

# **SRM RESEARCH HUB**

**A MINI PROJECT REPORT**

**21CSC303J**

**Software Engineering and Project Management**

*Submitted by*

**Ponnuri Aniruddha [Reg No: RA2112704010015]**

**Vamshi Gadde [Reg No: RA2112704010017]**

**Y Shabanya Kishore [Reg No: RA2112704010018]**

*Under the Guidance of*

**Dr.A.Shanthini**

(Associate Professor, Department of Data Science and Business Systems)

*In partial fulfillment of the Requirements for the Degree  
of*

**M.TECH (Integrated)**

**COMPUTER SCIENCE WITH SPECIALIZATION IN  
DATA SCIENCE**



**SRM**

INSTITUTE OF SCIENCE & TECHNOLOGY  
Deemed to be University u/s 3 of UGC Act, 1956

**DEPARTMENT OF DATA SCIENCE AND BUSINESS  
SYSTEMS**

**FACULTY OF ENGINEERING AND TECHNOLOGY**

**SRM INSTITUTE OF SCIENCE AND TECHNOLOGY**

**MAY 2024**

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY  
KATTANKULATHUR-603203

BONAFIDE CERTIFICATE

Certified that this project report titled “SRM RESEARCH HUB” is the bonafide work of “**Ponnuri Aniruddha [Reg No: RA2112704010015], Vamshi Gadde [Reg No: RA2112704010017], Y Shabanya Kishore [Reg No: RA2112704010018]** “ who carried out the project work under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion for this or any other candidate.

Dr.A.Shanthini  
**GUIDE**  
Associate Professor  
Dept. of DSBS

Dr. M.Lakshmi  
**HEAD OF THE DEPARTMENT**  
Dept. of DSBS

Signature of Internal Examiner

Signature of External Examiner

## **Abstract**

The Research Paper Repository is a web-based platform designed to facilitate the management and access of research papers. It provides users with the ability to explore and contribute to a curated collection of academic publications, with a focus on papers published by SRM Institute of Science and Technology (SRMIST) researchers. The platform aims to streamline the process of discovering, sharing, and collaborating on research within the academic community.

# TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	
	LIST OF FIGURES	
	LIST OF ABBREVIATIONS	
1	PROBLEM STATEMENT AND BUSINESS CASE	1
2	PROCESS METHODOLOGY AND STAKEHOLDERS	3
3	SYSTEM, FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS	5
4	PROJECT PLAN	6
5	WORK BREAKDOWN STRUCTURE, TIMELINE-CHART , RISK-IDENTIFICATION TABLE	10
6	SYSTEM ARCHITECTURE, USE CASE & CLASS DIAGRAM	16
7	ENTITY RELATIONSHIP DIAGRAM	19
8	DATA FLOW DIAGRAM	20
9	SEQUENCE & COLLABORATION DIAGRAM	22
10	DEVELOPMENT OF TESTING FRAMEWORK/USER INTERFACE	24
11	TEST CASES MANUAL	26
12	MANUAL TEST CASE REPORT	28
13	ARCHITECTURE/DESIGN/Framework/IMPLEMENTATION	30
	CONCLUSION	

## **LIST OF FIGURES**

<b>FIGURE NO</b>	<b>TITLE</b>	<b>PAGE NO</b>
<b>1</b>	<b>WORK BREAKDOWN STRUCTURE</b>	<b>10</b>
<b>2</b>	<b>GANTT CHART</b>	<b>12</b>
<b>3</b>	<b>RISK ANALYSIS -SWOT</b>	<b>13</b>
<b>4</b>	<b>SYSTEM ARCHITECTURE</b>	<b>16</b>
<b>5</b>	<b>CLASS DIAGRAM</b>	<b>17</b>
<b>6</b>	<b>USE CASE DIAGRAM</b>	<b>18</b>
<b>7</b>	<b>ENTITY RELATIONSHIP DIAGRAM</b>	<b>19</b>
<b>8</b>	<b>DATA FLOW DIAGRAM LEVEL 0</b>	<b>20</b>
<b>9</b>	<b>DATA FLOW DIAGRAM LEVEL 1</b>	<b>21</b>
<b>10</b>	<b>SEQUENCE DIAGRAM</b>	<b>22</b>
<b>11</b>	<b>COLLABORATION DIAGRAM</b>	<b>23</b>

## Problem Statement

### Title: SRM RESEARCH HUB

DATE	18/01/2024
SUBMITTED BY	Ponnuri Aniruddha
TITLE / ROLE	Developer



### PROJECT DESCRIPTION:

- Current academic research papers are scattered across various sources, making it difficult for researchers to discover and access relevant publications.
- SRM Institute of Science and Technology (SRMIST) research papers may not receive adequate exposure beyond traditional academic circles, hindering their impact and recognition.
- Existing methods for sharing and collaborating on research papers are often cumbersome and lack user-friendly features, leading to inefficiencies in knowledge dissemination and collaboration.
- Researchers require a secure and intuitive platform to access, contribute, and collaborate on research papers while ensuring data privacy and integrity.
- By providing a centralized repository for research papers, the project aims to facilitate collaboration among researchers, students, and academic professionals, fostering innovation and knowledge exchange within the academic community.

### THE HISTORY:

- Academic research papers from SRM Institute of Science and Technology (SRMIST) are scattered across various sources including academic journals, conference proceedings, and institutional repositories.
- Researchers, students, and academic professionals struggle to discover and access relevant SRMIST research papers due to the lack of a centralized platform.
- Existing methods for sharing and collaborating on research papers are often inefficient and fragmented, leading to challenges in knowledge dissemination and collaboration.
- SRMIST research papers may not receive adequate visibility and recognition beyond traditional academic circles, limiting their impact and contribution to the broader research community.

**LIMITATIONS:**

- Limited access to research papers: The project's success may be hindered if there are restrictions or limitations on accessing SRM Institute of Science and Technology (SRMIST) research papers from official sources.
- Technical challenges: Complexities in integrating with SRMIST's official website or obtaining data from various websites may pose technical hurdles to the project's development and functionality.

**APPROACH:**

- Research and Analysis:
  - Conduct research on existing academic repositories and collaboration platforms.
  - Analyze user requirements and preferences within the SRM Institute of Science and Technology (SRMIST) community.
  - Investigate technical feasibility and compatibility with SRMIST's official website and data sources.
- Development Resources:
  - Determine the technologies and frameworks needed for web application development.
  - Set up development environments and version control systems for collaborative coding.
- Data Integration:
  - Establish connections with SRMIST's official website or data sources to fetch research paper metadata.
  - Develop scripts to retrieve and parse data from external sources.
  - Implement a data cleansing process to ensure that the data conforms to the database structure.
- Database Management:
  - Design and implement a scalable database schema for storing user information, research papers, and metadata.
  - Optimize database queries and indexing for efficient data retrieval and storage.

**BENEFITS:**

- Centralized repository for research papers enhances accessibility and visibility of SRM Institute of Science and Technology (SRMIST) publications.
- Facilitates collaboration and knowledge sharing among researchers, students, and academic professionals within the institution.
- Streamlines the process of discovering, accessing, and citing relevant research papers, thereby fostering academic excellence and innovation.

- Enhances the reputation and recognition of SRMIST as a hub for cutting-edge research and scholarly contributions.
- Encourages interdisciplinary collaboration and cross-departmental engagement by providing a platform for interdisciplinary research exploration.
- Strengthens the institution's research ecosystem by promoting transparency, accountability, and integrity in scholarly communication.
- Empowers researchers and students to showcase their work and contributions to the broader academic community, leading to increased citations and impact.
- Establishes SRMIST as a leader in digital innovation and academic excellence by leveraging modern technologies and best practices in research dissemination and collaboration.

#### **Selection of Methodology (WaterFall Methodology):**

- **Well-defined Requirements:** If the requirements for the SRM Research Hub are clear from the outset, with little expectation of change, the Waterfall model allows for a systematic progression through the stages of development.
- **Sequential Phases:** The Waterfall model's sequential nature – with distinct phases for requirements gathering, design, implementation, verification, and maintenance – ensures that each phase can be completed and reviewed for quality before moving on to the next.
- **Documentation and Rigor:** This model emphasizes rigorous documentation and adherence to initial plans, which can be beneficial for aligning with institutional policies and maintaining clarity of purpose throughout the project lifecycle.
- **Stakeholder Communications:** It can be easier to communicate progress to stakeholders with a Waterfall approach because the project is divided into distinct stages with concrete deliverables at the end of each phase.
- **Resource Planning:** The Waterfall model allows for more predictable resource allocation and scheduling since each phase is planned in detail at the start of the project.



**Incorporate information to below table regarding stakeholders of the project [Make use of below examples]**

Stakeholder Name	Activity/ Area /Phase	Interest	Influence	Priority (High/ Medium/ Low)
Researchers	Access to latest research papers	High	Low	Medium
Students	Access to educational resources	High	High	High
Faculty Members	Research dissemination and collaboration	High	High	High
External Researchers	Access to SRMIST research papers	High	Low	Low

### **System Requirements:**

#### Hardware

- Server infrastructure capable of hosting web services, managing databases, and storing sensitive data.
- Scalable storage solutions can handle increasing volumes of research papers and user data.

#### Software:

- Web server software and a database management system (such as MySQL or PostgreSQL).
- Frameworks and libraries designed for web application development

#### Compatibility:

- Compatibility with a variety of research paper file formats (for example, PDF and DOCX).

## Functional Requirements

### User Authentication and Authorization:

- Users should be able to register for accounts with a valid email address and password.
- Registered users should be able to log in to the website securely.
- The system should enforce password strength requirements and provide options for password recovery and account management.
- Different user roles (e.g., regular user, administrator) should have different levels of access permissions.

### Research Paper Discovery and Search:

- Users should be able to browse and search for research papers by title, authors, publication date, keywords, and categories.
- The search functionality should support advanced filtering options to refine search results.
- Users should be able to view detailed information about each research paper, including abstracts, authors, publication details, and related metadata.

### Research Paper Upload and Submission:

- Registered users should have the ability to upload research papers to the platform.
- The upload process should include fields for entering metadata such as title, authors, abstract, keywords, and publication date.
- The system should validate uploaded papers to ensure they meet specified file format and size requirements.
- Uploaded papers should undergo review by administrators before being made publicly accessible.

### User Interaction and Collaboration:

- Users should be able to bookmark or save research papers for later reference.
- Users should have the ability to create personal collections or folders to organize and manage saved papers.
- The system should support user comments, annotations, and discussions on individual research papers.
- Collaboration features such as sharing papers with colleagues or inviting collaborators should be available.

#### Administrative Tools:

- Administrators should have access to tools for managing user accounts, reviewing uploaded papers, and moderating user interactions.
- The system should provide administrative dashboards and reporting tools for monitoring platform usage and performance.

### **Non-Functional Requirements**

#### Usability:

- The website should have a clean and intuitive user interface, with consistent navigation and layout across pages.
- Content should be organized logically, with clear labels and descriptive headings to aid user comprehension.
- The website should be responsive and mobile-friendly, adapting to different screen sizes and devices

#### Reliability:

- The system should be reliable and resilient, with mechanisms in place to detect and recover from failures automatically.
- Error handling and logging should be implemented to track system errors and provide useful feedback to users and administrators.
- Backup and recovery procedures should be documented and tested regularly to ensure data integrity and availability.

## 1. Project Management Plan

Describe the key issues driving the project. **[Min 3 Focus Areas]**

Focus Area	Details
Integration Management	Governance Framework Project Team Structure Roles & Responsibilities of Team Change Management (Change Control, Issue Management) Project Closure
Scope Management	Scope Statement Requirement Management (Gathering, Control, Assumption, Constraint Stakeholder) Define Deliverable Requirement Change Control Activities and Sub-Tasks
Schedule Management	Define Milestones Schedule Control
Cost Management	Estimate Effort Assign Team Budget Control
Quality Management	Quality Assurance: Quality assurance will be managed including governance, roles and responsibilities, tools and techniques and reporting Quality Control: Specify the mechanisms to be used to measure and control the quality of the work products
Resource Management	Estimate and Manage the need People: People & Skills Required Finance: Budget Required Physical: Facilities, IT Infrastructure
Stakeholder	Identifying, Analyzing, Engaging Stakeholders
Communication Management	Determine communication requirements, roles and responsibilities, tools and techniques. [Type of Communication, Schedule, Mechanism Recipient]
Risk Management	Identifying, analysing, and prioritizing project risks
Procurement Management	Adhering to organization procurement process

## 2. Estimation

### 2.1. Effort and Cost Estimation

Activity Description	Sub-Task	Sub-Task Description	Effort (in hours)	Cost in INR
Design the user screen	E1R1A1T1 (Effort-Requirement-Activity-Task)	Designing the User Login page	5	10,000
	E1R1A1T2	Designing the Home Page	7	14,000
	E1R1A1T4	Designing the student dashboard	8	10,000
	E1R1A1T4	Designing the teacher dashboard	8	10,000
	E1R1A1T3	Designing the Back End	10	24,000
Identify Data Source for displaying units of Energy Consumption		Go through Interface contract (Application Data Exchange) documents	5	10,000
		Document	3	6,0000
		Software testing	8	16,000

Effort (hr)	Cost (INR)
1	5,000

### 2.2. Infrastructure/Resource Cost [CapEx]

Infrastructure Requirement	Qty	Cost per qty	Cost per item
IR1	PC's	4	70,000
IR2	Wi-Fi	1	4,000
	Hosting Server	1	6,000
	Backup Server	1	6,000
	Tie-Up with all Department	NA	NA

## 2.3 Maintenance and Support Cost [OpEx]

Category	Details	Qty	Cost per qty per annum	Cost per item
People	Network, System, Middleware and DB admin	3	2,000,000	60,00,000
	Developer , Support Consultant			
License	Operating System	10	10000	1,00,000
	Database Middleware IDE			
Infrastructures	Server, Storage and Network	20	20000	4,00,000

## 3. Project Team Formation

### 3.1. Identification Team members

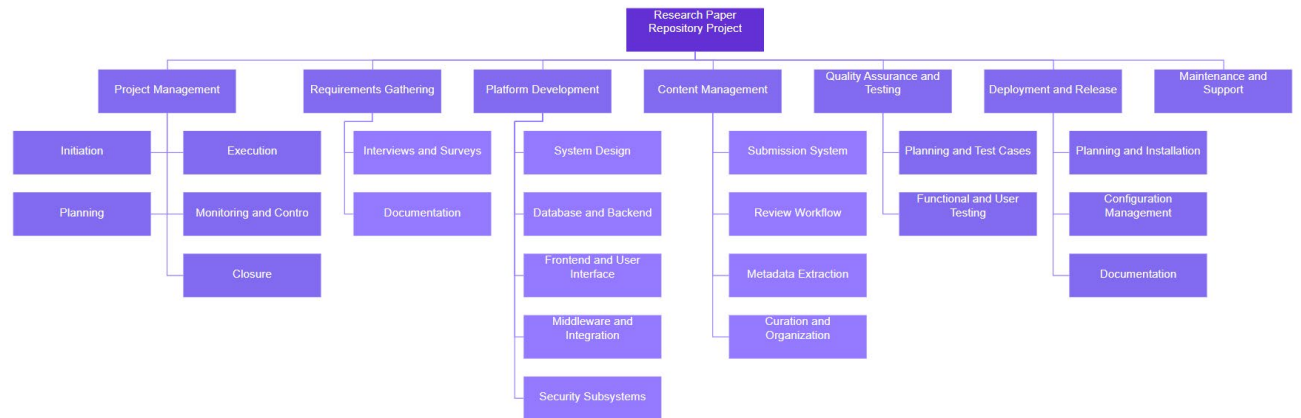
Name	Role	Responsibilities
vamshi	Key Business User (Product Owner)	Provide clear business and user requirements
vamshi,aniruddha	Project Manager	Manage the project
shabanya	Business Analyst	Discuss and Document Requirements
aniruddha	Technical Lead	Design the end-to-end architecture
vamshi	UX Designer	Design the user experience
vamshi	Frontend Developer	Develop user interface
aniruddha	Backend Developer	Design, Develop and Unit Test Services/API/DB
shabanya	Cloud Architect	Design the cost effective, highly available and scalable architecture
shabanya	Cloud Operations	Provision required Services
vamshi	Tester	Define Test Cases and Perform Testing

### 3.2. Responsibility Assignment Matrix

RACI Matrix	Team Members			
	Activity Tharun (Designer)	Praveen (Developer)	Chidvilas (Project Manager & BA)	Chidvilas - Key Business User
User Requirement Documentation	-	C/I	A	R
Advertisement	C	C	R	
Development	C	R	I	
Website Design	R	A	I	
Testing / Deployment	-	A	C	
Bug Fixes	A	R	I	
Update & Upgrade	-	C	A	

A	Accountable
R	Responsible
C	Consult
I	Inform

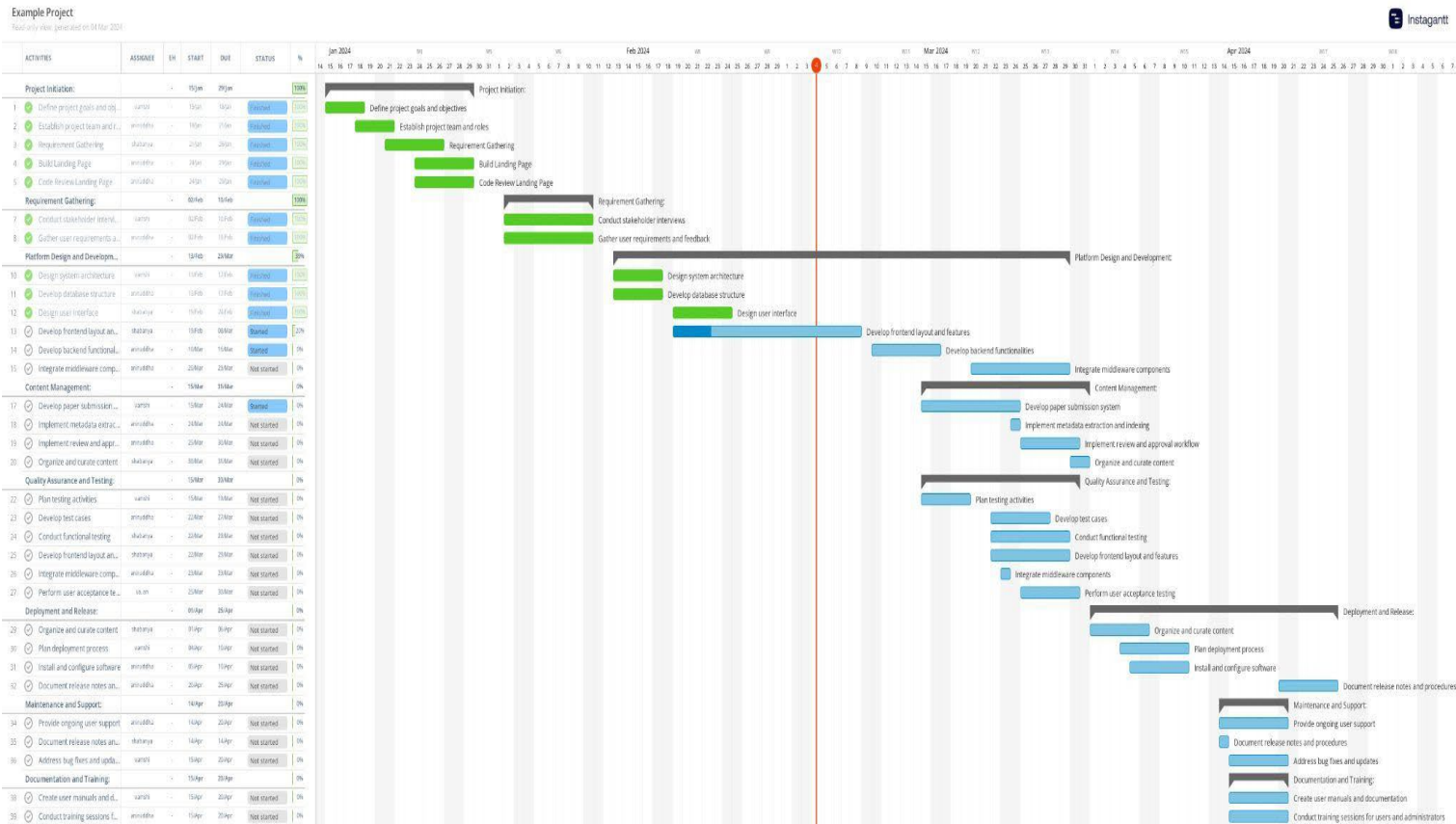
# WBS



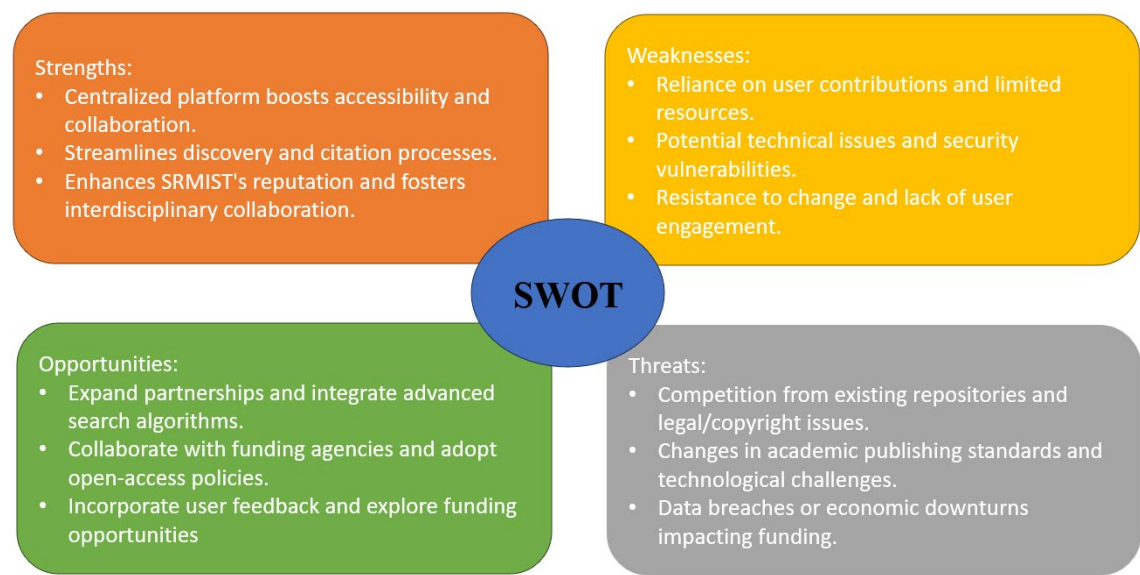


1. **Project Initiation**
  - Define project goals and objectives
  - Establish project team and roles
2. **Requirement Gathering**
  - Conduct stakeholder interviews
  - Gather user requirements and feedback
3. **Platform Design and Development**
  - Design system architecture
  - Develop database structure
  - Design user interface
  - Develop backend functionalities
  - Develop frontend layout and features
  - Integrate middleware components
4. **Content Management**
  - Develop paper submission system
  - Implement review and approval workflow
  - Implement metadata extraction and indexing
  - Organize and curate content
5. **Quality Assurance and Testing**
  - Plan testing activities
  - Develop test cases
  - Conduct functional testing
  - Perform user acceptance testing
6. **Deployment and Release**
  - Plan deployment process
  - Install and configure software
  - Document release notes and procedures
7. **Maintenance and Support**
  - Address bug fixes and updates
  - Provide ongoing user support
8. **Documentation and Training**
  - Create user manuals and documentation
  - Conduct training sessions for users and administrators

TIMELINE – GANTT CHART

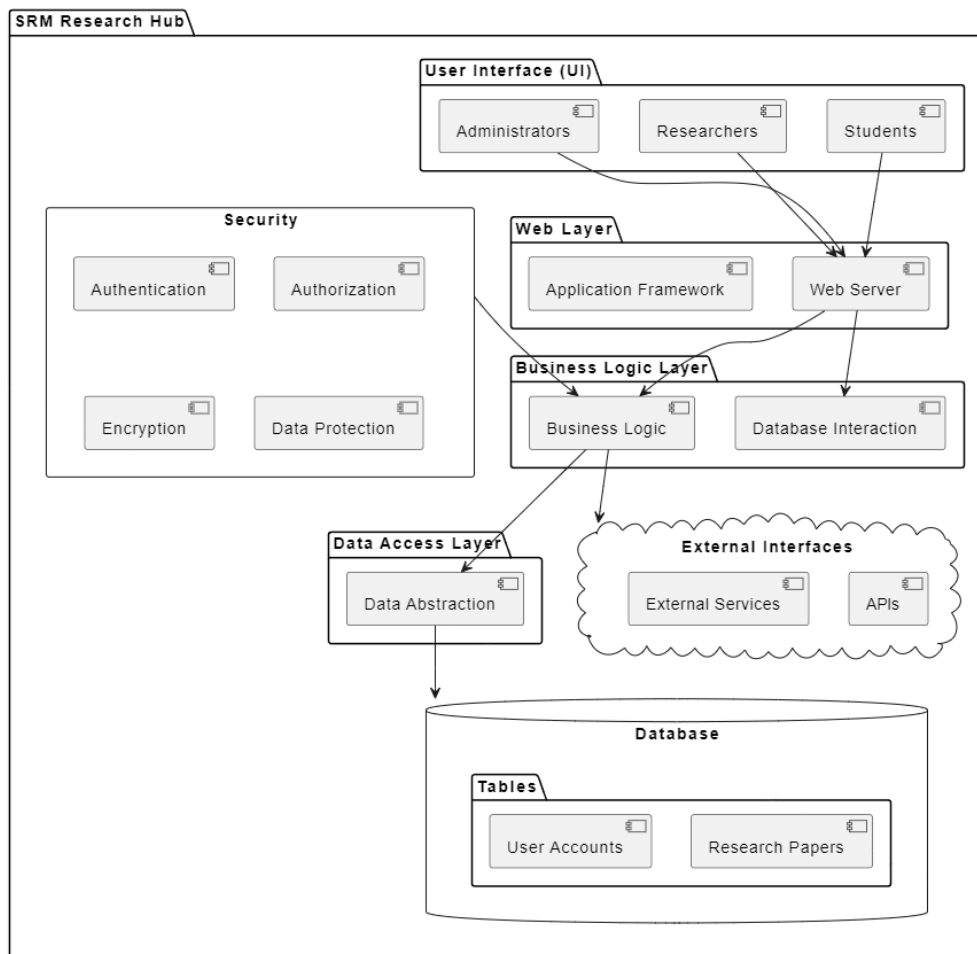


RISK ANALYSIS – SWOT & RMMM

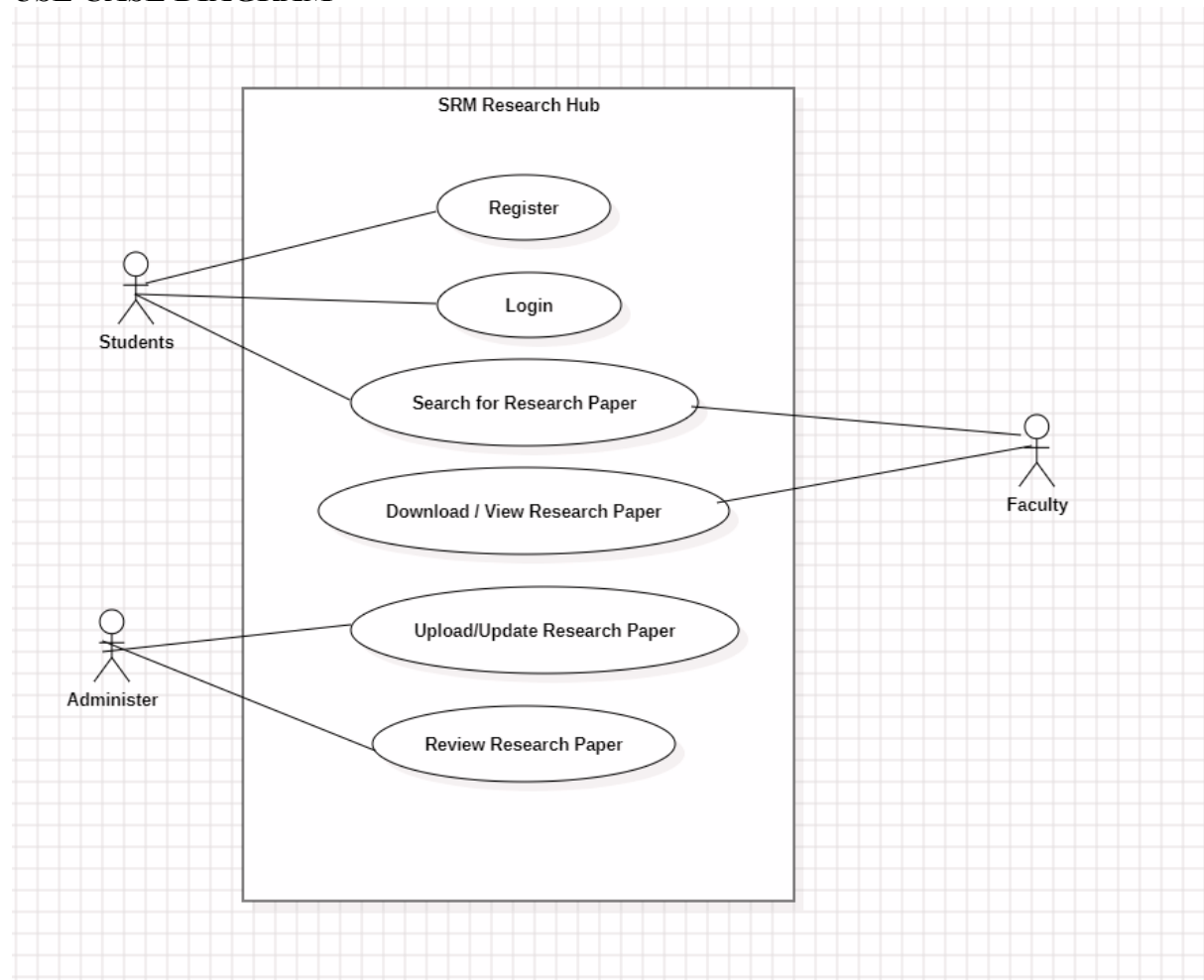


Risk	Response Strategy	Examples
Technical Issues	Regular testing and quality assurance	Conducting beta testing before full deployment
	Immediate troubleshooting and bug fixes	Assigning dedicated technical support team
Data Breaches	Enhanced security measures and encryption	Implementing two-factor authentication
	Regular security audits and monitoring	Conducting penetration testing to identify flaws
Content Accuracy	Content moderation and user reporting mechanisms	Establishing review processes for user submissions
	Educating users on proper citation and referencing	Providing guidelines and tutorials on platform
Funding Constraints	Diversifying funding sources and seeking grants	Applying for research grants and sponsorships
	Cost-cutting measures and resource optimization	Streamlining platform development and maintenance

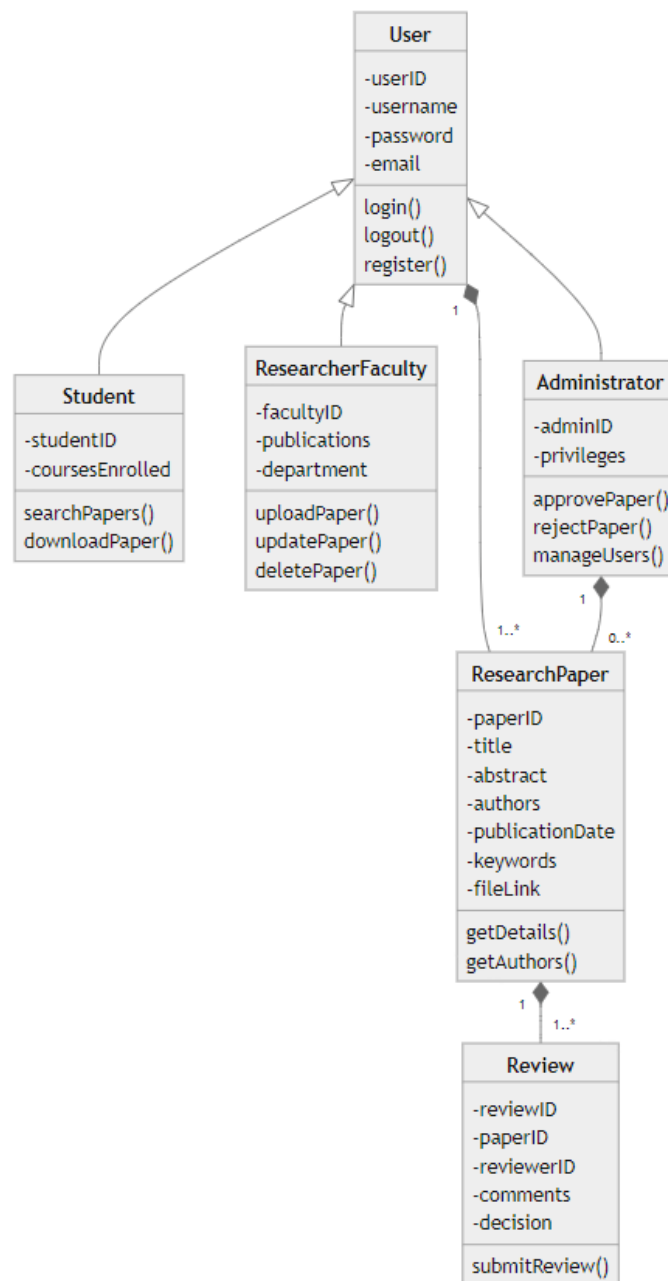
## SYSTEM ARCHITECTURE –



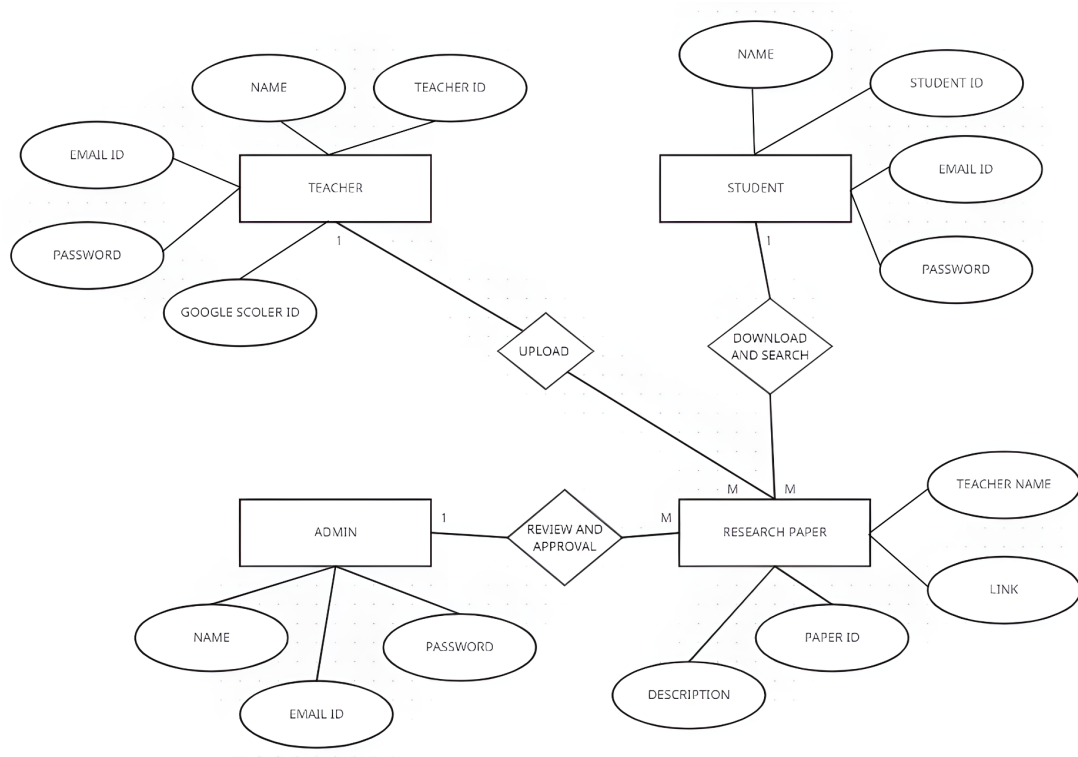
## USE CASE DIAGRAM



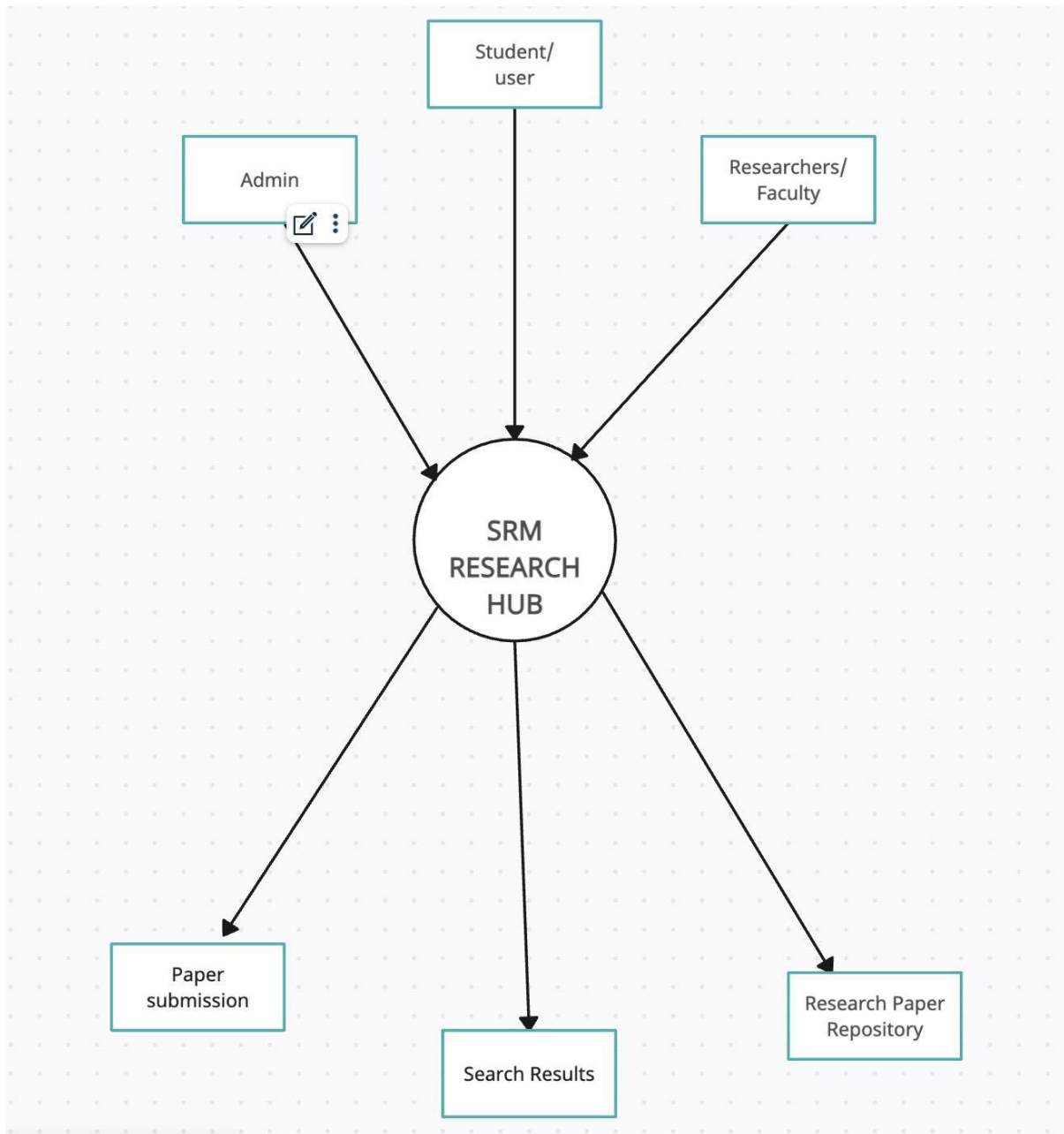
## CLASS DIAGRAM



# ER Diagram

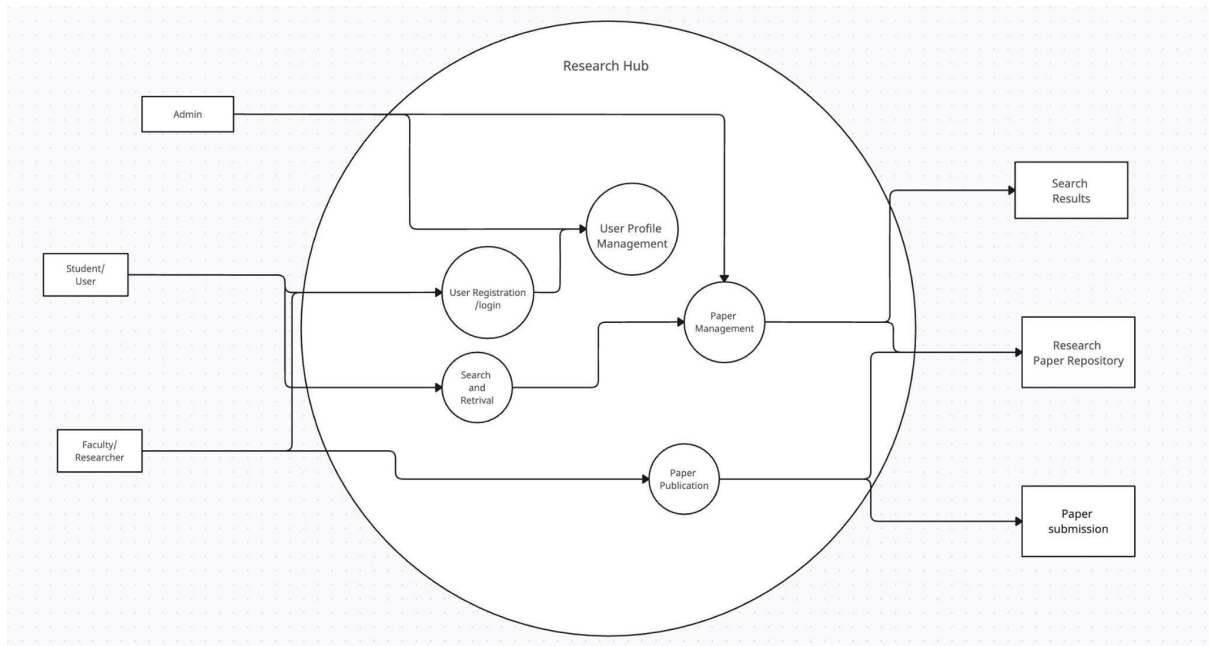


## DFD Level - 0

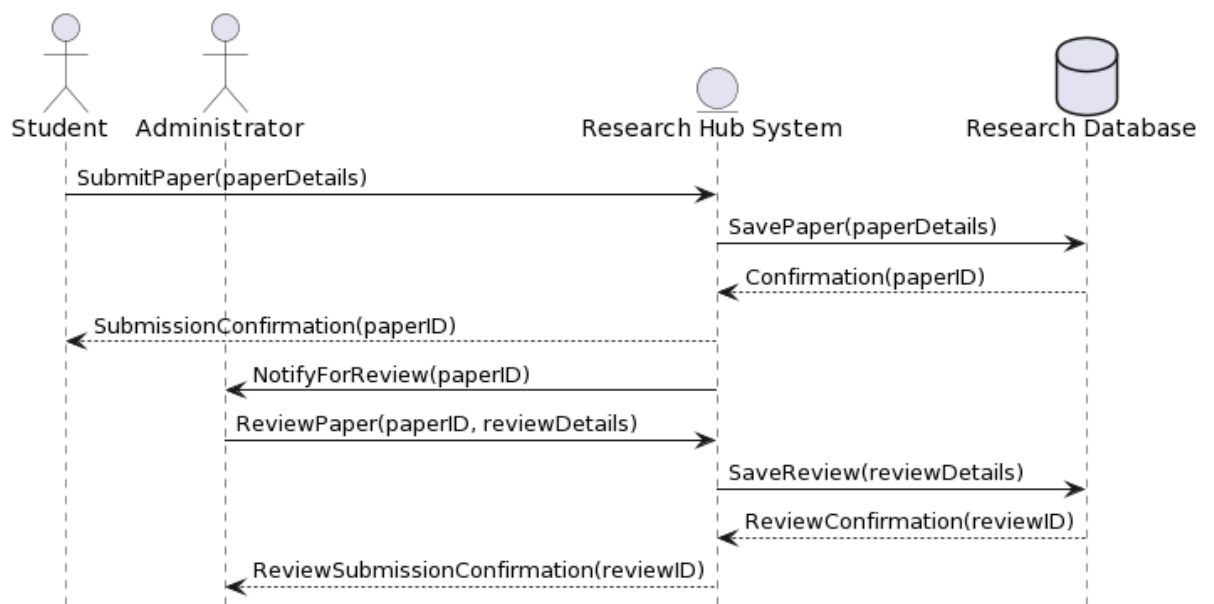




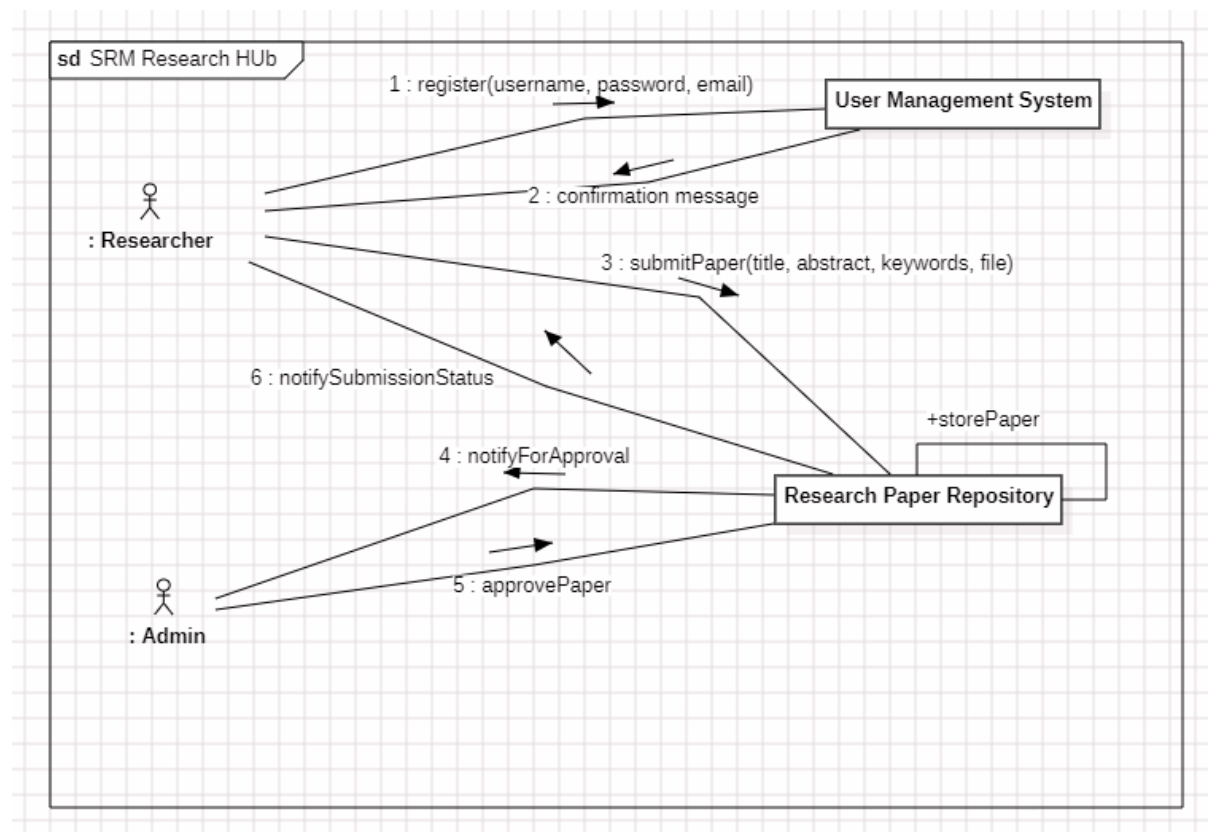
## DFD Level - 1



## Sequence Diagram



## Collaboration Diagram



# Executive Summary

The primary objective is to ensure the application functions as intended, meets user requirements, and delivers a high-quality user experience. We will employ a combination of manual and automated testing methodologies to achieve comprehensive coverage of both functional and non-functional aspects.

## Test Plan

### Scope of Testing:

- Functional Testing:
  - All application modules will be covered, with a focus on core functionalities like user registration, login, research paper management (upload, download, edit, delete), search and filtering, user profiles, and integration with SRMIST sources.
  - Automation will cover critical functionalities and regression testing of core user flows.
  - Manual testing will address specific test cases and edge cases that are less suitable for automation.
- Non-Functional Testing:
  - Performance testing will assess response times, load capacity, and scalability under varying user loads.
  - Usability testing will involve user feedback to evaluate the interface's intuitiveness, learnability, and overall user experience.
  - Security testing will ensure data security, role-based access control, and protection against vulnerabilities.

Category	Methodology	Tools Required	Description
Functional Requirements	Manual & Automated	Test case management tool Automation framework	Manual testing will cover user interaction, user interface elements, and various test cases. Automated testing will focus on critical functionalities, regression testing, and API interactions.
Non-Functional Requirements	Manual & Automated	Performance testing tool (e.g., JMeter, LoadRunner) Usability testing platform (e.g., UserTesting, Lookback) Security scanning tools (e.g., OWASP ZAP, Nessus)	Performance testing will assess response times, load capacity, and scalability using automated tools. Usability testing will involve user recruitment and observation through online platforms. Security testing will employ automated tools for vulnerability scanning and manual penetration testing (if resources permit).

## Functional Test Case

Test ID	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
1	Verify User Registration	Accept Valid SRM Registration Number	1. User clicks on "User Registration" link. 2. Enter a valid SRM registration number (e.g., RA2112704010015). 3. Click "Register" button.	User should be directed to the next page for entering more user details.			
2	Verify User Registration	Don't Accept Invalid Registration Number	1. User clicks on "User Registration" link. 2. Enter an invalid registration number (e.g., incorrect format or non-existent number). 3. Click "Register" button.	An error message should be displayed indicating invalid registration number format.			
3	Verify User Login	Successful Login with Valid Credentials	1. Enter a registered email address in the username field. 2. Enter the corresponding password in the password field. 3. Click "Login" button.	User should be successfully logged in and directed to the user dashboard.			
4	Verify User Login	Failed Login with Invalid Username	1. Enter an unregistered email address in the username field. 2. Enter a valid password. 3. Click "Login" button.	An error message should be displayed indicating invalid username or password.			
5	Verify User Login	Failed Login with Incorrect Password	1. Enter a registered email address in the username field.   2. Enter an incorrect password.   3. Click "Login" button.	An error message should be displayed indicating invalid username or password.			

	Verify Research 6 Paper Upload	Upload a Valid PDF Document	1. Navigate to the "Upload Research Paper" section. 2. Select a valid PDF document from the local storage. 3. Click "Upload" button.	The document should be uploaded successfully, and a confirmation message displayed.			
	Verify Research 7 Paper Upload	Upload an Unsupported File Format	1. Navigate to the "Upload Research Paper" section. 2. Select a file with an unsupported format (e.g., .docx). 3. Click "Upload" button.	An error message should be displayed indicating invalid file format.			

## Non-Functional Test Cases

Test ID	Test Scenario	Test Case	Execution Steps	Expected Outcome	Actual Outcome	Status	Remarks
1	Performance Testing	Load Capacity	1. Simulate concurrent user logins with increasing load.   2. Monitor response times and system performance.	The application should maintain acceptable response times and functionality under increasing load.			
2	Usability Testing	User Interface Clarity	1. Recruit users from the target audience (students, faculty, researchers).   2. Observe users interacting with the application and record their feedback.	Users should find the interface intuitive and easy to navigate.			
3	Security Testing	Data Encryption	1. Use security testing tools to scan for vulnerabilities related to data encryption.	All user data and research papers should be encrypted in transit and at rest.			

## **Current Status of the Testing:**

The testing of the SRM Research Hub application is currently progressing well, with an overall Amber status. We've made significant headway in functional testing of both User Management (Admin, Student, Teacher) and Paper Repository modules, achieving an overall coverage of 85%. User Login functionality is fully tested (100%), while other areas like User Registration, Profile Management, and Access Control are in progress (update percentages based on your project). Similarly, Paper Repository functionalities like Upload, Search & Filtering, and Download & Viewing are under development with progress (update percentages). Integration testing across modules is the next focus area to ensure a seamless user experience.

Non-functional testing has not yet begun. We'll prioritize completing remaining test cases for the core functionalities and establish a test data management plan to streamline the process. Open communication with stakeholders remains crucial to address any resource limitations and ensure alignment with project goals. By taking these steps, we can achieve a successful testing phase and launch a high-quality SRM Research Hub application.

## **Present obstacles to proceed further**

While the SRM Research Hub testing is progressing well, there are a few obstacles to achieving a complete green status. Integration testing, which ensures smooth interaction between User Management and Paper Repository modules, requires additional focus. Furthermore, a comprehensive test data management plan needs to be established to streamline testing across various user roles and functionalities. Resource limitations might also come into play, potentially slowing down progress.

To address these challenges, scheduling collaborative sessions between functional and development teams will be crucial for seamless integration. Additionally, developing a test data plan outlining data requirements for different test scenarios will optimize testing efficiency. If resource limitations become a bottleneck, expanding the testing team or prioritizing critical functionalities might be necessary. By proactively addressing these obstacles, we can ensure a successful testing phase and deliver a high-quality application.

## **Engaging Stakeholders:**

- Continuous communication with stakeholders will be crucial for addressing these challenges and ensuring project success.
- Stakeholders can provide valuable feedback on user needs and system priorities.
- Collaboration will help maintain alignment with project goals and secure necessary resources for a smooth testing completion.

## **Next Steps:**

- Prioritize completion of remaining test cases for User Management and Paper Repository modules.
- Conduct comprehensive integration testing to ensure seamless user experience across modules.
- Finalize a test data management plan to streamline test execution.

- Address resource limitations through collaboration or prioritization strategies.
- Maintain open communication with stakeholders to ensure project goals are met.

Category	Progress Against Plan	Status
Functional Testing	Amber	Completed
Non-Functional Testing	Amber	In-Progress

Functional	Test Case Coverage (%)	Status
User Login	100	Completed
Student page	70	In-Progress
Teacher Page	65	In-Progress
Admin Page	70	In-Progress
Research Paper Database	75	In-Progress

## Framework and Implementation:

Frontend: React.js

Backend: Node.js with Express framework

Database: MongoDB

## Architectural Design:

Frontend: Built with React.js, this layer handles user interactions, rendering dynamic content, and presenting the user interface.

Backend: Implemented using Node.js with Express framework, this layer processes user requests, interacts with the database, and performs core application logic.

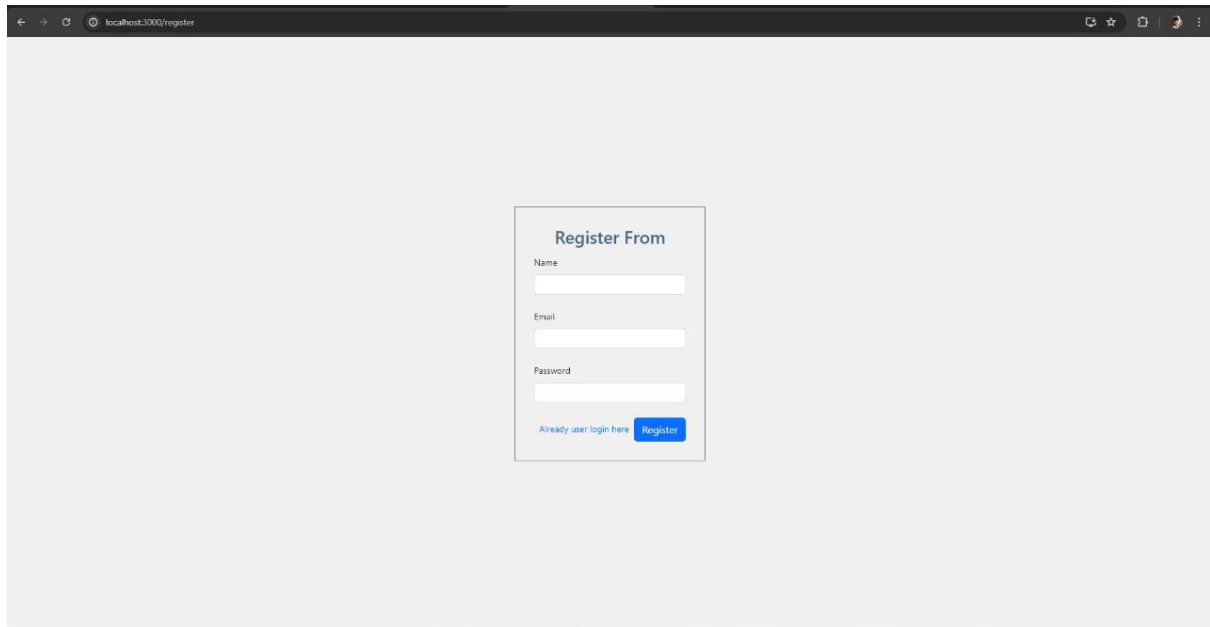
Database: MongoDB serves as the NoSQL database, storing research papers, user information, and other application data.



## User Management Module:

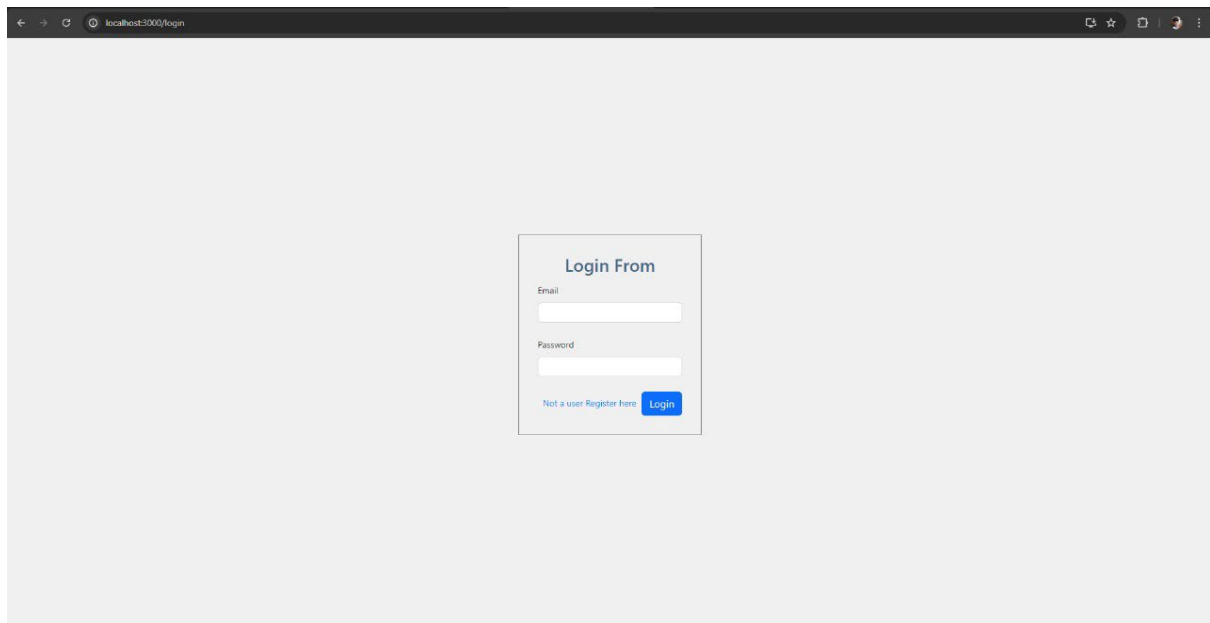
Description: This module handles user registration, login, profile management, and authorization.

### User Registration:



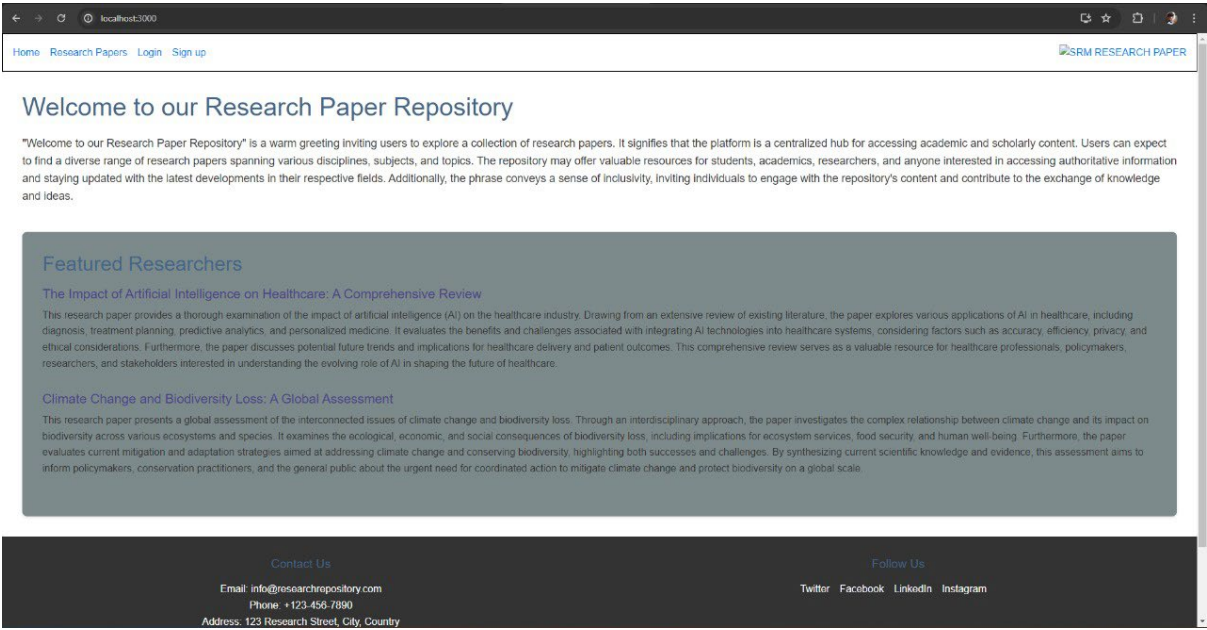
The screenshot shows a web browser window with the address bar displaying 'localhost:3000/register'. The main content area is a light gray background. In the center, there is a white rectangular form titled 'Register From'. The form contains three input fields: 'Name', 'Email', and 'Password'. Below the 'Password' field, there is a link that says 'Already user login here' and a blue button labeled 'Register'.

### User Login:

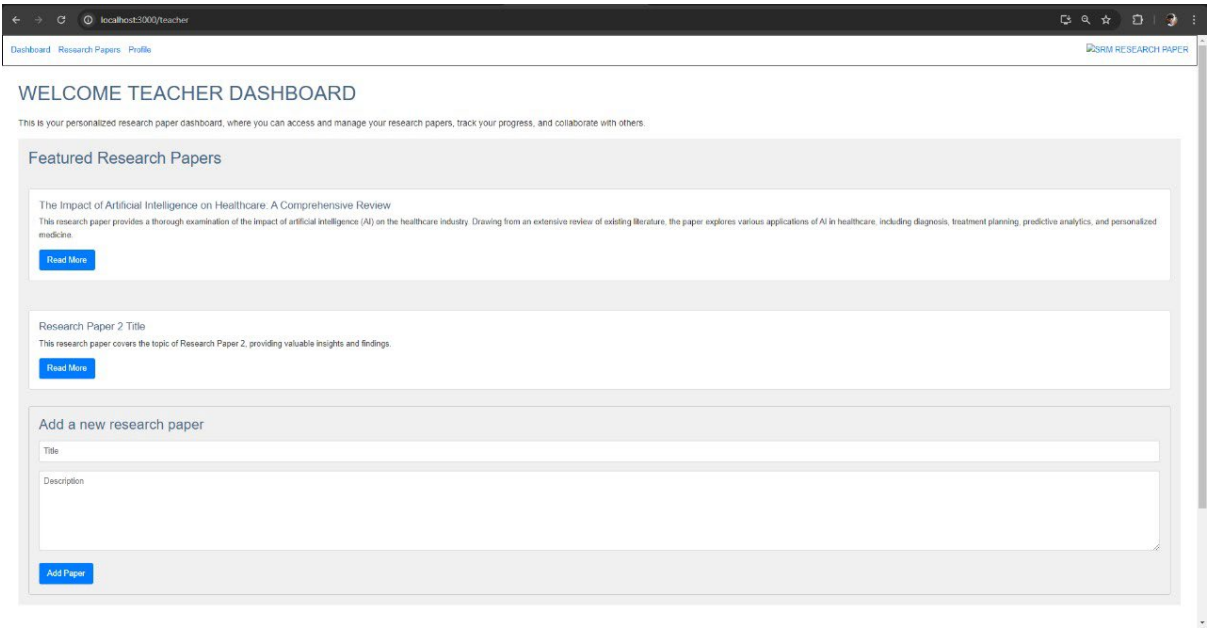


The screenshot shows a web browser window with the address bar displaying 'localhost:3000/login'. The main content area is a light gray background. In the center, there is a white rectangular form titled 'Login From'. The form contains two input fields: 'Email' and 'Password'. Below the 'Password' field, there is a link that says 'Not a user Register here' and a blue button labeled 'Login'.

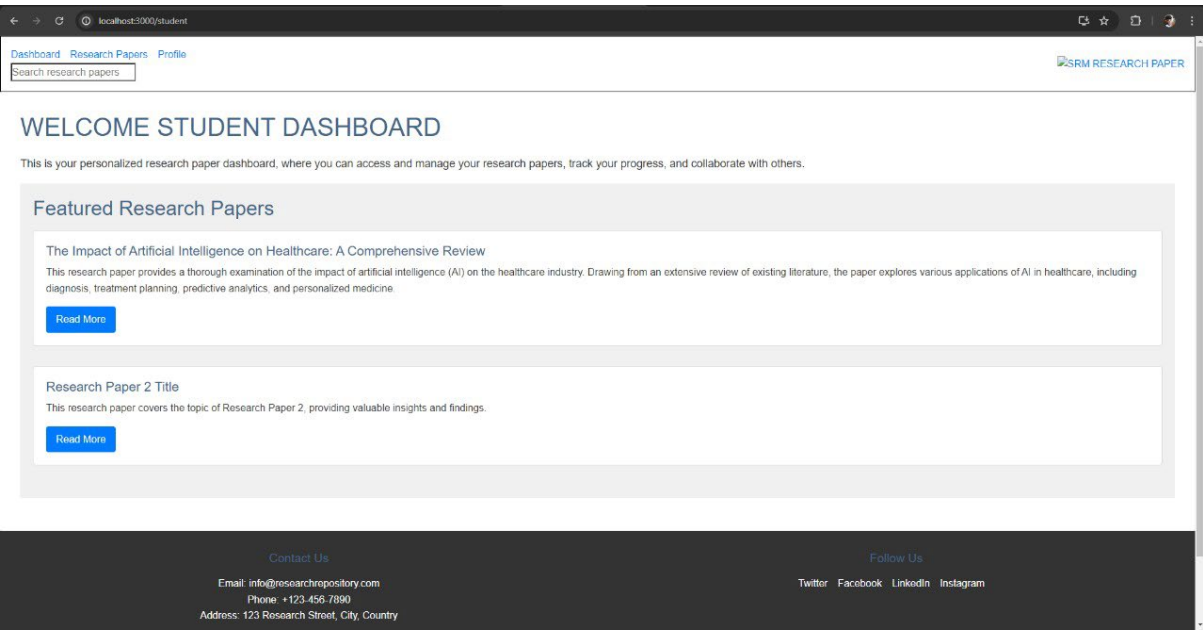
## Home Page (Landing Page):



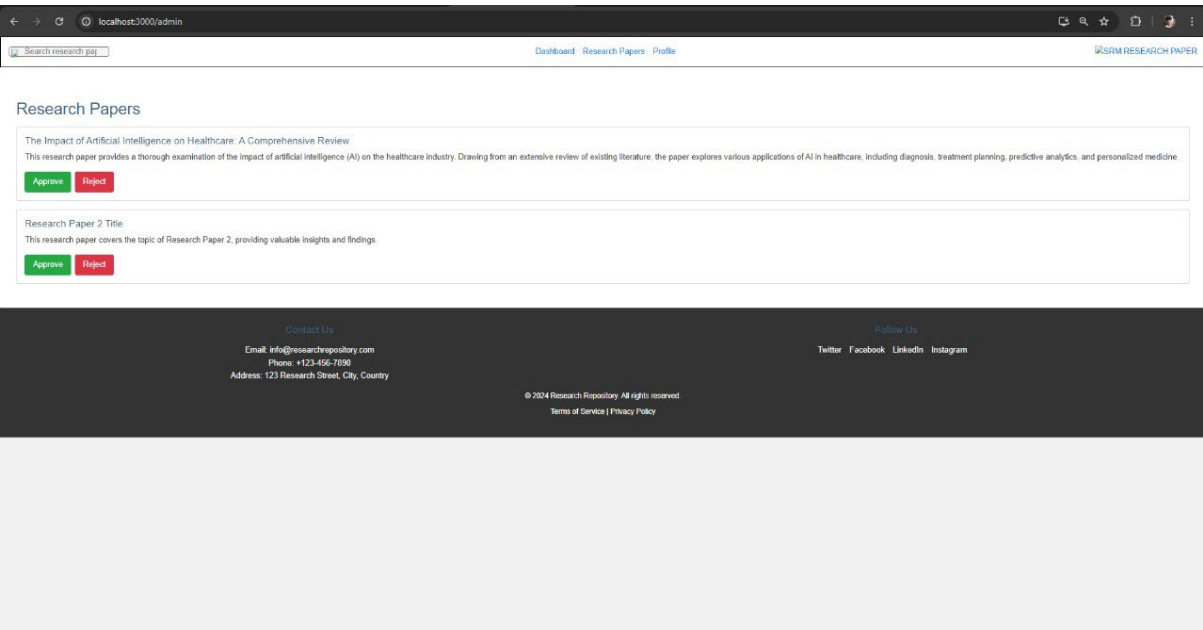
## Teacher Dashboard:



Student dashboard:



Admin Dashboard:



## Code:

```
client > src > pages > # Logins > # onfinishHandler
4 import { Link, useNavigate } from "react-router-dom";
5 import axios from "axios";
6
7 const Login = () => {
8   const navigate = useNavigate();
9
10  //form handler
11  const onfinishHandler = async (values) => {
12    try {
13      const res = await axios.post("/api/v1/user/login", values);
14      if (res.data.success) {
15        localStorage.setItem("token", res.data.token);
16        message.success("Login Successfully");
17
18        // Check the user role and navigate accordingly
19        if (res.data.user.role === "student") {
20          navigate("/student"); // Replace with the actual path if different
21        } else if (res.data.user.role === "teacher") {
22          navigate("/teacher"); // Replace with the actual path if different
23        } else {
24          message.error("User role not found");
25          navigate("/");
26        }
27      } else {
28        message.error(res.data.message);
29      }
30    } catch (error) {
31      console.log(error);
32      message.error("something went wrong");
33    }
34  };
35
36  return (
37    <div className="form-container">
38      <form
39        layout="vertical"
40        onFinish={values => onfinishHandler(values)}
41        className="register-form">
42        <div>
43          <h3 className="text-center">Login From /h3>
44
45          <Form.Item label="Email" name="email">
46            <input type="email" required />
47          </Form.Item>
48          <Form.Item label="Password" name="password">
49            <input type="password" required />
50          </Form.Item>
51          <Link to="/register" className="a-2">
```

```
1 import React from "react";
2 import "../styles/HomePage.css";
3 import "../pages/Login.js";
4 import { Link } from "react-router-dom";
5
6 const HomePage = () => {
7   return (
8     <div>
9       <header>
10        <nav>
11          <div>
12            <a href="/home" class="home-btn">
13              Home
14            </a>
15          </div>
16          <div>
17            <a href="/research" class="research-btn">
18              Research Papers
19            </a>
20          </div>
21          <div>
22            <a href="/login" class="login-btn">
23              Login
24            </a>
25          </div>
26          <div>
27            <a href="/register" class="signup-btn">
28              Sign up
29            </a>
30          </div>
31        </nav>
32
33        <div class="logo">
34          <a href="/link-to-your-homepage">
35            
36          </a>
37        </div>
38      </header>
39
40      <main>
41        <section className="intro">
42          <h1>Welcome to SRM RESEARCH HUB </h1>
43
44          <p>
45            "Welcome to our Research Paper Repository" is a warm greeting
46            inviting users to explore a collection of research papers. It
47            signifies that the platform is a centralized hub for accessing
48            academic and scholarly content. Users can expect to find a diverse
49            range of research papers spanning various disciplines, subjects, and
```

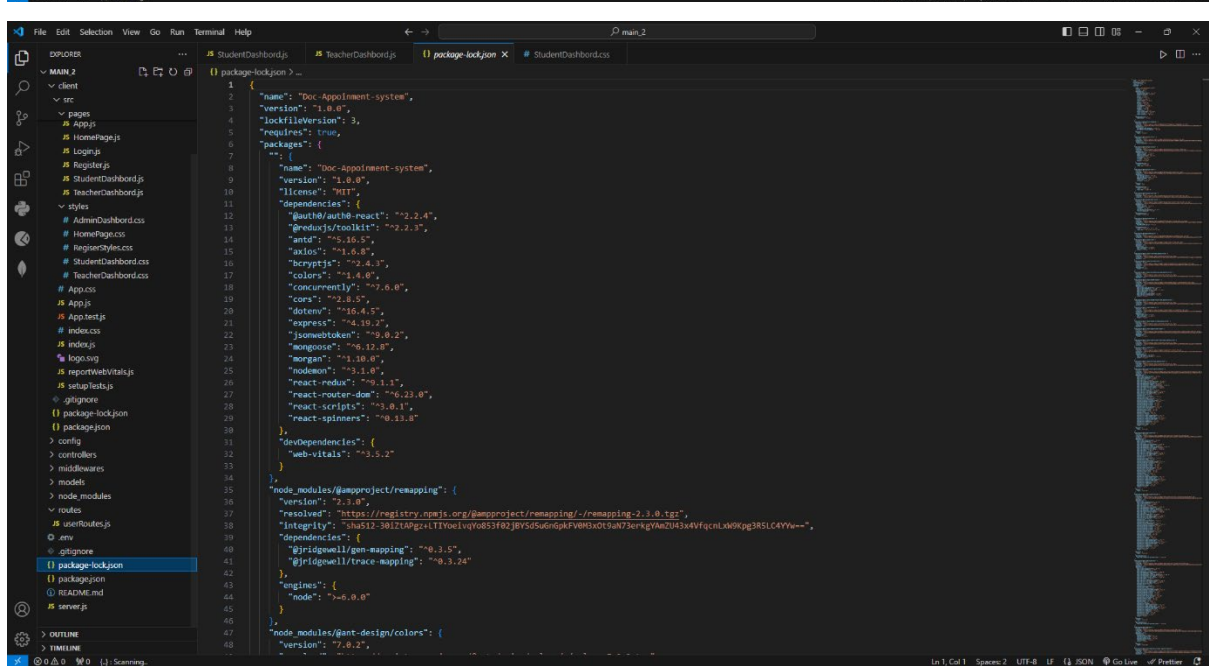
The screenshot shows the VS Code editor with the `main_2` project open. The file explorer on the left shows the project structure, including `server.js`. The main editor displays the content of `server.js`, which is a Node.js server setup using Express, Morgan, Dotenv, and MongoDB. The code includes imports for these modules, configuration for environment variables and database connection, and a REST API endpoint for user authentication.

```
1 const express = require("express");
2 const colors = require("colors");
3 const morgan = require("morgan");
4 const dotenv = require("dotenv");
5 const connectDB = require("./config/db");
6 const cors = require("cors");
7
8 // Dotenv config
9 dotenv.config();
10
11 // MongoDB connection
12 connectDB();
13
14 // REST object
15 const app = express();
16
17 // Middlewares
18 app.use(express.json());
19 app.use(morgan("dev"));
20 app.use(cors());
21
22 // routes
23 app.use("/api/v1/user", require("../routes/userRoutes"));
24
25 // port
26 const port = process.env.PORT || 8080;
27
28 // listen port
29 app.listen(port, () => {
30   console.log(
31     `Server Running in ${process.env.NODE_ENV} mode on port ${port}`.cyan
32   );
33 });
34
```

The screenshot shows the VS Code editor with the `main_2` project open. The file explorer on the left shows the project structure, including `TeacherDashboard.js`. The main editor displays the content of `TeacherDashboard.js`, which is a React component for a teacher dashboard. The component uses `useState` to manage state for the current view, papers, and user profile. It includes a `handleAddNewPaper` function to handle adding new papers and a `render` method to return the JSX for the dashboard.

```
1 import React, { useState } from "react";
2 import "../styles/TeacherDashboard.css";
3
4 const TeacherDashboard = () => {
5   const [currentView, setCurrentView] = useState("dashboard");
6   const [papers, setPapers] = useState([]);
7   const [newPaperTitle, setNewPaperTitle] = useState("");
8   const [newPaperDescription, setNewPaperDescription] = useState("");
9   const [teacherProfile, setTeacherProfile] = useState({
10     name: "John Doe",
11     email: "johndoe@example.com",
12     bio: "I am a teacher at SRM University",
13   });
14
15   const handleAddPaper = () => {
16     // Add your logic here to handle adding a new paper
17     console.log("Adding new paper...");
18   };
19
20   const handleAddNewPaper = () => {
21     const newPaper = {
22       id: Date.now(),
23       title: newPaperTitle,
24       description: newPaperDescription,
25     };
26     setPapers([...papers, newPaper]);
27     setCurrentView("dashboard");
28     setNewPaperTitle("");
29     setNewPaperDescription("");
30   };
31
32   return (
33     <div>
34       <div>
35         <nav>
36           <ul>
37             <li>
38               <a
39                 href="#dashboard"
40                 className="dashboard-btn"
41                 onClick={() => setCurrentView("dashboard")}
42               >
43                 Dashboard
44             </a>
45           </li>
46         </ul>
47       </div>
48     </div>
49   );
50 }

```



## CONCLUSION

In conclusion, this project presents a comprehensive approach to address the challenges associated with research paper collaboration through the development of SRM RESEARCH HUB. By leveraging modern technologies such as the MERN stack and cloud storage, coupled with automated tracking and real-time communication integration, the system aims to significantly enhance operational efficiency and decision-making capabilities for businesses and organizations. Despite the identified limitations, the project's focus on user-friendliness and engagement with stakeholders, including mental health experts, underscores its commitment to delivering a robust and accessible solution. Ultimately, the envisioned Inventory Control System promises to revolutionize inventory management processes, reduce costs, and pave the way for sustainable growth and success in today's competitive market landscape.

## REFERENCES

- **Boosting Java Application Development with Eclipse:** This book, available as of December 2020, offers a comprehensive guide on using Eclipse for Java application development, covering aspects like configuring servers, automatic code generation, and exception handling, which could be particularly useful for backend development aspects of your project.
- **Analysis and Classification of Requirements Specification for Web Application Development:** A study from July 2020 that provides a model for classifying stakeholders' requirements in web application development. This could be instrumental in ensuring that your project meets the needs of its intended users and stakeholders effectively.
- **Web Application Development Based on Gamification Technique:** This thesis from July 2020 explores using gamification techniques in web application development to enhance learning achievement and motivation. This approach could be useful if your project aims to engage users in an educational context or to increase user interaction and retention.
- **Leukemia Decision Support Web Application Development Process:** A poster from April 2020 detailing the development process of a web application aimed at aiding medical decision-making. This resource could provide insights into designing web applications for specialized fields and ensuring user needs are met effectively.
- **Research and Analysis of the Front-end Frameworks and Libraries in E-Business Development:** This paper discusses the design and development of web-based platforms for conducting online research, highlighting the importance of scalable architecture and the use of REST API for component communication, which could be applicable to your project's design.
- Each of these papers provides a unique perspective that could be applied to different aspects of your project, from technical implementation to user engagement strategies. Exploring these resources could offer valuable insights and methodologies to enhance the development and success of your web application.