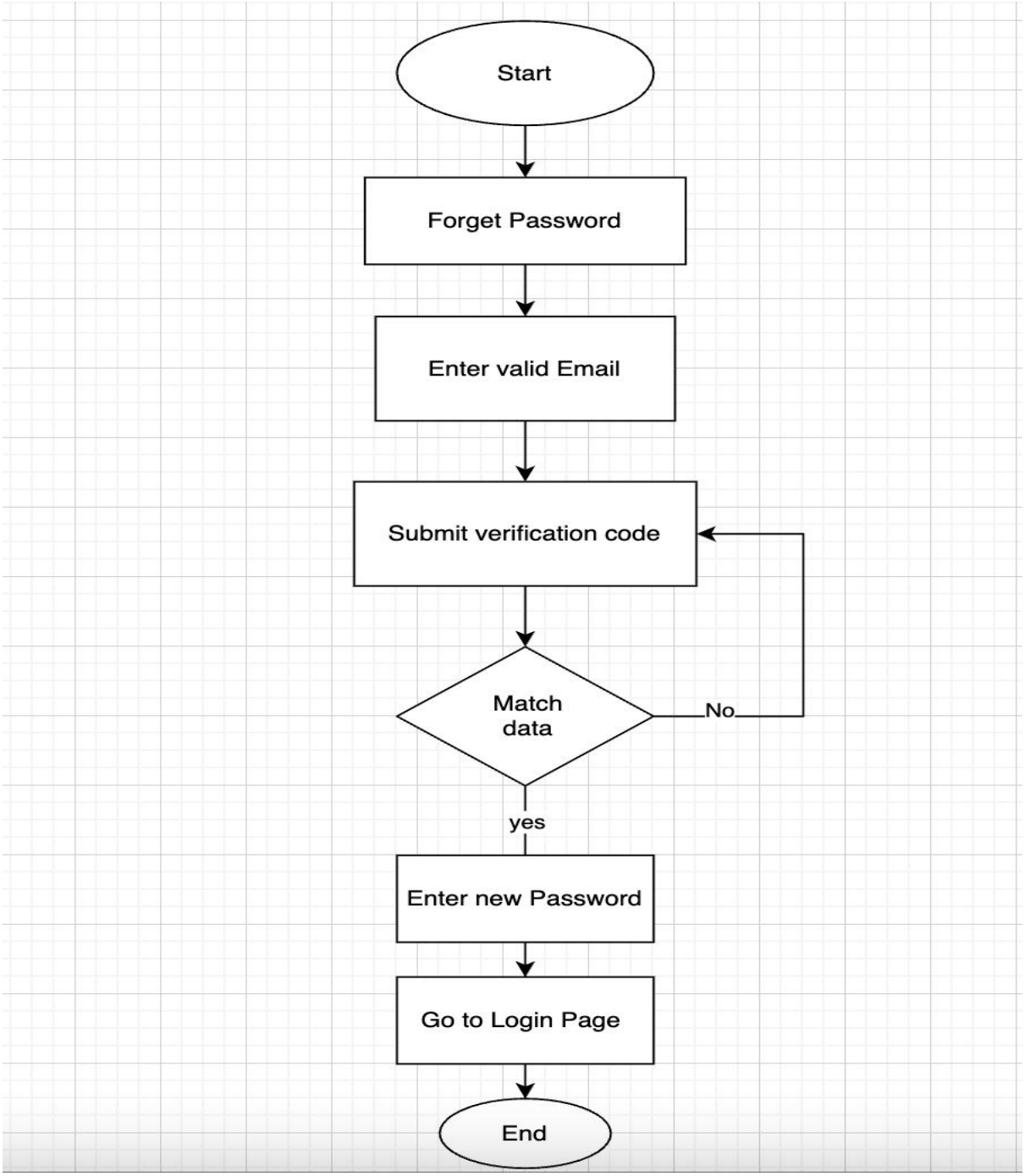
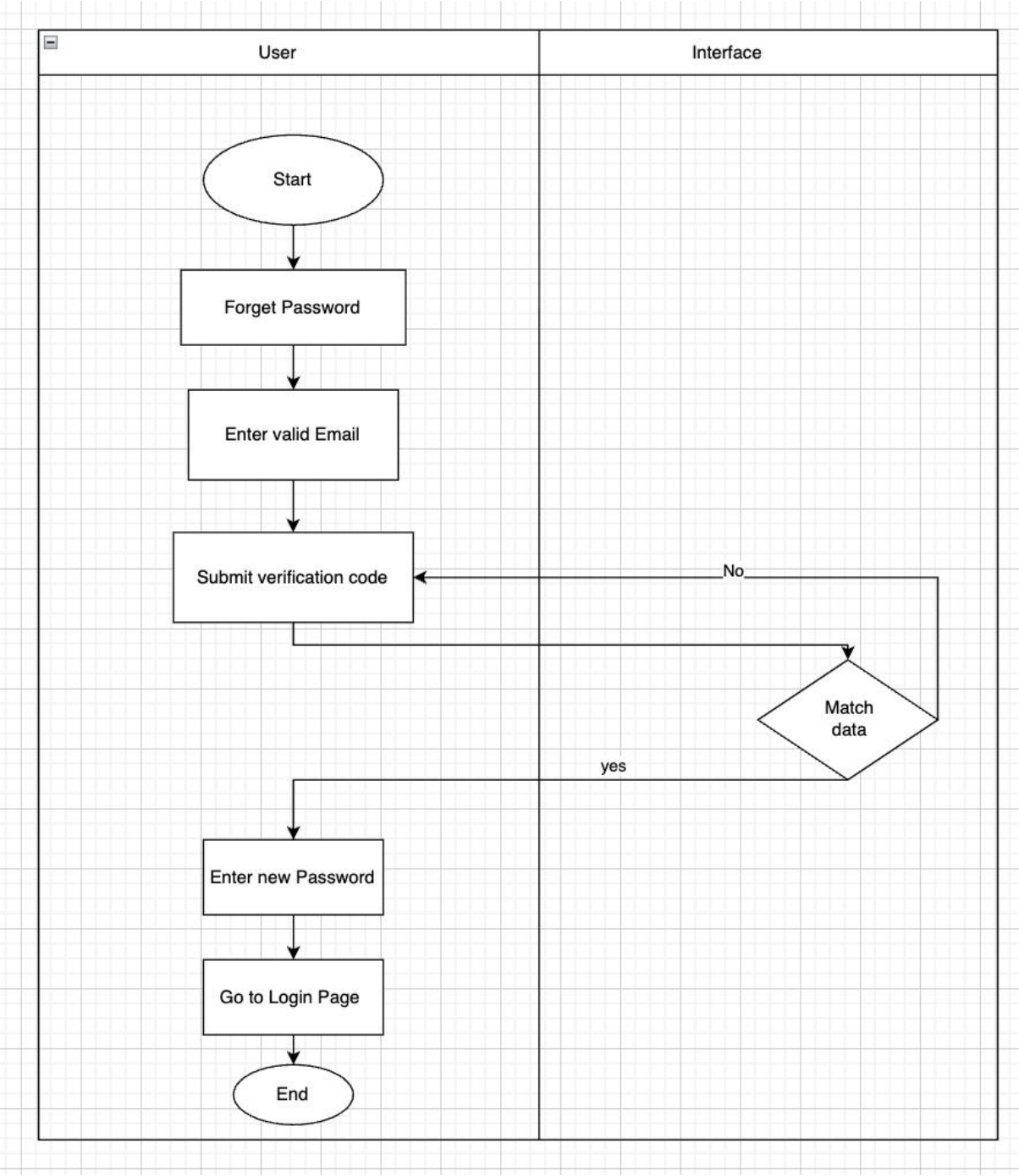


Fig : Forget Password

Swimlane Diagram: Forget Password



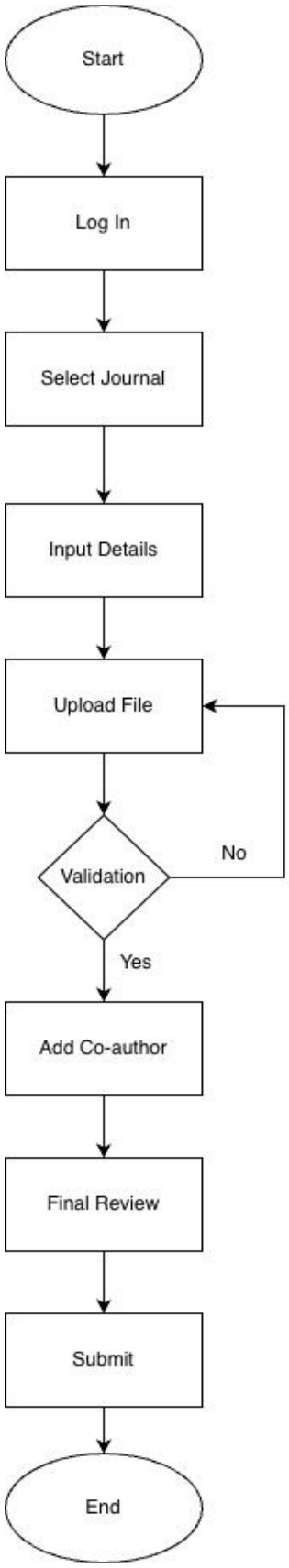
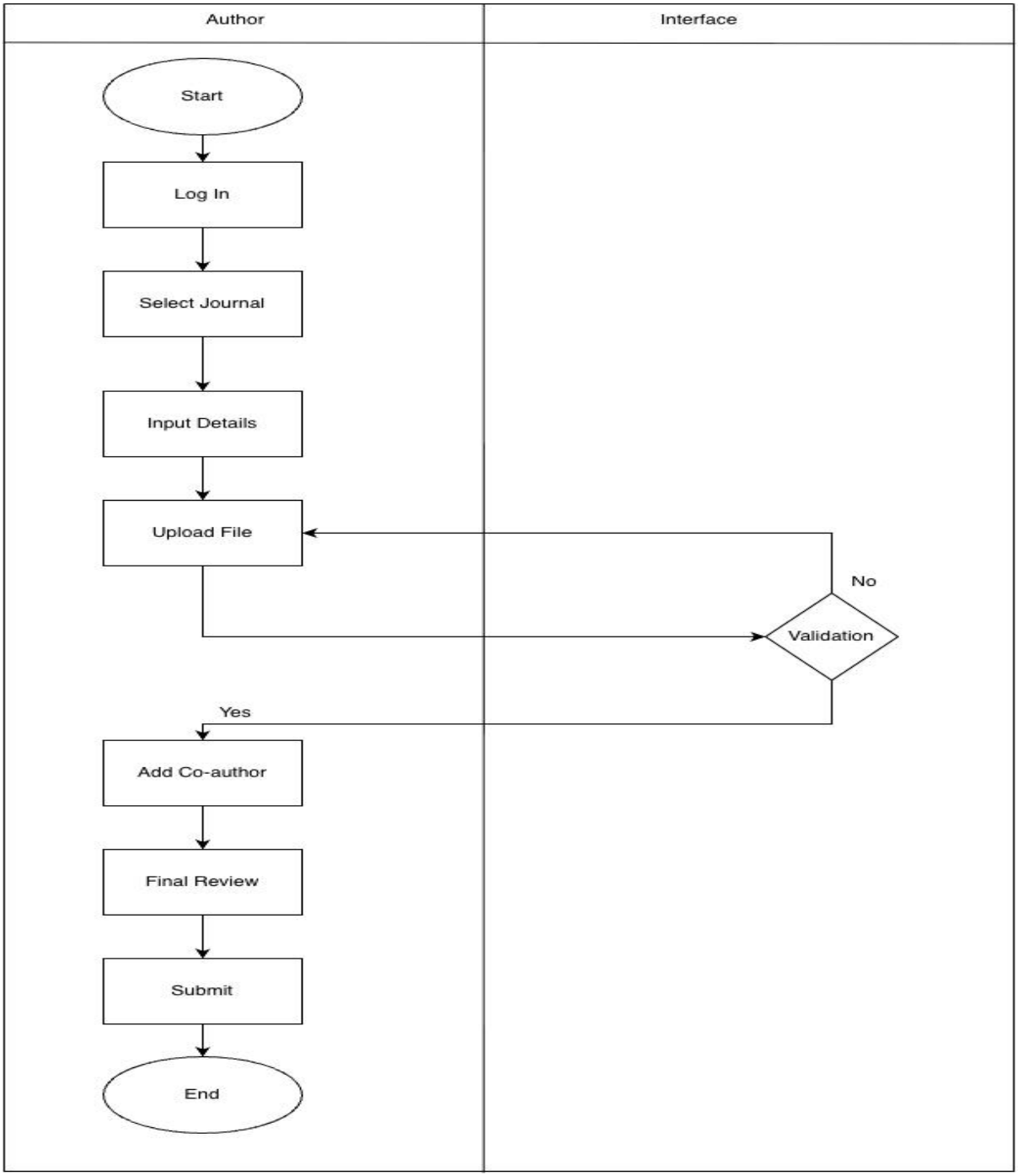


Fig : Submit New Manuscript

Swimlane Diagram : Submit New Manuscript



This swimlane activity diagram illustrates the manuscript submission process from the author's perspective and system interface interaction.It shows steps from login and journal selection to file upload, validation, final review, and successful submission.

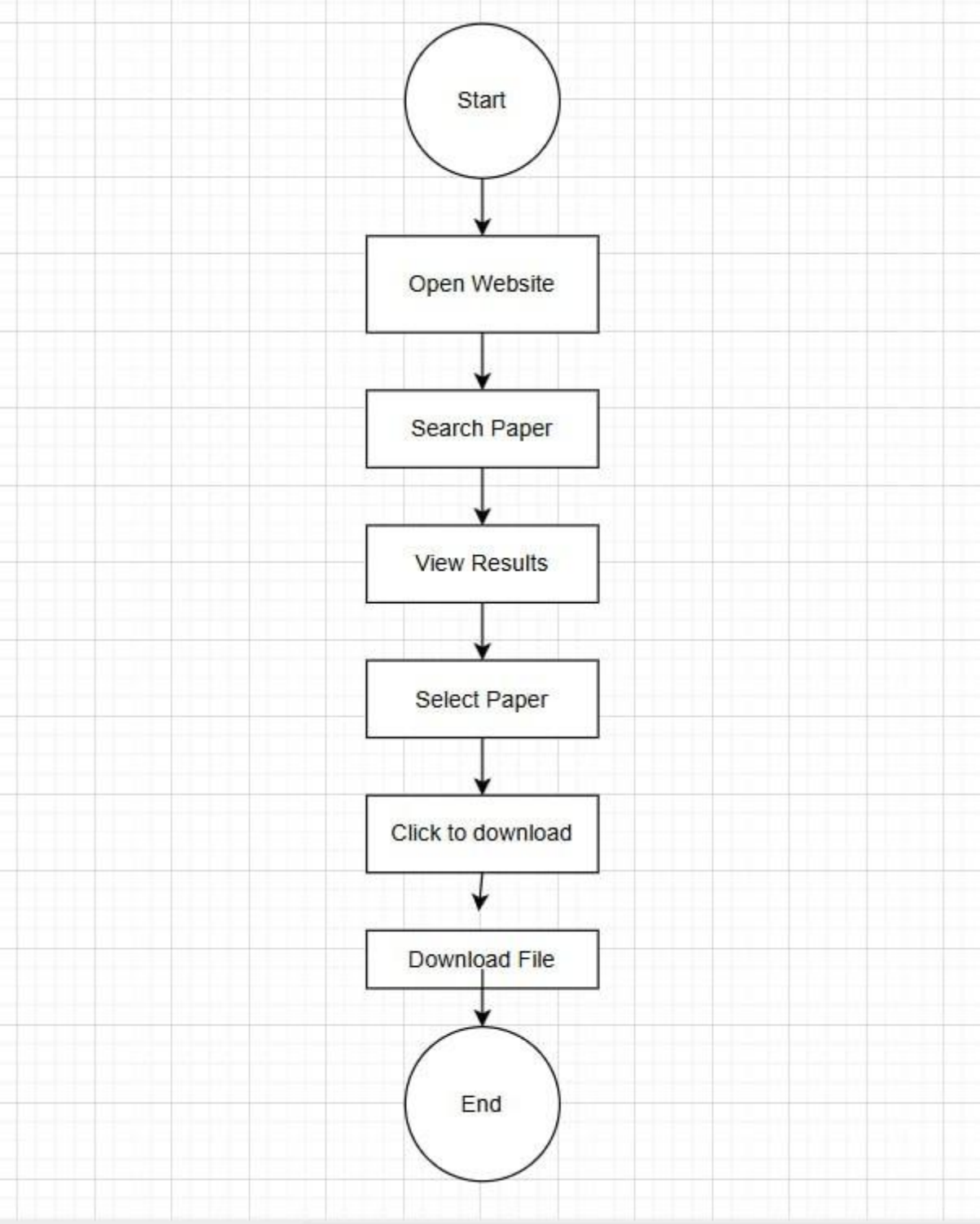
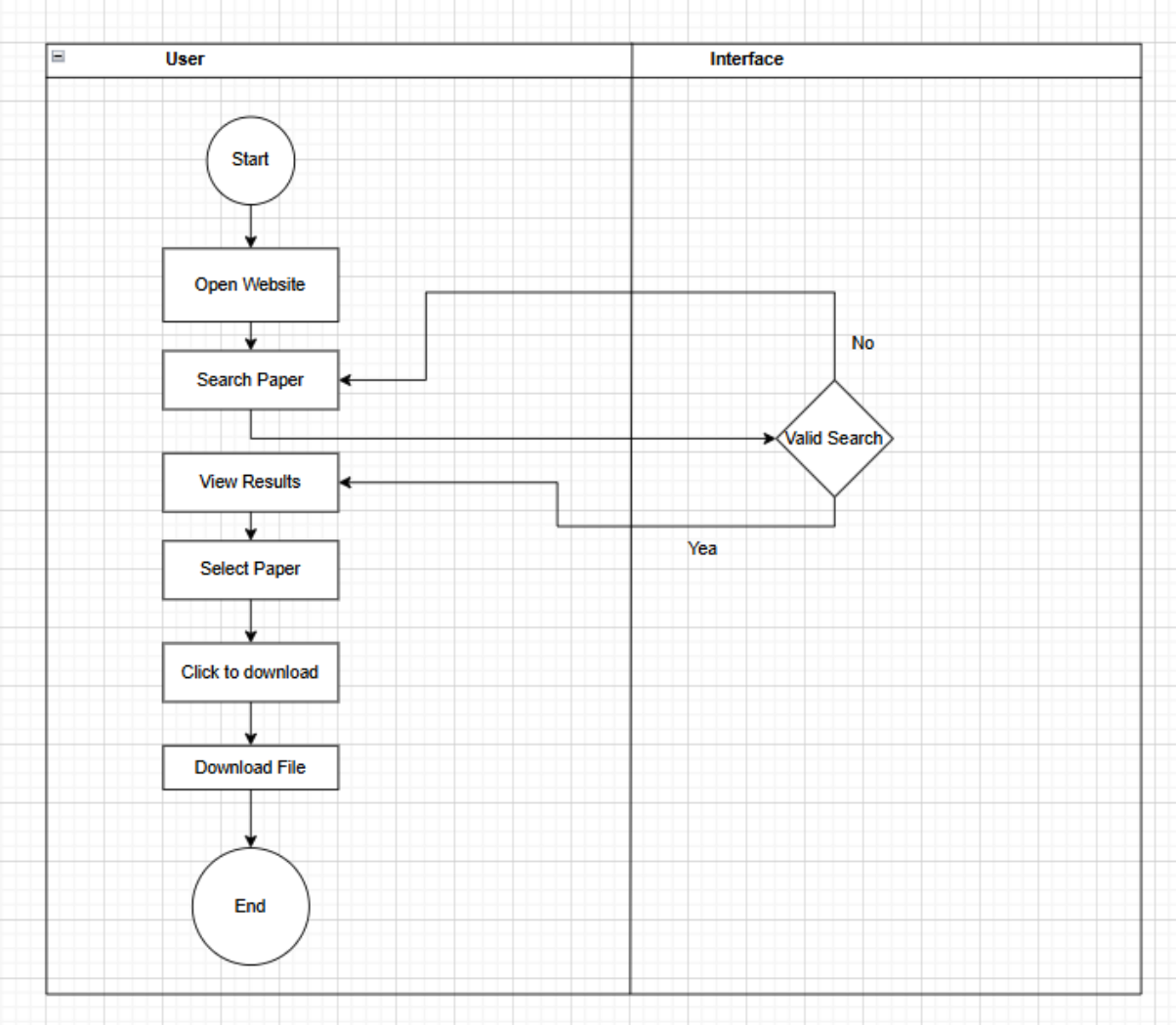


Fig : Download Paper

Swimlane Diagram : Download Paper



This swimlane activity diagram illustrates the process of searching and downloading a research paper from the system.It shows user actions and system validation, from opening the website and searching papers to selecting and downloading the file.

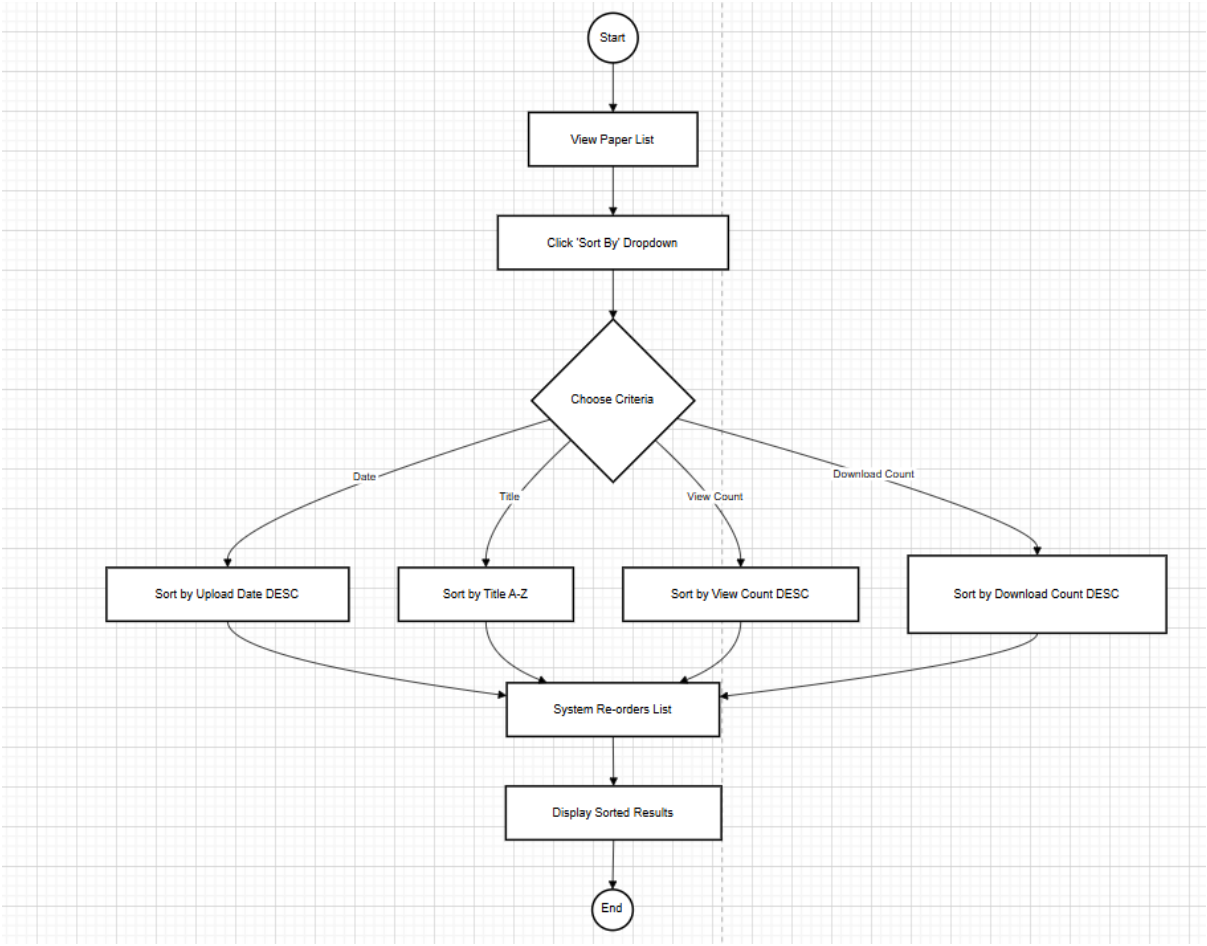
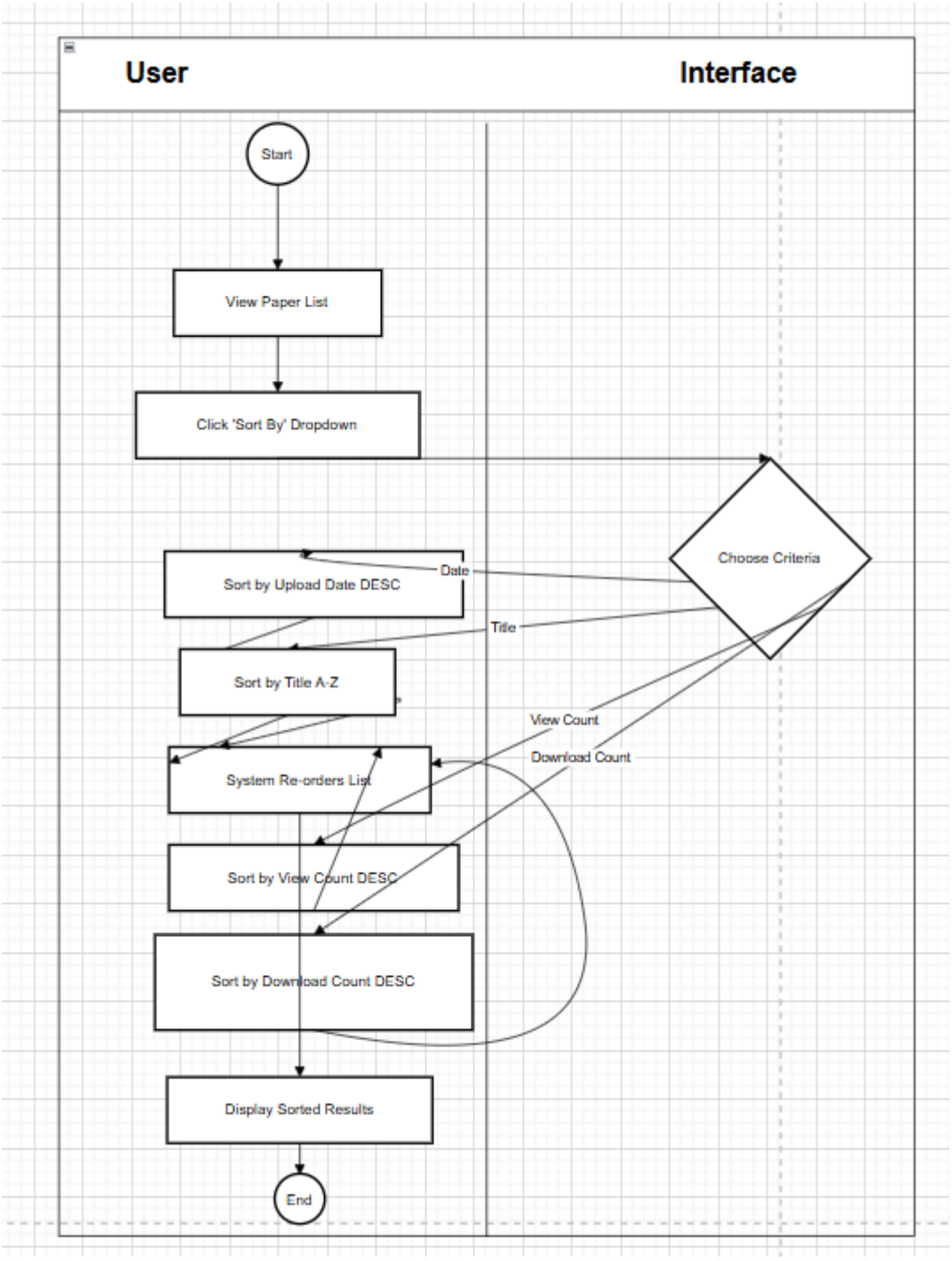


Fig : Sort paper

This activity diagram illustrates the paper sorting process where users select a sorting criterion from a dropdown menu.The system reorders the paper list based on date, title, view count, or download count and then displays the sorted results.

Swimlane Diagram : Sort Paper



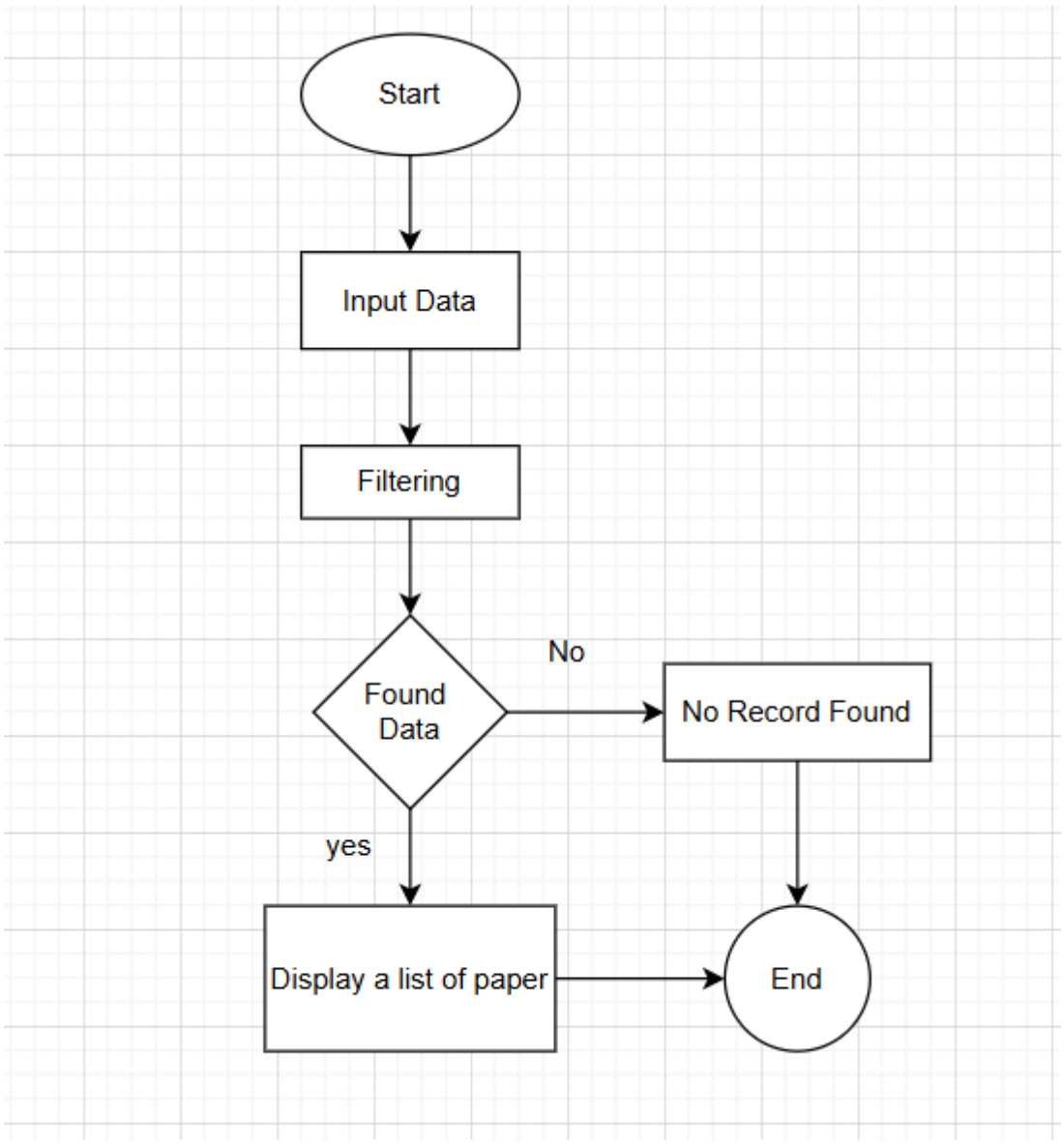
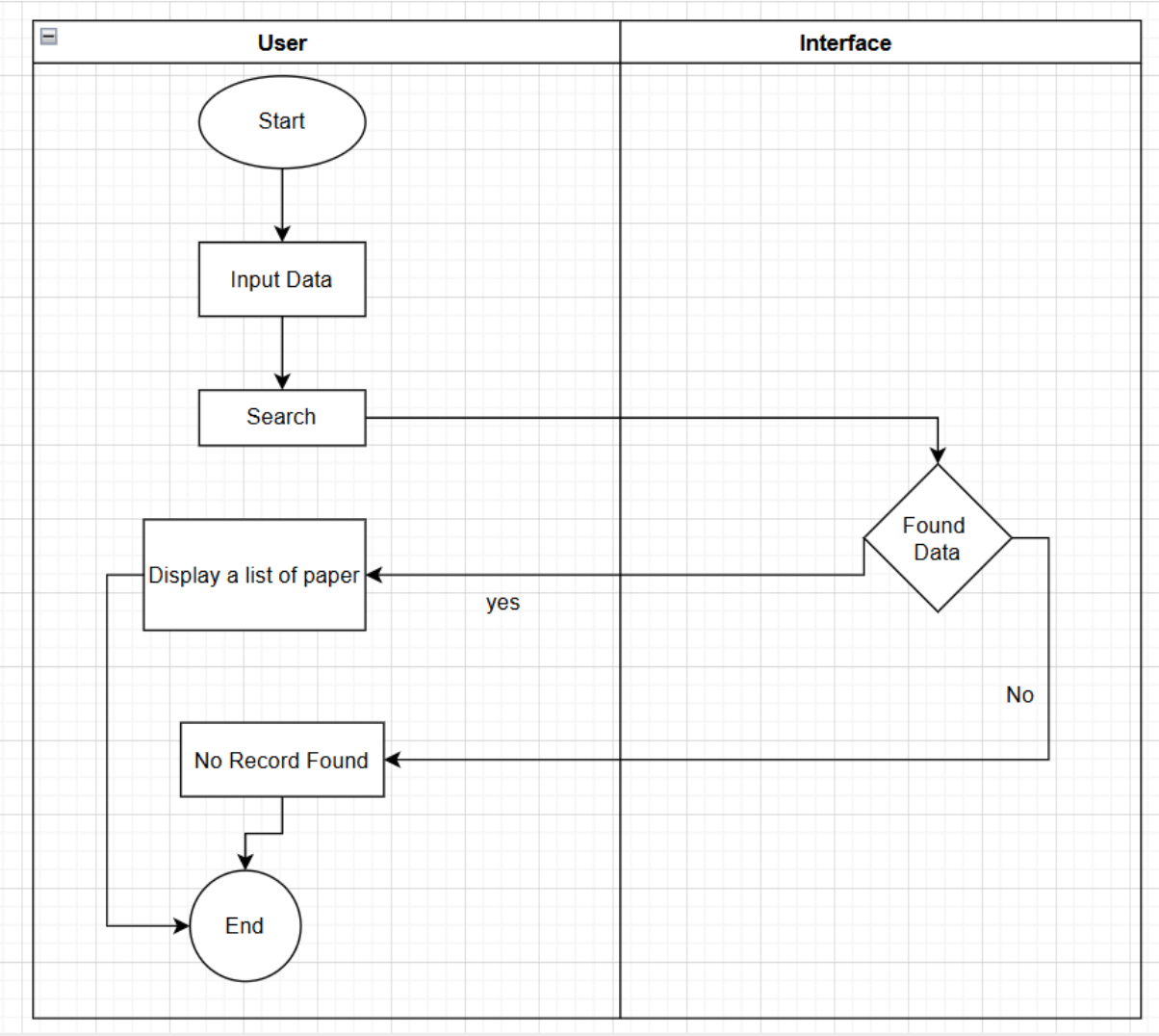


Fig : Search Paper

This activity diagram represents the data search and filtering process in the system. User input is filtered to find matching records, displaying a list of papers if found or showing a “no record found” message if not.

Swimlane Diagram : Search Paper



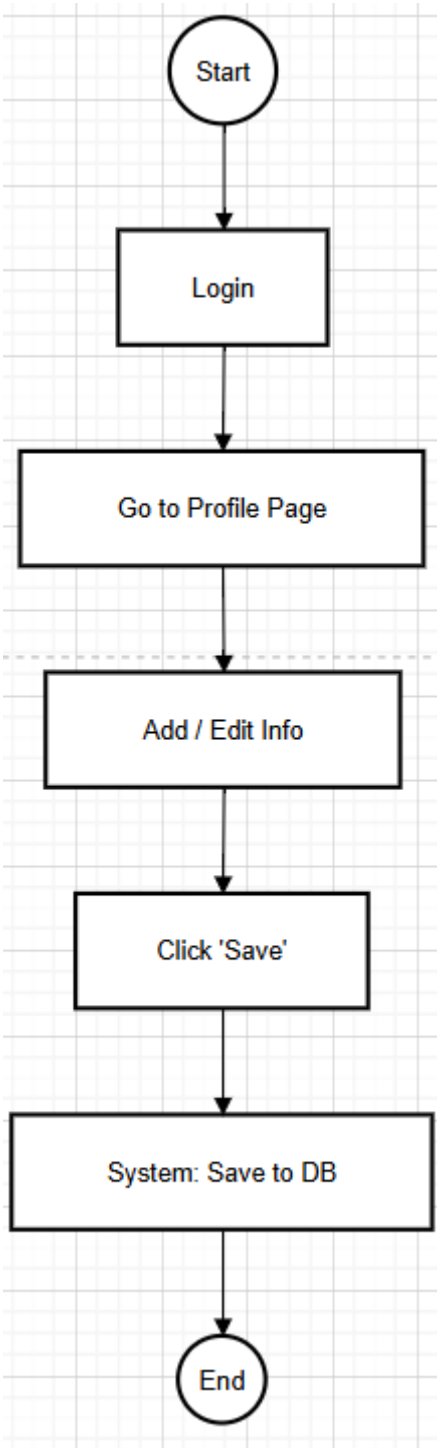


Fig : Manage User Accounts

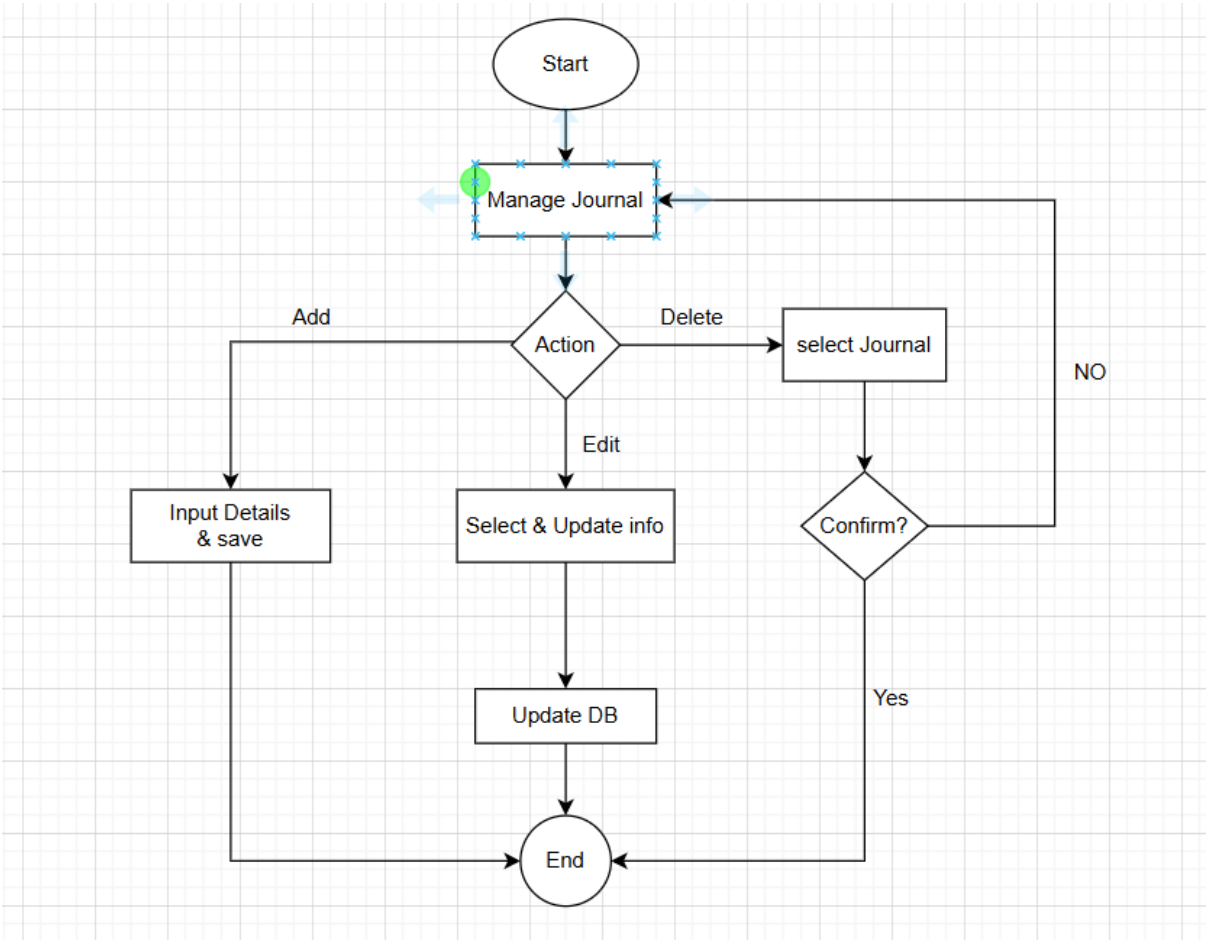
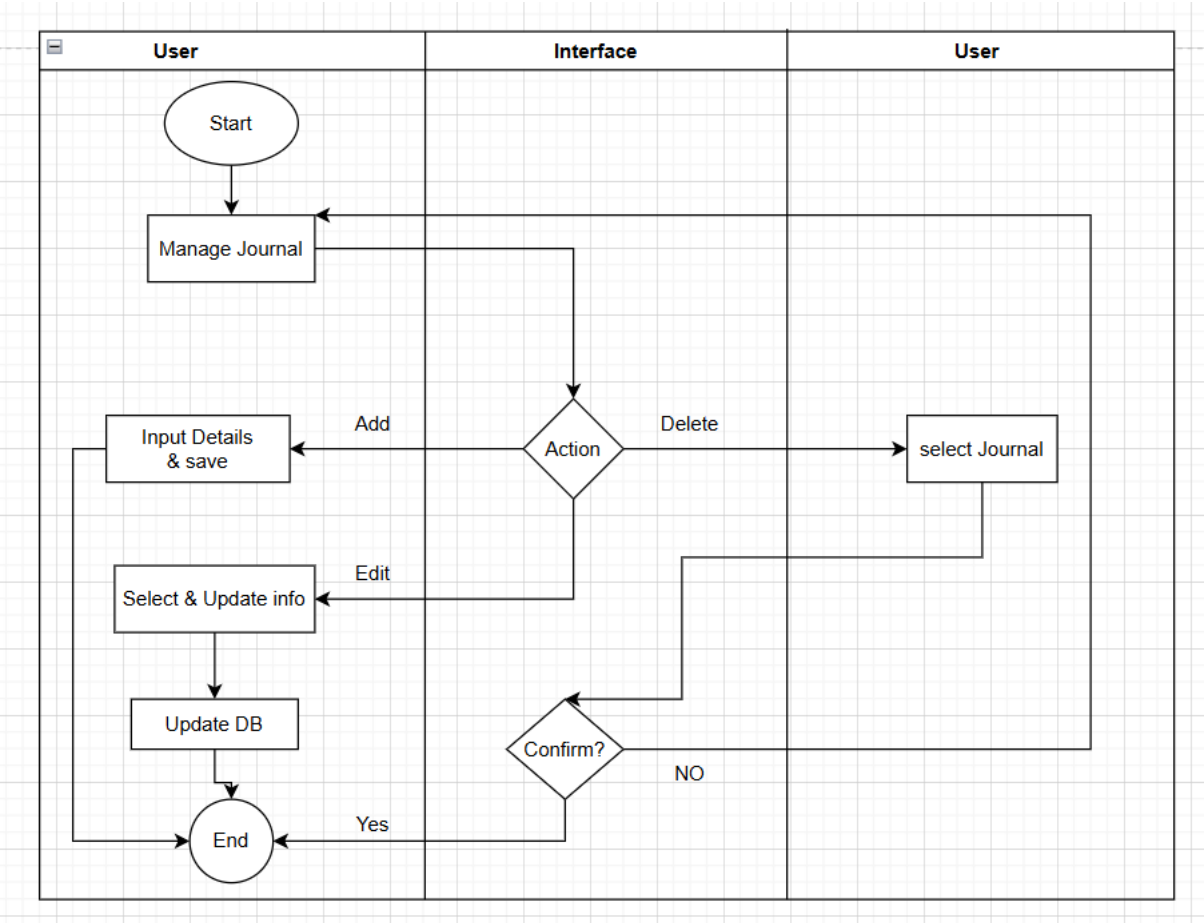


Fig : Manage Journal

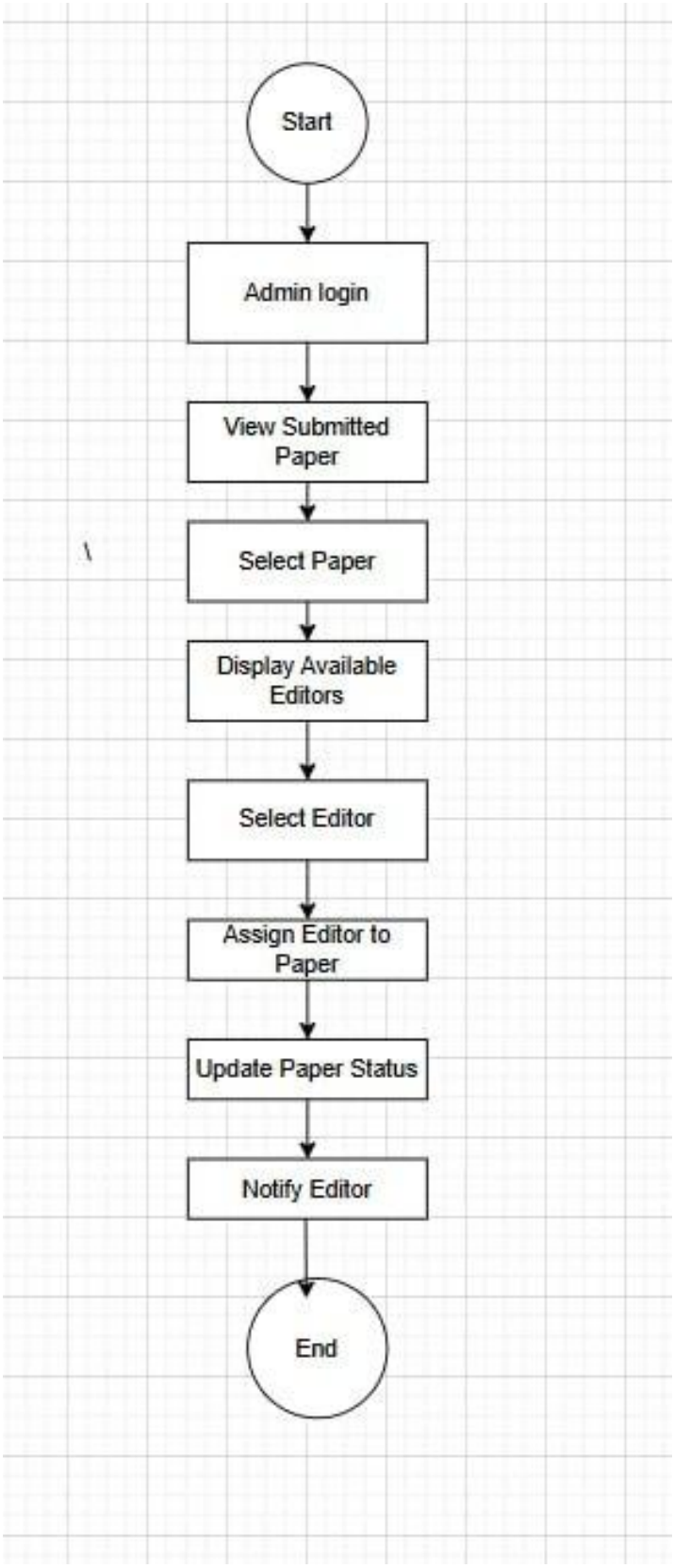
This activity diagram illustrates the journal management workflow, allowing users to add, edit, or delete journals.Based on the selected action, the system saves new details, updates existing records, or confirms and removes journals before completing the process.

Swimlane Diagram: Manage Journal

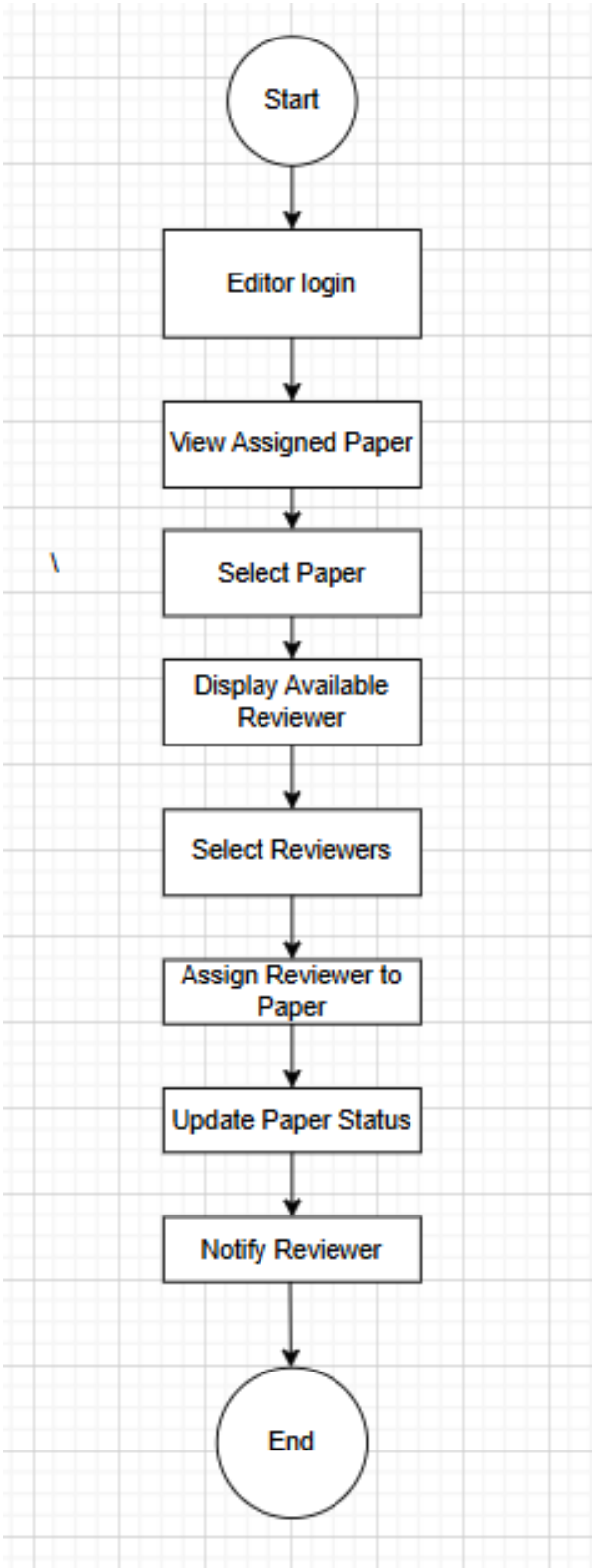


This swimlane activity diagram illustrates the journal management process, including adding, editing, and deleting journals.It shows user actions, system decisions, and database updates required to confirm and complete journal management tasks.

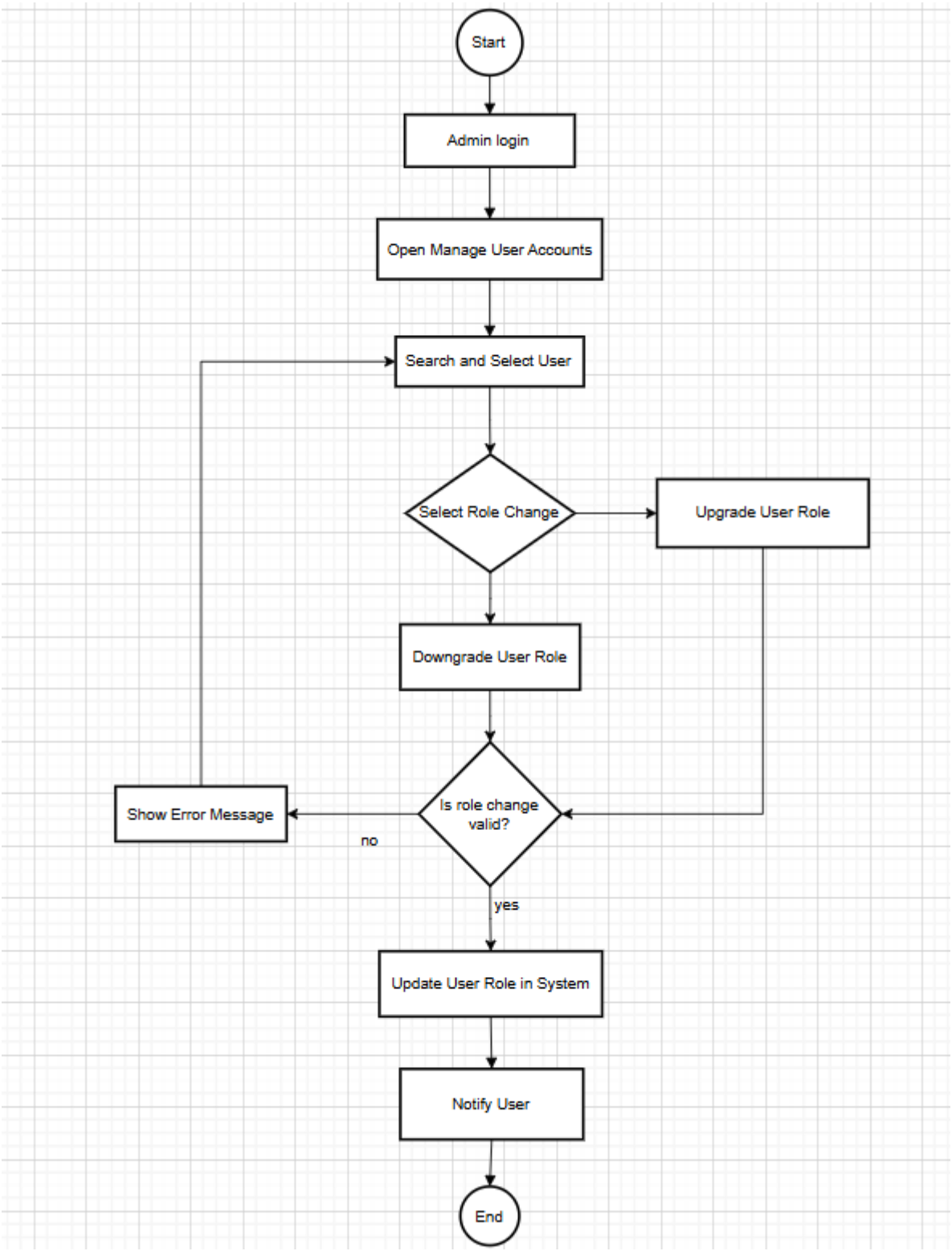
Activity Diagram: Assign Editor for Paper



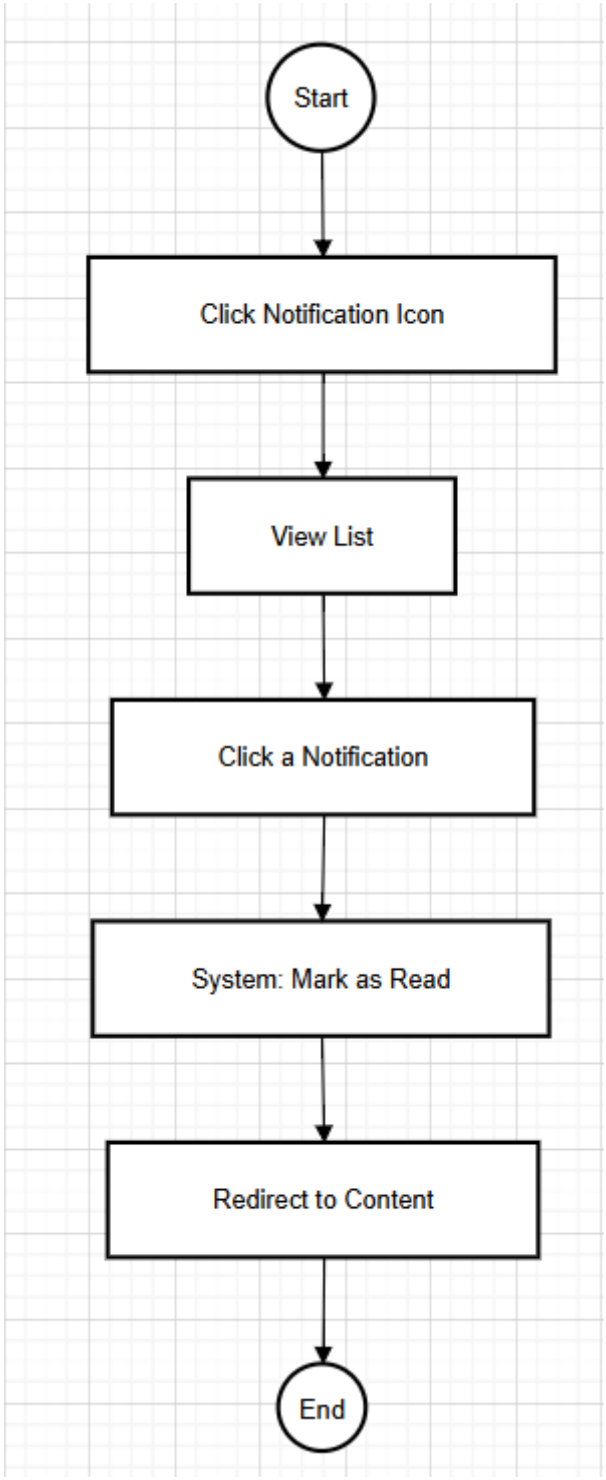
Activity Diagram: Assign Reviewer



Activity Diagram : Role Upgrade / Downgrade System



Activity Diagram : View Notification



Data Based Modeling

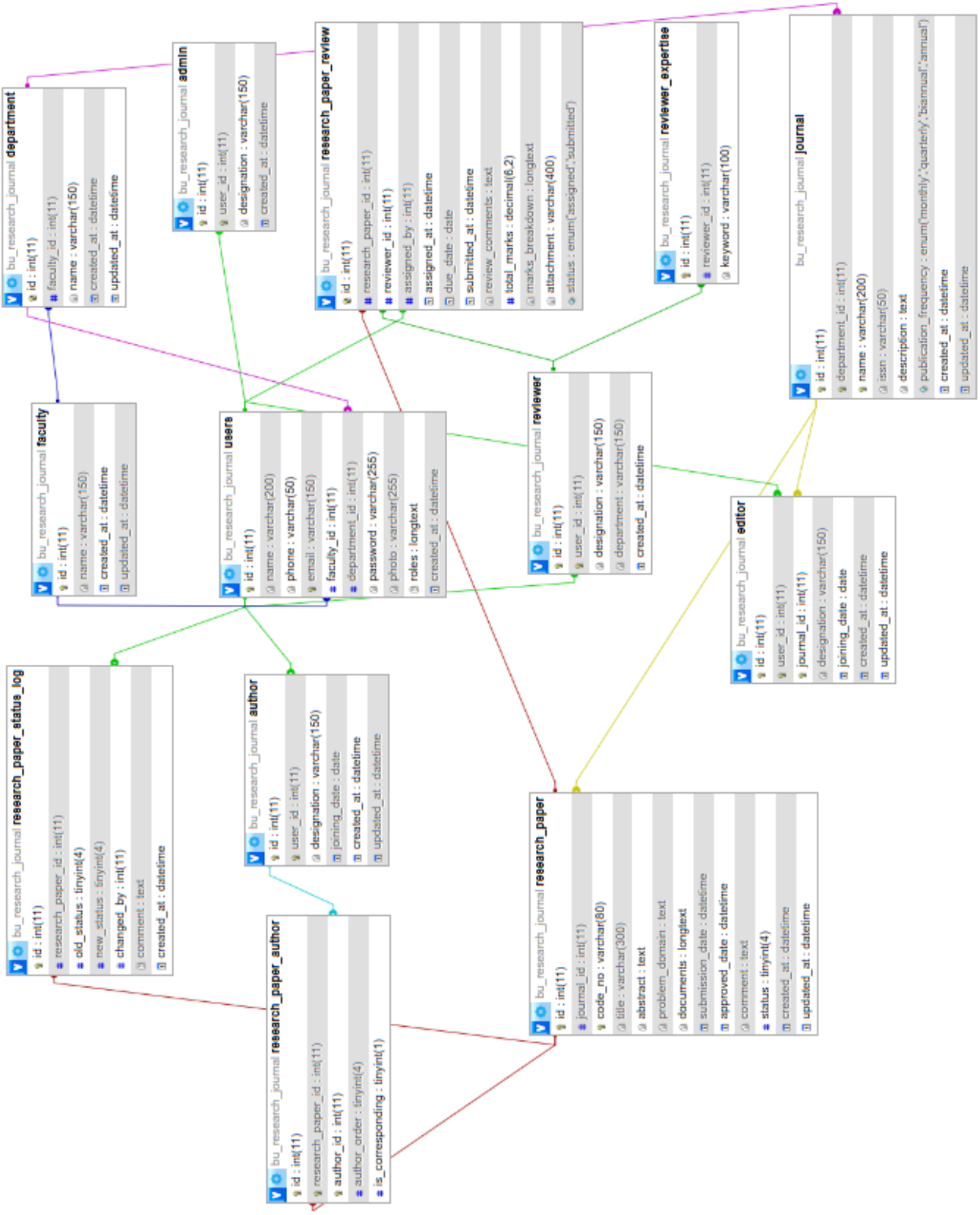
Sl. No	Noun	P/S	Attributes (Ref No.)
1	User	S	1,2,3,4,5,6,7,8,9,11,71
2	Faculty	S	1,2,11,12
3	Department	S	1,2,8,11,12
4	Journal	S	1,2,17,18,19,20,11,12
5	Sign Up / Registration	P	—
6	Sign In / Login	P	—
7	Sign Out	P	—
8	Authentication	P	—
9	Verify Email	P	13,14
10	Forgot Password	P	15,16
11	Session	S	70,53,71
12	Research Paper	S	1,17,22,23,24,25,26,27,28,29,30,31, 32,33,11,12
13	Paper Submission	P	—
14	Upload Manuscript	P	—
15	Revision	S	35,26,31,27,29,30
16	Paper Status Update	P	—
17	Paper Status Log (Audit)	S	1,35,49,50,51,30,11
18	Author	S	1,53,10,21,11,12
19	Reviewer	S	1,53,10,2,11
20	Editor	S	1,53,17,10,21,11,12
21	Admin	S	1,53,10,11
22	Paper-Author Mapping	S	1,35,36,37,38
23	Reviewer Expertise	S	1,39,34

24	Review Assignment	P	—
25	Paper Review	S	1,35,39,40,41,42,43,44,45,46,47,48, 29
26	Search Papers	P	—
27	Sort Papers	P	—
28	Show Paper Details	P	32
29	Download Paper	P	33
30	Favorite	S	52,53,35,11
31	Notification	S	1,53,54,55,56,57,11
32	Manage Journals	P	—
33	Manage User Accounts	P	—
34	Role Change	S	63,53,64,65,66,11
35	Assign Editor	P	—
36	Assign Reviewer	P	—
37	Set Review Deadline	P	42
38	Submit Verdict	P	29
39	Publish Paper	S	35,67,68,69,29
40	Submit Report Form	S	58,53,59,60,61,62,11
41	Generate System Reports	P	—

Final Data Objects

1. **User**
2. **Faculty**
3. **Department**
4. **Journal**
5. **Session**
6. **Author**
7. **Reviewer**
8. **Editor**
9. **Admin**
10. **ResearchPaper**
11. **PaperRevision**
12. **ResearchPaperStatusLog**
13. **ResearchPaperAuthor (Paper–Author Mapping)**
14. **ReviewerExpertise**
15. **ResearchPaperReview**
16. **Favorite**
17. **Notification**
18. **RoleChangeLog**
19. **Publication (PublishedPaper / IssueInfo)**
20. **ReportForm**

Schema Diagram



Class Based Modeling

List of Nouns in the Solution Space and Their General Classification :

Sl. No	Noun	General Classification
1	User	4, 5, 7
2	Faculty	5, 7
3	Department	5, 7
4	Journal	5, 7
5	Research Paper	2, 3, 7
6	Document	2, 7
7	Review	3, 7
8	Status	3, 7
9	Reviewer	4, 5, 7
10	Editor	4, 5, 7
11	Author	4, 5, 7
12	Admin	4, 5, 7
13	Status Log	3, 7

Potential Classes :

Sl. No	Potential Class
1	User
2	Faculty
3	Department
4	Journal
5	ResearchPaper
6	ResearchPaperReview
7	ResearchPaperStatusLog

Selection Criteria

1. Retain information
2. Needed services
3. Multiple attributes
4. Common attributes
5. Common operations
6. Essential requirements

Potential Class Selection Criteria Table :

Class Name	Selection Criteria Matched
User	1, 2, 3, 4, 5, 6
Faculty	1, 3, 6
Department	1, 3, 6
Journal	1, 2, 3, 6
ResearchPaper	1, 2, 3, 4, 5, 6
ResearchPaperReview	1, 2, 3, 4, 5, 6
ResearchPaperStatusLog	1, 2, 3, 6

Final Selected Classes :

Sl. No	Class Name
1	User
2	Faculty
3	Department
4	Journal

5	ResearchPaper
6	ResearchPaperReview
7	ResearchPaperStatusLog

Database :

Attributes	Methods
DATABASE_URL	create_db_and_tables()
engine	get_session()
	read()
	write()
	update()
	delete()

Responsibilities	Collaborators
Manage database connection and persistence	All classes

User

Attributes	Methods
id	register()
name	login()
email	logout()
phone	updateProfile()
password	changePassword()
photo	hasRole()
roles (JSON)	
faculty_id	
department_id	
created_at	

Responsibilities	Collaborators
User authentication and authorization	Faculty
Role-based access (Admin/Editor/Author/Reviewer)	Department
Profile management	Journal
Perform submission, review, decision tasks	ResearchPaper

Faculty :

Attributes	Methods
id	addFaculty()
name	updateFaculty()
created_at	deleteFaculty()

updated_at	viewDepartments()

Responsibilities	Collaborators
Maintain faculty information	Department

Department :

Attributes	Methods
id	addDepartment()
faculty_id	updateDepartment()
name	deleteDepartment()
created_at	viewJournals()
updated_at	

Responsibilities	Collaborators
Maintain department information	Faculty
Organize journals	Journal

Journal :

Attributes	Methods
id	createJournal()
department_id	updateJournal()
name	deleteJournal()

issn	listPapers()
description	
publication_frequency	
created_at	
updated_at	

Responsibilities	Collaborators
Maintain journal profile	Department
Provide container for research papers	ResearchPaper

ResearchPaper :

Attributes	Methods
id	submitPaper()
journal_id	uploadDocuments()
code_no	updatePaper()
title	changeStatus()
abstract	viewPaperDetails()
problem_domain	
documents (JSON)	
submission_date	
approved_date	
comment	
status	
created_at	

updated_at	

Responsibilities	Collaborators
Store paper metadata and documents	User
Maintain paper lifecycle	Journal
Provide paper for review	ResearchPaperReview
Track status changes	ResearchPaperStatusLog

ResearchPaperReview :

Attributes	Methods
id	assignReviewer()
research_paper_id	setDeadline()
reviewer_id	submitReview()
assigned_by	updateMarks()
assigned_at	
due_date	
submitted_at	
review_comments	
total_marks	
marks_breakdown	
attachment	
status	

Responsibilities	Collaborators
Manage review assignment and feedback	User
Store review results	ResearchPaper
Support editorial decisions	ResearchPaperStatusLog

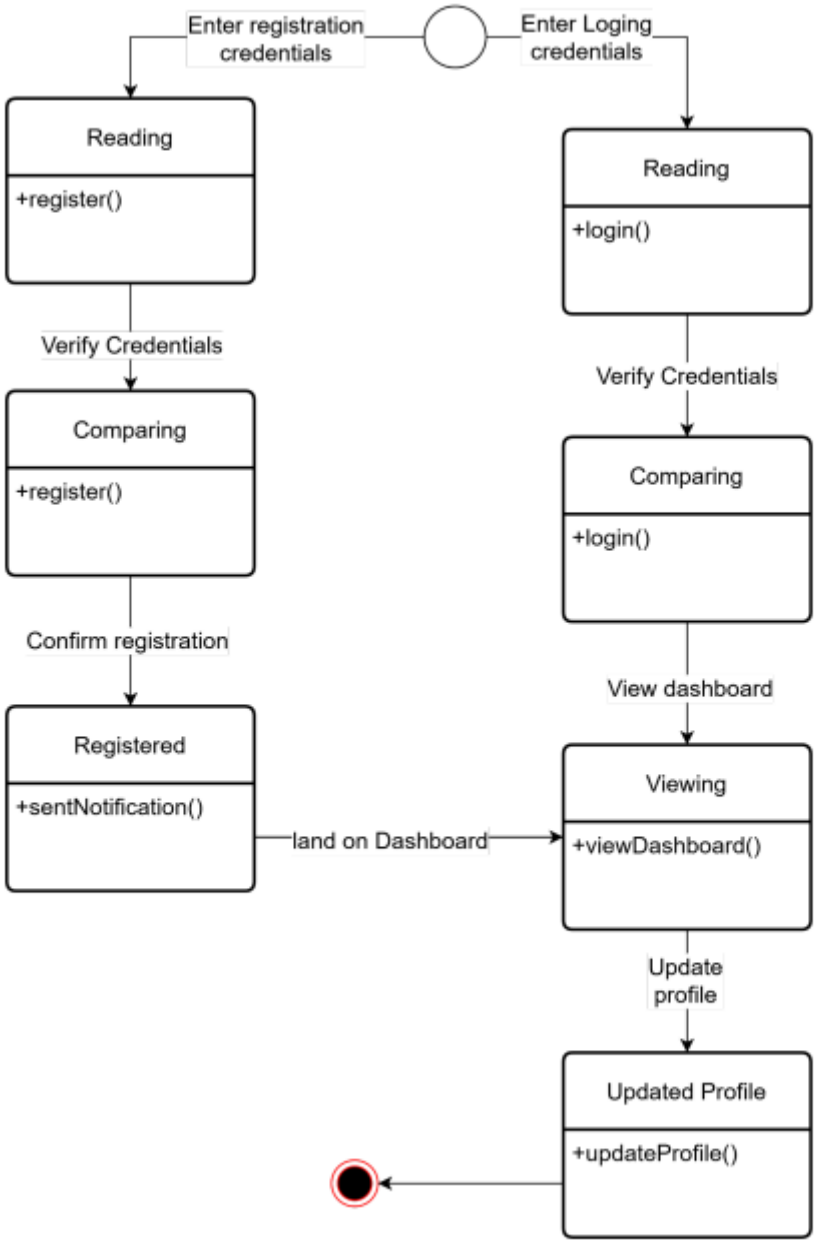
ResearchPaperStatusLog :

Attributes	Methods
id	logStatusChange()
research_paper_id	getPaperHistory()
old_status	
new_status	
changed_by	
comment	
created_at	

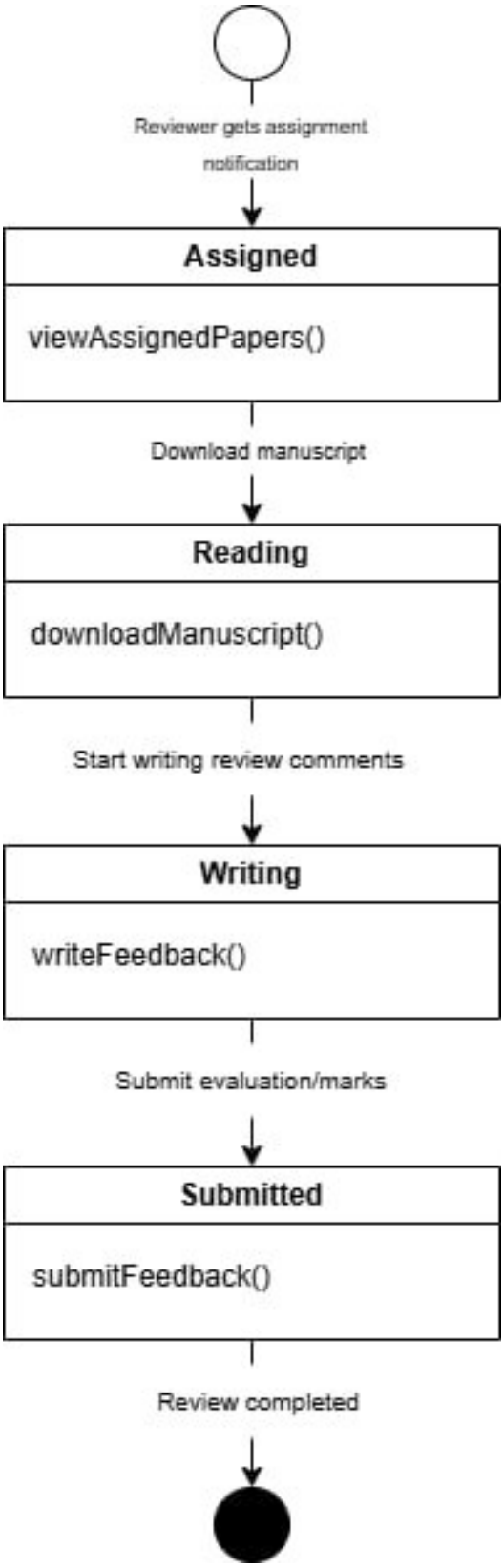
Responsibilities	Collaborators
Maintain audit history of paper status	ResearchPaper
Support tracking and transparency	User

State Transition Diagram :

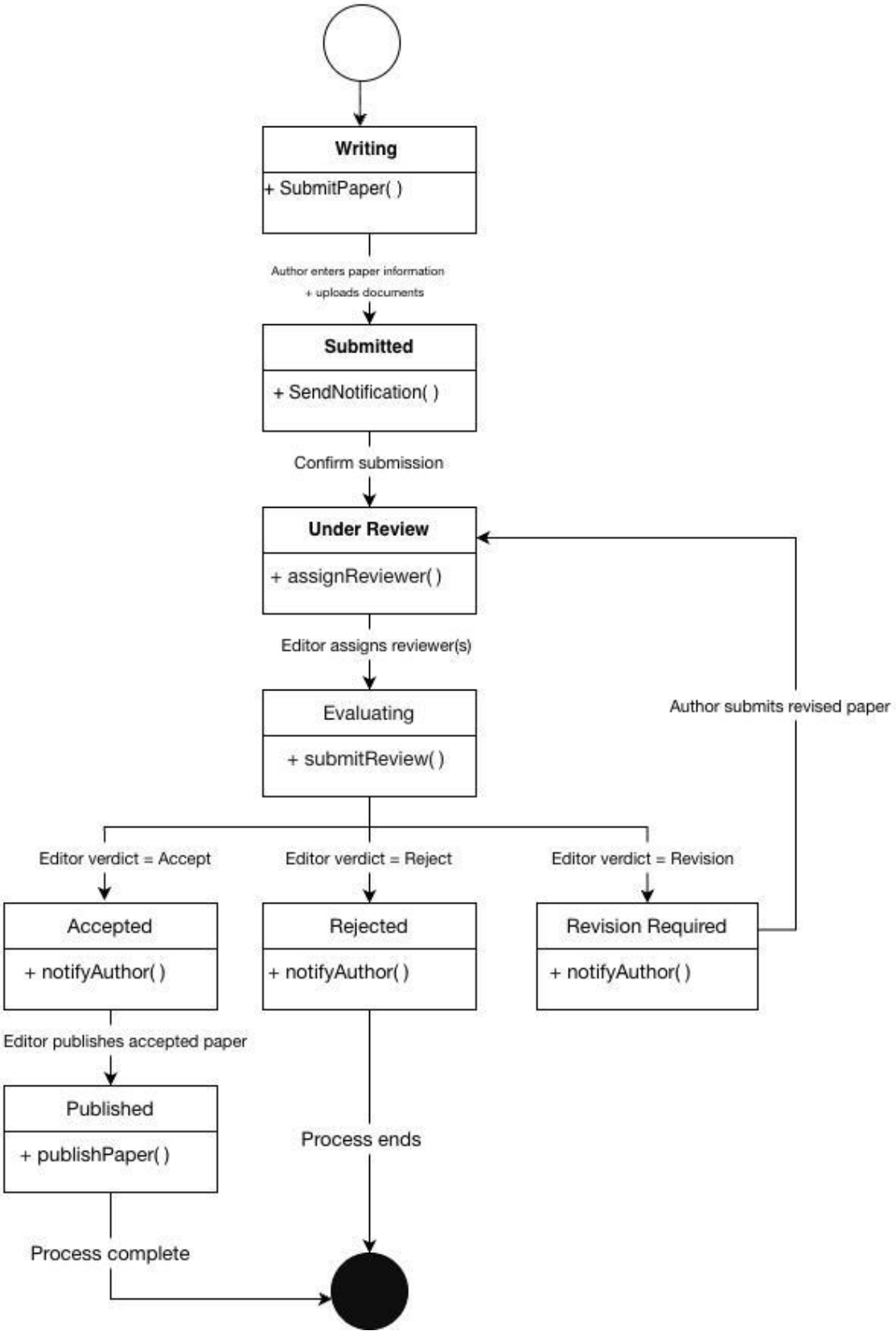
Name : User



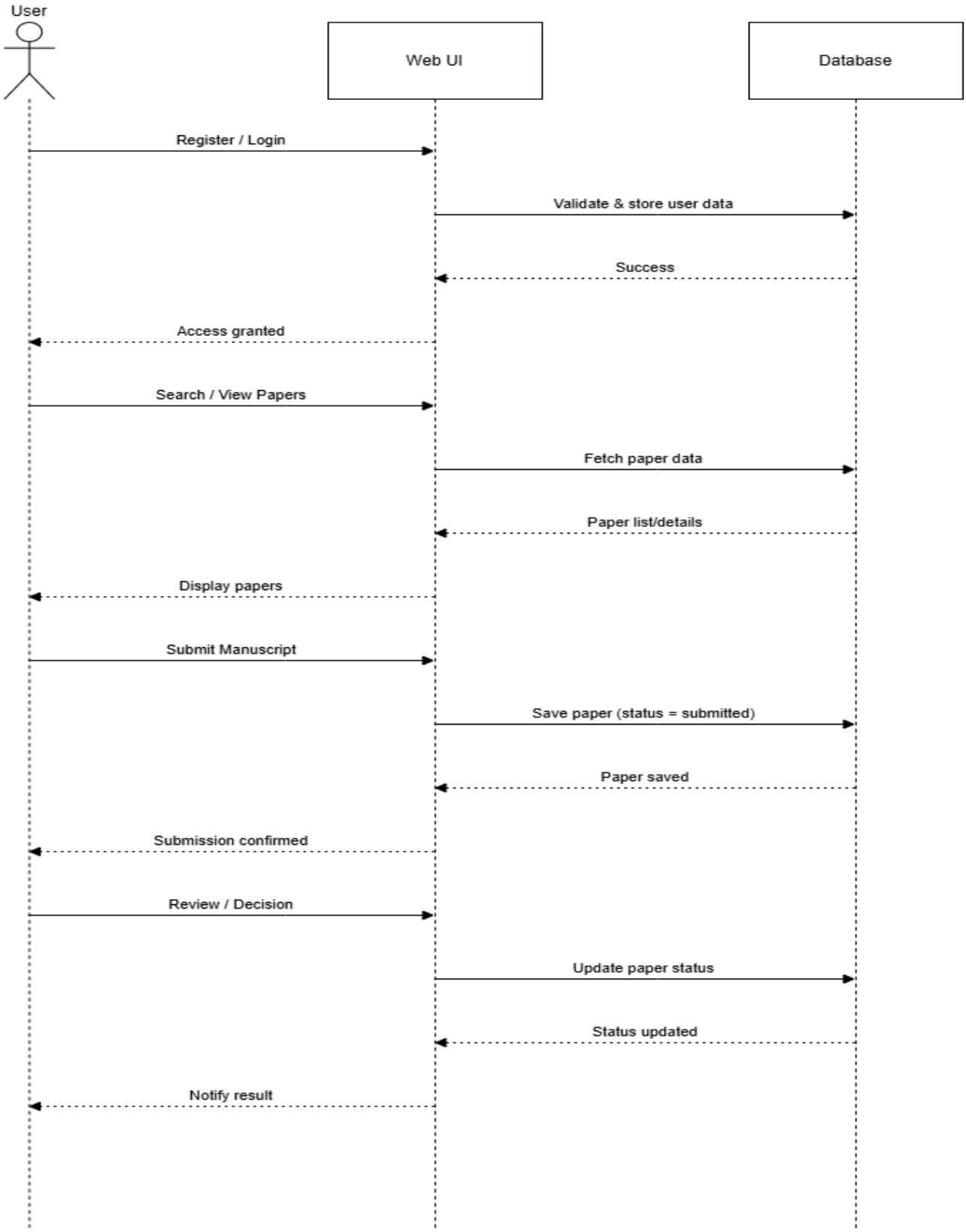
Name: Review Process

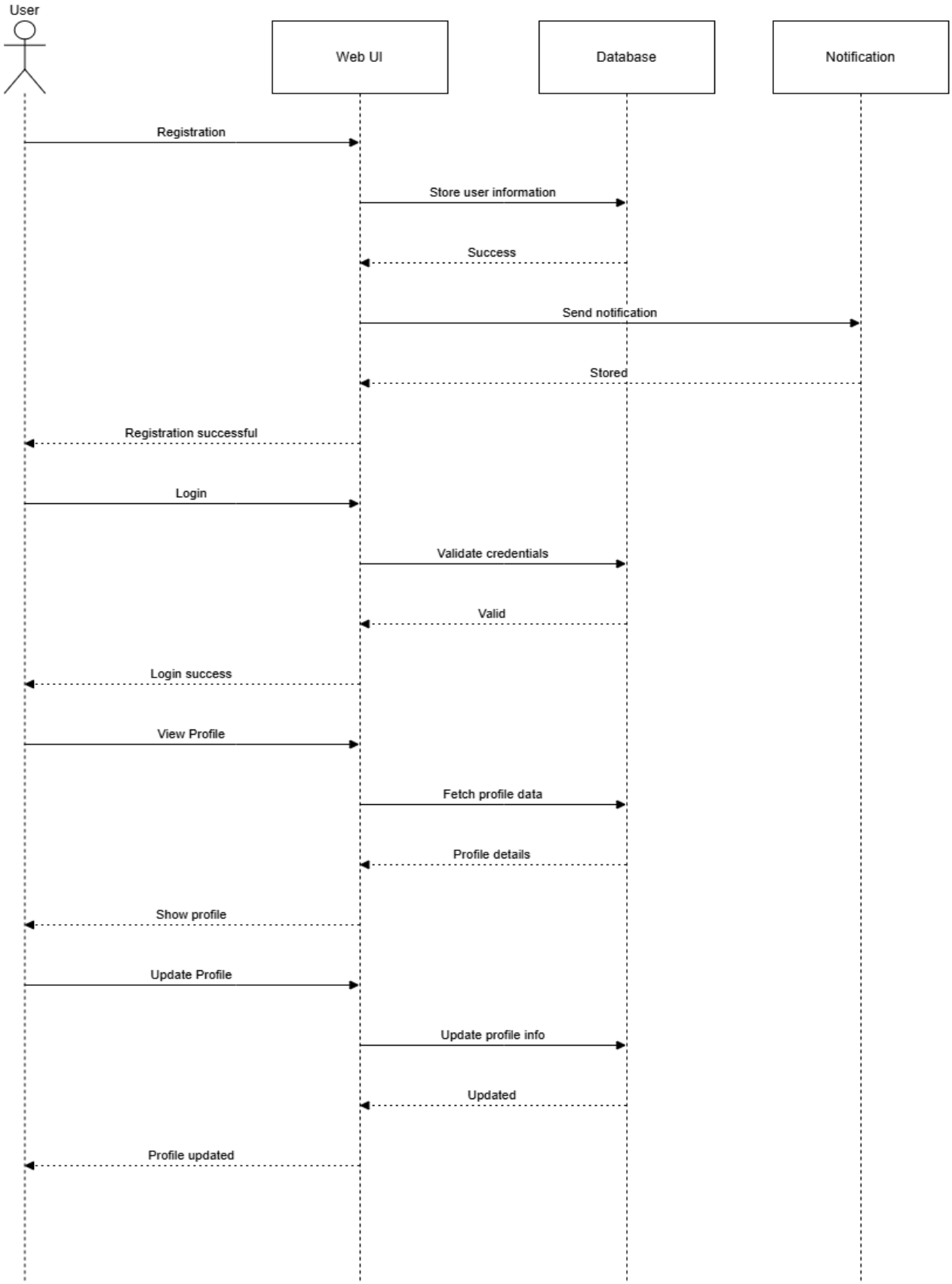


Name: Research Paper (Submission to Publication)



Sequence Diagram




User
Web UI
Database
Notification
Registration
Store user information
Success
Send notification
Stored
Registration successful
Login
Validate credentials
Valid
Login success
View Profile
Fetch profile data
Profile details
Show profile
Update Profile
Update profile info
Updated
Profile updated

Data Flow Diagram