Research Management System

Project Overview

The project aims to create a web application with role-based access control, including admin, researcher, projecter, and reviewer panels. Users, primarily teachers, can register and choose roles. Admins have full control over the system, researchers create and manage projects, reviewers assess projects, and projecters oversee the entire process. The system enhances communication with a notification system, introduces user profiles, and admin profiles.

- **Admin Panel**: Allows administrators to manage all aspects of the system, including users, projects, and reviewers/researchers.
- **Researcher Panel**: Provides researchers with tools to create and manage research projects.
- **Reviewer Panel**: Offers reviewers the ability to evaluate and review research projects.
- **User Registration**: Allows users, including teachers, to register and select roles as researchers, reviewers, or both.

Proposed Features

1. **User Registration and Authentication:** Users can register for the system and log in with secure authentication.
2. **Role-Based Access Control:** The application enforces role-based access control, allowing administrators, researchers, and reviewers to access specific features.
3. **Admin Panel:** Administrators have full control over the system, enabling them to manage users, projects, and review processes.
4. **Researcher Panel:** Researchers can create and oversee research projects through a dedicated panel.

5. **Reviewer Panel:** Reviewers can evaluate and provide feedback on research projects.
6. **Projecter Panel:** Introduce a new panel for projecters who oversee the entire project lifecycle.
7. **Notification System:** Implement a notification system to keep users informed about project status updates and reviewer comments.
8. **Improved User Profiles:** Enhance user profiles with additional information and customization options.
9. **User Dashboard:** Create a user dashboard for an overview of user activities and project involvement.
10. **User Messaging:** Enable in-system messaging for seamless communication between users, projecters, and administrators.
11. **Project Overview Dashboard:** A dedicated dashboard that provides a summarized view of the overall project progress and key metrics.
12. **Advanced Search and Filter:** Implement advanced search and filtering options to help users find specific projects or users more efficiently.
13. **Data Visualization:** Add data visualization tools to help researchers and projecters analyze project data more effectively.
14. **Review Assignment System:** Automatically assign projects to reviewers based on specific criteria and preferences.
15. **User Analytics:** Track and present user behavior data to improve the overall user experience and system performance.
16. **Integrated File Storage:** Include secure and integrated file storage for project-related documents and data.

Proposed Technologies

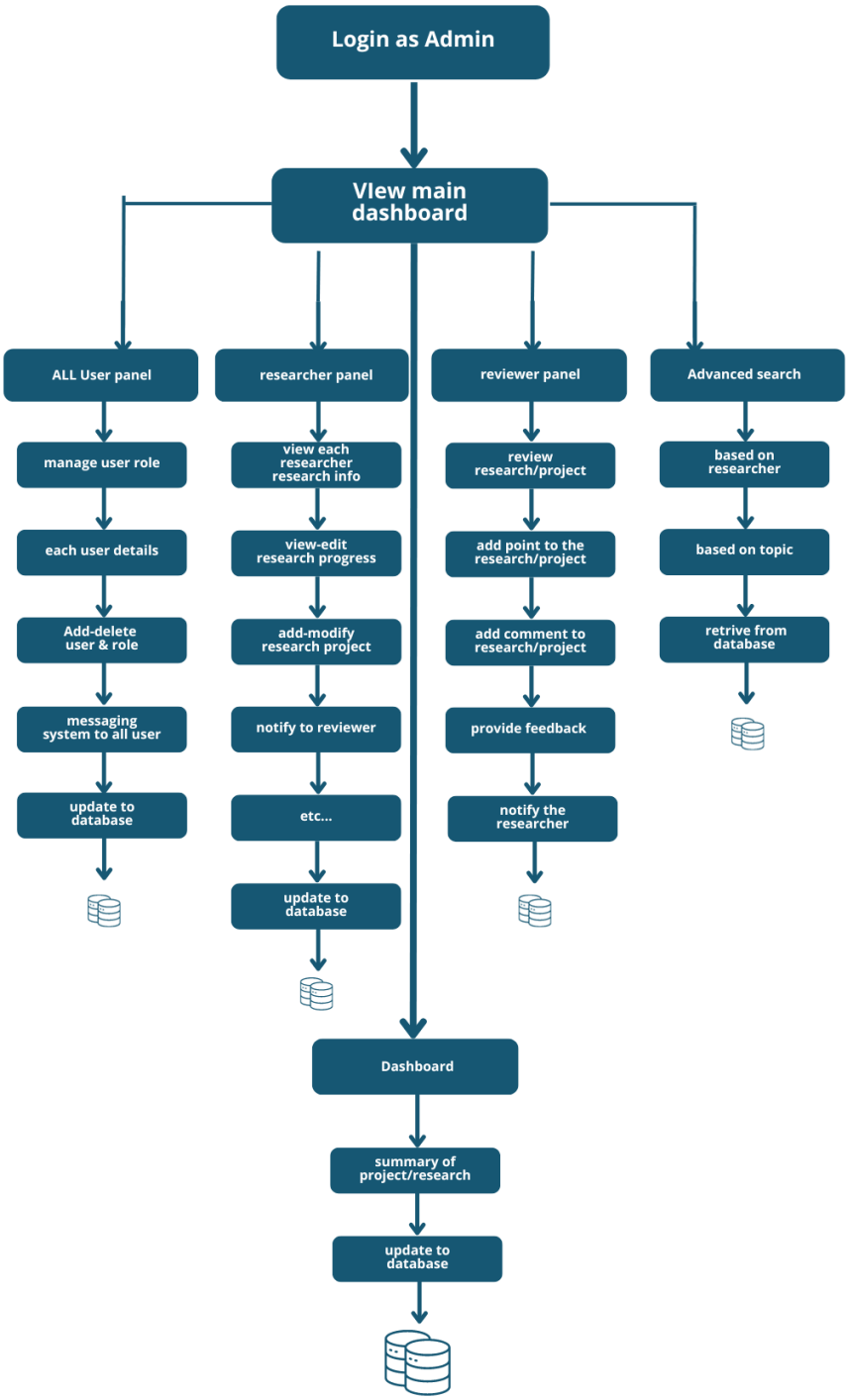
Front-end:

- **React**: A popular and powerful JavaScript library for building user interfaces.
- **Redux**: For state management and handling complex interactions.
- **React Router**: For routing within the application.
- **Material-UI**: A UI framework for consistent design.

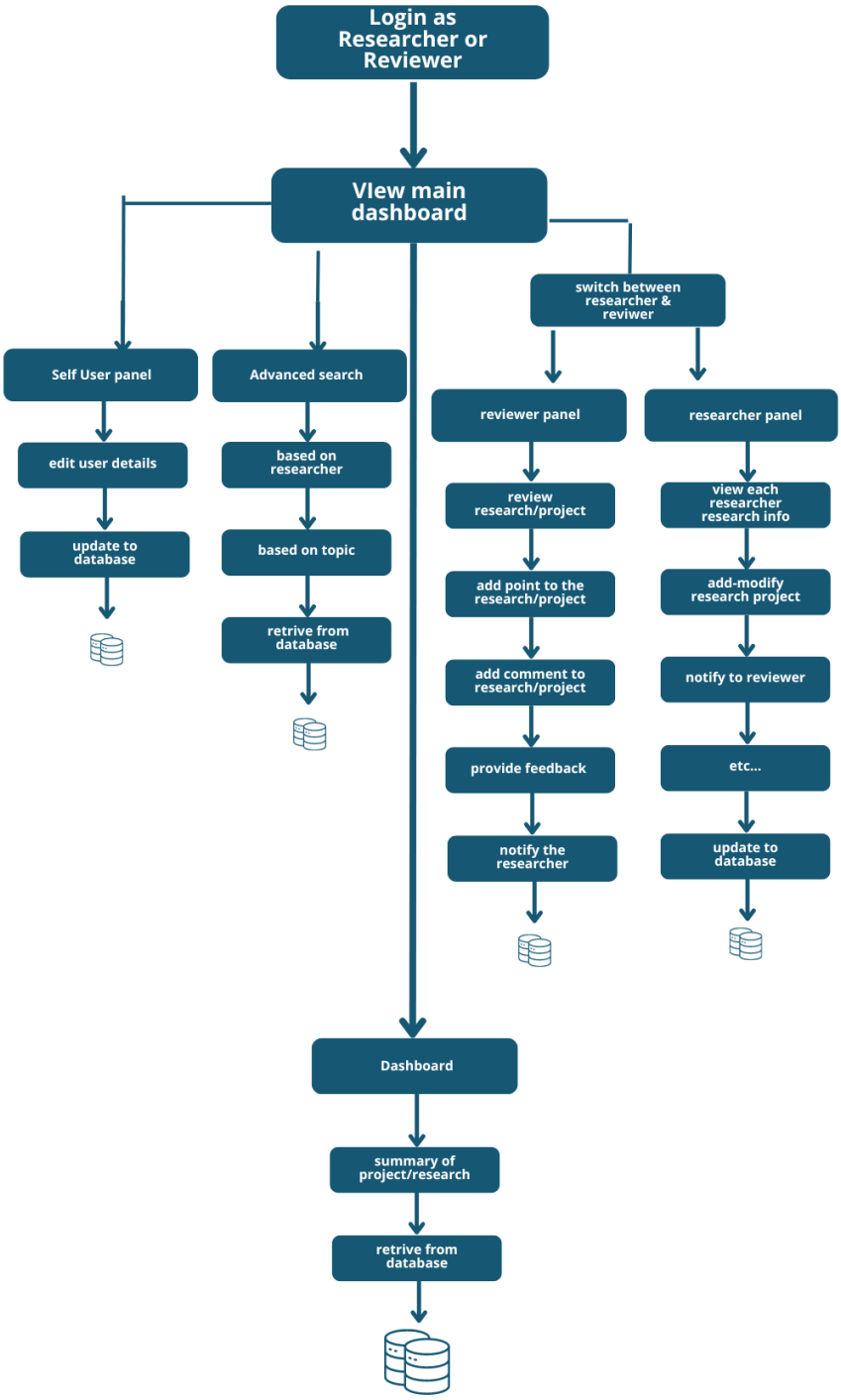
Back-end:

- **Node.js**: A runtime environment for server-side applications.
- **Express.js**: A web application framework for building APIs.
- **MongoDB**: A NoSQL database for flexible data storage.
- **Mongoose**: An Object-Data Modeling library for MongoDB.
- **Passport.js**: For user authentication and authorization.
- **JWT**: JSON Web Tokens for secure user authentication.

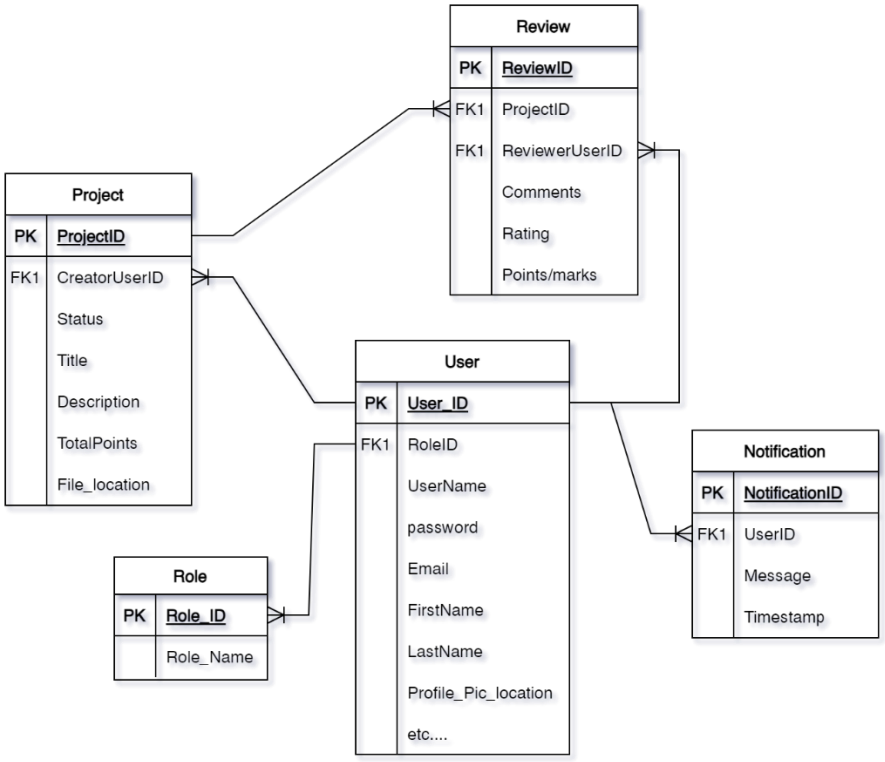
Proposed Data Flowchart (Admin)



Proposed Data Flowchart (Researcher/Reviewer)



Proposed ER Diagram



Proposed Timeline (Gantt Chart)

- **Phase 1: Planning and Design (2 weeks)**
    - Define project requirements and objectives.
    - Create system architecture and design wireframes.
- **Phase 2: Front-end Development (6 weeks)**
    - Develop the user interface components.
    - Implement user registration and authentication.
- **Phase 3: Back-end Development (8 weeks)**
    - Build API endpoints for user management.
    - Develop role-based access control.
- **Phase 4: Additional Features (4 weeks)**
    - Create the Projecter Panel.
    - Implement the notification system.
- **Phase 5: Testing and Quality Assurance (4 weeks)**
    - Conduct thorough testing of the entire system.
    - Address and fix any bugs or issues.
- **Phase 6: Documentation and Deployment (2 weeks)**
    - Create user and admin documentation.
    - Deploy the application on a hosting platform.
- **Phase 7: User Testing and Feedback (2 weeks)**
    - Invite users to test the system.
    - Gather feedback for further improvements.