Software Requirements Specification

for

ePathshala (Student ERP System)

Version 1.0

Prepared by Project Team 02

Name	PRN
Anushree Upadhye	250240320019
Gaurav Ahirrao	250240520020
Ravi Yadav	250240320092
Saket Kharche	250240520073
Satyajeet Khamkar	250240520076

Centre for Development of Advanced Computing

Raintree Marg, Near Bharati Vidyapeeth, Opp. Kharghar Railway Station, Sector 7,
CBD Belapur, Navi Mumbai - 400 614 - Maharashtra (India)
Phone: +91-22-27565303
Fax: +91-22-2756-0004

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed and structured description of the ePathshala (Student ERP System), a centralized web-based solution designed to streamline academic and administrative operations within an educational institution. This Software Requirements Specification (SRS) outlines the system's functional and non-functional requirements and describes the core features, including role-based dashboards, secure authentication, student information management, attendance tracking, grade viewing, assignment sharing, leave workflows, an Online Exam Module with Auto-Evaluation, support for Time-Bound Exams with Auto-Submission, an enhanced Assignment Workflow with File Upload capabilities, integrated Chat Forums for communication and collaboration, a Chatbot feature for user guidance, online meeting and lecture support for teachers and students, a comprehensive Academic Calendar, and a Notification feature with priority-based sequencing. The document aims to guide the development team, testers, stakeholders, and future maintainers in understanding the system scope, user roles, and performance expectations.

1.2 Intended Audience and Reading Suggestions

The document is intended for a broad audience, including developers, students, teachers, parents, and educational institution administrators.

For developers, this Software Requirements Specification (SRS) serves as a detailed guide for the design, implementation, and testing of the Student ERP System. It outlines all functional and non-functional requirements, system architecture, performance metrics, and security protocols. Developers should study the document thoroughly to ensure their code aligns with the system's objectives and stakeholder expectations, covering areas such as online examination handling with auto-evaluation and time-bound auto-submission, assignment workflows with secure file uploads, integrated chat forums, chatbot-assisted navigation, online meeting functionalities, academic calendar integration, and priority-based notification management.

For students, this document offers an overview of the ERP system's functionalities, including grade viewing, attendance tracking, assignment downloads and uploads, the leave application process, participation in chat forums, receiving priority-sequenced notifications, accessing academic calendar updates, attending online lectures, and attempting exams through the online examination module with both auto-evaluation and time-bound submission capabilities. It helps students understand how the platform works and how to navigate their role-based dashboard effectively, with additional guidance provided through the integrated chatbot.

For teachers and parents, this document describes how the system supports academic monitoring, communication, and collaboration. Teachers can manage attendance, grades, assignments with file submissions, conduct and evaluate online exams, schedule and host online lectures, interact through chat forums, and send or respond to priority notifications. Parents can track their child's performance, review academic schedules via the calendar, receive important updates promptly, and approve leaves through the platform.

For educational institutions, the SRS defines the project's vision, purpose, and capabilities. School administrators and decision-makers should review the document to assess how well the system fits their operational needs, from automated exam evaluation to communication tools, event scheduling, and notification management, ensuring it aligns with institutional goals.

1.3 Project Scope

The scope of the ePathshala (Student ERP System) defines the boundaries and primary objectives of the project, specifying the functionalities to be delivered and the roles supported. This system is designed to automate and simplify academic and administrative workflows within an educational institution through a centralized, secure, and role-based web platform.
The project includes the following key features and capabilities:

- Development of a web platform with separate dashboards for Admin, Student, Teacher, and Parent roles, each offering role-specific functionality.
- Implementation of secure user authentication and role-based authorization using JWT (JSON Web Tokens), ensuring controlled access and data protection across different user types.
- Admin functionalities including adding and managing Students, Teachers, and Parents, assigning teachers to classes and subjects, managing academic calendars, and overseeing system-wide data.
- Teacher functionalities for entering student grades, marking attendance, uploading and managing assignments with file submission support, approving student leave applications, hosting online lectures, creating and managing time-bound online exams with auto-evaluation, and participating in academic discussions via chat forums.
- Student functionalities to view grades, attendance, download and submit assignments with file uploads, submit leave applications for dual approval, attempt time-bound online exams with auto-evaluation, attend scheduled online classes, participate in chat forums, receive prioritized notifications, and access event details through the integrated academic calendar.
- Parent functionalities to track their child’s academic performance, attendance records, academic calendar events, receive priority notifications, and participate in the leave approval workflow.
- Leave approval workflow that ensures coordinated decision-making between Teachers and Parents before a student's leave request is finalized.
- Academic calendar management to allow the Admin to define events, holidays, exams, and lecture schedules visible across all dashboards.
- Chat forums for academic discussions and peer-to-peer communication, and an integrated chatbot for user guidance and quick navigation assistance.
- Priority-based notification system to ensure important messages are delivered and highlighted according to urgency.
- Online meeting and lecture functionality for real-time virtual interaction between teachers and students.
- Responsive and user-friendly interface developed using React.js to ensure easy access and usability across devices.
- Profile management for all users, displaying essential personal and academic information.
- Public-facing features like the “About Us” and “Contact Us” pages to provide project context, team member information, and institute contact details.

- Secure storage and handling of user data including password hashing, role-based API access, and account-level authentication.

- Scalable backend architecture built using Spring Boot, Spring Security, Hibernate and .NET Core Web API and MySQL, supporting RESTful services and future system expansion.

This scope ensures that the ePathshala (Student ERP System) provides a robust foundation for academic management while maintaining usability, data integrity, and security.

1.4 References

- https://spring.io/
- https://www.javascript.com/
- https://developer.mozilla.org/en-US/docs/Web/JavaScript
- https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93controller
- Spring Boot + React: JWT Authentication with Spring Security - BezKoder

2. Overall Description

2.1 Product Perspective

The ePathshala (Student ERP System) is designed to address the limitations and inefficiencies associated with traditional paper-based or fragmented digital systems used for managing student information and academic workflows. As educational institutions expand in size and complexity, there is a growing need for a centralized, role-based platform that streamlines academic operations, fosters transparency, and enables modern, interactive learning experiences between students, teachers, parents, and administrators.

The system serves as a comprehensive academic and administrative management platform, enabling institutions to automate essential processes such as grade tracking, attendance management, assignment distribution with file upload capabilities, leave application approvals, and time-bound online examinations with auto-evaluation and auto-submission. It also supports real-time online meetings and lectures, interactive chat forums for academic discussions, a chatbot for instant user guidance, an integrated academic calendar for event and schedule visibility, and a priority-based notification system to ensure timely communication of important updates.

By offering separate dashboards for each user role—Admin, Teacher, Student, and Parent—the system ensures that every stakeholder has access to the tools and data relevant to their responsibilities, from managing institutional schedules to participating in live classes and collaborative forums.

This ERP solution enhances institutional efficiency by:

- Reducing manual efforts and paperwork through automation of academic and administrative workflows,
- Improving communication and collaboration through chat forums, real-time notifications, and online meeting capabilities,
- Increasing the accuracy and timeliness of academic records through features like auto-evaluated exams and centralized data management,
- Offering a unified academic calendar for transparent scheduling of events, lectures, and examinations,
- Providing responsive user interfaces and guidance tools like the integrated chatbot to improve navigation and accessibility.

By integrating secure authentication, centralized data handling, and an intuitive user experience, the ePathshala (Student ERP System) significantly improves the educational experience for all stakeholders, making academic administration more efficient, interactive, and future-ready.

2.2 Product Features

The following are the key features and requirements of the ePathshala (Student ERP System):

- **User Authentication and Role-Based Authorization:**
  The system supports secure login functionality for all users—Admin, Student, Teacher, and Parent. Authentication is managed through JWT (JSON Web Token), and access to features is strictly role-based, ensuring that users only see and interact with the components relevant to their role.

- **Student Management (Admin Role):**
  Admin users have the ability to add, view, edit, and delete student records. While adding a student, the admin also links a parent account. Each student is assigned a unique account number and associated with a specific class.

- **Teacher and Parent Management (Admin Role):**
  Admins can register teacher and parent users. Teachers are assigned to specific classes and subjects, which directly reflect on their dashboards. Admins maintain full control over all user records, including deletion.

- **Academic Calendar Setup(Admin Role):**
  Admins can create and manage the academic calendar, including holidays, events, exam dates, online lecture schedules, and institutional activities. This calendar is viewable by all roles—students, teachers, and parents—ensuring a unified view of important dates.

- **Grade Management (Teacher Role):**
  Teachers can enter subject-wise marks for students. These grades are then viewable on the student and parent dashboards. Teachers can also view class-wise average performance.

- **Attendance Management (Teacher Role):**
  Teachers mark daily attendance for each subject. This data is visible to students and parents in their respective dashboards, ensuring real-time attendance transparency.

- **Assignment Upload and Submission Workflow with File Uploads (Teacher Role):**
  Teachers can upload assignments with titles, due dates, file attachments, and class-subject mapping. Students can download these assignments and submit their completed work by uploading files directly to the system.

- **Online Exam Module with Auto-Evaluation:**
  Students can attempt online exams within the platform. The system automatically evaluates objective-type questions and instantly generates results, improving assessment speed and accuracy.

- **Time-Bound Exams with Auto-Submission:**
  Exams have a predefined duration, and the system automatically submits answers when the time expires, ensuring fairness and adherence to schedules.

- **Online Meetings and Live Lectures:**
  Teachers can schedule and conduct live online classes or meetings directly from the system, allowing students to join from their dashboard without third-party integrations.

- **Chat Forums:**
  Role-based chat forums allow students, teachers, and parents to engage in topic-specific discussions, share ideas, and clarify doubts in a moderated, organized environment.

- **Chatbot for User Guidance:**
  An integrated chatbot assists users with navigation, feature usage, and basic troubleshooting, ensuring a smooth experience for both new and returning users.

- **Notification System with Priority Sequence:**
  The system delivers announcements, reminders, and urgent alerts with assigned priority levels so that critical updates are always highlighted first on user dashboards.

- **Leave Application and Approval Workflow:**
  Students can submit leave requests, which must be approved by both the assigned teacher and parent. The system tracks individual approval statuses and computes the

final decision based on both responses.

- **Profile Management (All Roles):**
  Each user can view their profile, which includes name, role, account number, email, and associated class or subject (if applicable).

- **Performance Reports:**
  The system supports downloadable academic reports (e.g., grade reports in PDF format) from the student dashboard for personal tracking or institutional purposes.

- **Public Pages: About Us & Contact Us:**
  The landing page includes an "About Us" section that highlights system features and introduces the team behind ePathshala. A "Contact Us" page provides institute contact information, a feedback form, and an integrated Google Map for easy location access.

- **Secure Data Storage and Privacy Compliance:**
  The system ensures secure password hashing, role-based access control, and encrypted data transfers. All personal and academic data is handled in compliance with data privacy best practices.

2.3 User Classes and Characteristics

The ePathshala (Student ERP System) supports four main user classes:

1. **Students:**
   - **Characteristics:**
     Students are one of the primary users of the system. They interact with the platform to monitor their academic progress, participate in online assessments, attend live lectures, collaborate in chat forums, and access institutional updates.
   - **Requirements:**
     Students must be able to log in securely and access their personalized dashboard. They should be able to:
     - View subject-wise grades and performance reports.
     - Track daily and monthly attendance.
     - Download uploaded assignments and submit completed work through file uploads.
     - Attempt online exams with auto-evaluation and time-bound auto-submission.
     - Join scheduled online lectures or meetings with teachers.
     - Participate in chat forums for subject discussions and peer collaboration.
     - Seek help from the integrated chatbot for navigation and usage guidance.
     - Receive notifications with priority-based ordering for important updates.
     - View the academic calendar with events, exam dates, and lecture schedules.
     - Export their grades in PDF format.

2. **Teachers:**
   - **Characteristics:**
     Teachers are responsible for managing academic data for students in their assigned classes and subjects, conducting assessments, and facilitating online interactions.
   - **Requirements:**
     Teachers must have access to their dashboard to:

- View the list of students in their class.
- Enter grades for each subject.
- Mark daily attendance.
- Upload assignments with file attachments and due dates.
- Review and grade uploaded student submissions.
- Schedule and conduct live online lectures or meetings.
- Create and manage online exams, configure time limits, and utilize auto-evaluation.
- Approve or reject student leave applications.
- Participate in chat forums to answer queries and guide students.
- Send high-priority notifications to students and parents when needed.
- View class average performance and analytics.

3. **Parents:**
   - **Characteristics:**
     Parents are linked to specific student accounts and use the system to monitor their child's academic performance, attendance, and institutional participation.
   - **Requirements:**
     Parents should be able to log in and view their child's:
     - View their child's subject-wise grades and attendance records.
     - Access the academic calendar to stay updated on exams, lectures, and events.
     - Approve or reject their child's leave requests.
     - Receive and prioritize notifications about important academic activities.
     - Join online parent-teacher meetings if scheduled.
     - Participate in relevant chat forums for parent–teacher discussions.
     - Use the chatbot for quick assistance with system navigation.

4. **Admin:**
   - **Characteristics:**
     Admins are the system's primary administrators. They have full access to user data, configuration settings, and institutional-level features.
   - **Requirements:**
     Admins must be able to:
     - Create and manage accounts for students, teachers, and parents.
     - Assign teachers to subjects and classes.
     - Set up and maintain the academic calendar with events.
     - Schedule system-wide notifications with priority levels.
     - Perform CRUD operations on user data.
     - Moderate chat forums and manage chatbot content.
     - View institutional metrics, analytics, and system summaries.
     - Delete user accounts when necessary.

2.4 Operating Environment

The operating environment for the ePathshala (Student ERP System) consists of the following hardware and software components:

Hardware Requirements:

- A development machine with at least 8GB of RAM and a multi-core processor (e.g., Intel Core i5 or higher) to ensure smooth development and testing processes.

- Stable internet connection for accessing third party APIs, and performing version control operations.

Software Requirements:

Frontend:

- React.js (JavaScript-based, no TypeScript) for building interactive and responsive user interfaces.

Backend:

- Spring Boot, Spring Security, Hibernate, and .NET Core Web API with Entity Framework Core for implementing RESTful services and database operations.

Database:

- MySQL as the primary relational database management system for storing user data, attendance records, assignments, grades, and more.

Authentication:

- JWT (JSON Web Token) for implementing secure role-based authentication and stateless session management.

Development Tools & Applications:

- Visual Studio 2022 – Backend development (.NET Core Web API).
- Spring Tool Suite (STS) – Backend development using Spring Boot, Spring Security, and Hibernate.
- Visual Studio Code – Frontend development (React.js).
- Git – Version control for collaborative development using platforms like GitHub or GitLab.
- Command Line Interface (CLI) – For project setup, dependency management, and execution.
- Google Chrome (or any modern web browser) – Running and testing the web application.
- Postman – Testing and validating API endpoints during backend development.
- MySQL Workbench – Managing database schema and executing SQL queries.
- (Optional) AWS or Microsoft Azure – For scalable cloud hosting and deployment of the final application.Visual Studio 2022 / Visual Studio Code for backend and frontend development, respectively.

2.5 Design and Implementation Constraints

- **Single Language Support:**
  The user interface of ePathshala (Student ERP System) is available only in English. No multilingual support is implemented in this version.

- **Frontend–Backend Technology Mapping:**
  The project uses React.js for the frontend, Spring Boot, Spring Security, Hibernate, and .NET Core Web API with Entity Framework Core for the backend. Compatibility and integration challenges may arise while connecting these two technologies, especially in managing request formats, authentication headers (e.g., JWT), and role-based routing.

- **Concurrency Limitations:**
  Performing multiple operations simultaneously (e.g., concurrent submissions of leave requests or bulk grade entries) may cause a temporary load on the server, potentially

impacting performance if not handled with proper optimization and asynchronous processing.

- **HTTPS Only:**
  The application is restricted to HTTPS for secure communication. All API calls and data transfers must occur over encrypted connections to ensure data integrity and user privacy.

2.6 User Documentation

The user documentation for the ePathshala (Student ERP System) is designed to provide clear and accessible guidance to all user roles—students, faculty, and administrators—on how to navigate and operate various modules of the system.

Help Section (In-App):

- A dedicated Help section is integrated into the application.
- It provides step-by-step instructions for common tasks such as:
  - Logging in and logging out.
  - Viewing and marking attendance
  - Viewing results and grades
  - Requesting leave
  - Uploading and managing assignments
- The Help section is role-specific, ensuring that each user type sees guidance relevant to their permission.

Reference Documentation (SRS Report):

This Software Requirements Specification (SRS) serves as a comprehensive reference manual for both end users and developers, including:

- Overview of the project and objectives
- Detailed system functionality and workflows
- Role definitions and user capabilities
- Technologies and tools used in development
- Hardware and software requirements
- Operating environment details
- System design and implementation constraints

Purpose:

- Ensures end users can efficiently operate the system without prior training.
- Assists developers and maintainers in understanding the system's architecture, constraints, and intended behavior.

2.7 Assumptions and Dependencies

Assumptions:

- **Basic Technical Literacy:**
  Students, faculty, and administrators possess basic knowledge of operating web-based applications.Students, faculty, and admins have basic knowledge of using web-based applications.

- **Device and Internet Availability:**
  All users have access to internet-enabled devices such as laptops, desktops, or mobile phone.

- **Administrative Responsibility:**
  Admin users are responsible for creating, updating, and managing student and faculty accounts.

- **Faculty Data Management:**
  Faculty members will regularly update academic information, including attendance, assignments, and examination results.

- **Student Engagement:**
  Students will log in frequently to check updates, view results, request leaves, and complete other academic activities.

Dependencies:

- **Database Availability:**
  MySQL Server must be running and accessible to handle all database operations such as storing user information, attendance records, and academic results.

- **Authentication Security:**
  JWT-based authentication depends on secure token generation, validation, and expiration management to ensure secure user sessions.

- **Technology Stack Configuration:**
  Proper configuration of React.js (frontend), Spring Boot, Spring Security, .NET Core Web API (backend), and MySQL database is essential for seamless frontend–backend integration.

- **API Compatibility:**
  Consistent compatibility between frontend requests and backend REST API responses must be maintained throughout the development lifecycle. Compatibility between frontend and backend REST APIs must be maintained throughout development.

3. System Features

3.1 User Login

3.1.1 Description and Priority

- Description:
  All users (Admin, Student, Teacher, Parent) can log in using their credentials and selected role. JWT is used to manage session and enforce role-based access.
- Priority : High

3.1.2 Stimulus/Response Sequences

- User navigates to login page.
- Enters email, password, and selects role.
- On valid credentials, user is redirected to their specific dashboard.

3.1.3 Functional Requirements

- JWT-based authentication with role claims.
- Secure password hashing.
- Role selection and validation.
- Session expiration and logout functionality.

3.2 Dashboard

3.2.1 Description and Priority

- Description:
  Each user sees a role-specific dashboard upon login showing personalized data and actions.
- Priority: High

3.2.2 Stimulus/Response Sequences

- User logs in → System displays dashboard relevant to their role (Admin, Student, Teacher, Parent).

3.2.3 Functional Requirements

- Admin dashboard: Manage users, assign roles, view statistics.
- Student dashboard: View grades, attendance, assignments, leave status, attempt exams.
- Teacher dashboard: Manage student data, grades, attendance, leaves and add exams.
- Parent dashboard: View child's data and approve leaves.

3.3 Assignment Management

3.3.1 Description and Priority

- Description:
  Teachers can upload assignments; students can view and download them.
- Priority: High

3.3.2 Stimulus/Response Sequences

- Teacher uploads assignment → It becomes visible to assigned class students.

3.3.3 Functional Requirements

- File upload with subject/class tagging.

- Students can view/download assignments.
- Due date validation.

3.4 Attendance Management

3.4.1 Description and Priority

- Description:
  Teachers mark attendance, which becomes visible to students and parents.
- Priority: High

3.4.2 Stimulus/Response Sequences

- Teacher selects class and marks daily attendance → Data is updated and reflected in Student/Parent dashboards.

3.4.3 Functional Requirements

- Attendance marking by date and subject/class.
- Monthly attendance summary for students/parents.
- Export options (PDF).The system should allow reports to be viewed, downloaded.

3.5 Grade Management

3.5.1 Description and Priority

- Description: Teachers enter subject-wise grades for students, viewable by students and parents.
- Priority: High

3.5.2 Stimulus/Response Sequences

- Teacher inputs grades → Student and Parent can view updated results.

3.5.3 Functional Requirements

- Grade entry interface for teachers.
- Role-based viewing permissions.
- Optional: Export to PDF.

3.6 Leave Management

3.6.1 Description and Priority

- Description:
  Students submit leave requests, which must be approved by both Teacher and Parent.
- Priority: High

3.6.2 Stimulus/Response Sequences

- Student applies for leave → Teacher and Parent receive request → Once both approve → Final status = "Approved".

3.6.3 Functional Requirements

- Leave form with start/end dates and reason.
- Approval flow with status tracking.
- Notification system for leave status updates.

3.7 User Management (Admin Only)

3.7.1 Description and Priority

- Description:
  Admin manages all user accounts including Students, Teachers, and Parents.
- Priority: High

3.7.2 Functional Requirements

- Add, View, Edit, Delete users.
- Assign Teacher to Subject/Class.
- Parent-Student linkage logic.

3.8 Academic Calender

3.8.1 Description and Priority

- Description:
  Admin sets academic calendar entries like holidays, events visible to Students and Teachers.
- Priority: Medium

3.8.2 Functional Requirements

- Add calendar events with title, date, type.
- Events shown in dashboards of Students and Teachers.

3.9 Profile Management

3.9.1 Description and Priority

- Description:
  Every user can view their profile with role-specific data.
- Priority: Medium

3.9.2 Functional Requirements

- Display: Name, Role, Email and other related information.
- Additional: Class (Student), Assigned Subjects (Teacher), Child ID (Parent).

3.10 About Us and Contact Us

3.10.1 Description and Priority

- Description:
  Informational pages for users and external visitors..
- Priority: Low

3.10.2 Functional Requirements

- About Us: ERP description, team cards.
- Contact Us: Institute details, feedback form, embedded Google Map.

4. External Interface Requirements

4.1 User Interfaces

The system will have a role-based and responsive web interface designed using React.js. Each type of user (Admin, Student, Teacher, Parent) will have a separate dashboard based on their role:

- Login Page: Common for all users. Users will enter their credentials to access their dashboard.
- Admin Dashboard: Manage students, teachers, parents, academic calendar, and role assignment.
- Student Dashboard: View attendance, grades, assignments, announcements, apply for leave and attempt exams.
- Teacher Dashboard: Upload grades, mark attendance, assign homework, handle leave requests and add exams.
- Parent Dashboard: View child's academic performance and approve/reject leave applications.
- Responsive Design: Optimized for both desktop and mobile viewports using CSS and Bootstrap.

4.2 Hardware Interfaces

- The system will run on standard computing devices (desktops, laptops, tablets, or smartphones).
- (Optional) Server Deployment will require a cloud server (such as AWS EC2 or Azure) or local host with:
    - Minimum 8 GB RAM
    - Quad-core processor
    - SSD storage (preferred for performance)
    - Stable Internet connectivity

4.3 Software Interfaces

- Frontend: React.js (JavaScript), Axios for HTTP calls, Bootstrap for styling.
- Backend API: Spring Boot, Spring Security, Hibernate, .NET Core Web API.
- Database: MySQL Server.
- Authentication: JWT (JSON Web Token)-based role-specific authentication.
- Version Control: GitHub for source code repository.
- Middleware & Tools:
    - Entity Framework Core for DB interaction
    - Postman for API testing

- Visual Studio / VS Code for development

4.4 Communications Interfaces

The Student ERP System requires secure and efficient communication between frontend, backend, and external systems. The communication interfaces include the following:

4.4.1. Client-Server Communication

- Protocol: HTTPS (HTTP Secure) to ensure encrypted communication.
- Frontend to Backend:
  - Communication will occur via RESTful APIs using JSON data format.
  - All API requests (GET, POST, PUT, DELETE) will include a JWT token in the header for user authentication and role verification.

4.4.2. Database Communication

- The backend (.NET Core Web API) will interact with the MySQL database using Entity Framework Core.
- All database queries and updates will be handled via LINQ and SQL commands securely.

4.4.3. Authentication Service

- The system uses JWT-based authentication for secure role-based access.
- Tokens will be stored on the client side and attached to every API request after login.

4.4.4. Notifications

- In-app Notifications for assignment submission, leave approval, etc. will be managed via backend push responses or polling from the frontend.

4.4.5. Version Control and Collaboration

- Developers will collaborate through GitHub, using Git protocol for:
  - Source code management
  - Branch handling
  - Pull requests and code reviewsThe HTPP or HTTPS protocol(s) will be used to facilitate communication between the client and server.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

- The system should load the dashboard and important pages (assignments, attendance, leave requests) within 2 seconds under normal conditions.
- The backend API should respond to user requests within 500 milliseconds on average.
- Pagination and lazy loading will be implemented where necessary to handle large datasets (e.g., student records, assignment lists).

5.2 Safety Requirements

- The system must regularly back up the database to prevent data loss.
- Critical actions such as data deletion or role assignment will prompt confirmation dialogs to prevent accidental operations.
- The system should log all major actions (login, submission, approvals) for future traceability and issue resolution.

5.3 Security Requirements

- JWT Authentication will be implemented to ensure secure access control and session handling.
- Role-based access control (RBAC) will ensure users (Admin, Student, Teacher, Parent) only access authorized modules.
- Passwords will be hashed and salted using secure algorithms (e.g., BCrypt) before storing in the database.
- All API endpoints will be protected from SQL Injection, and CSRF vulnerabilities.
- Sensitive data such as student personal details and academic records will be transmitted only over HTTPS.

5.4 Software Quality Attributes

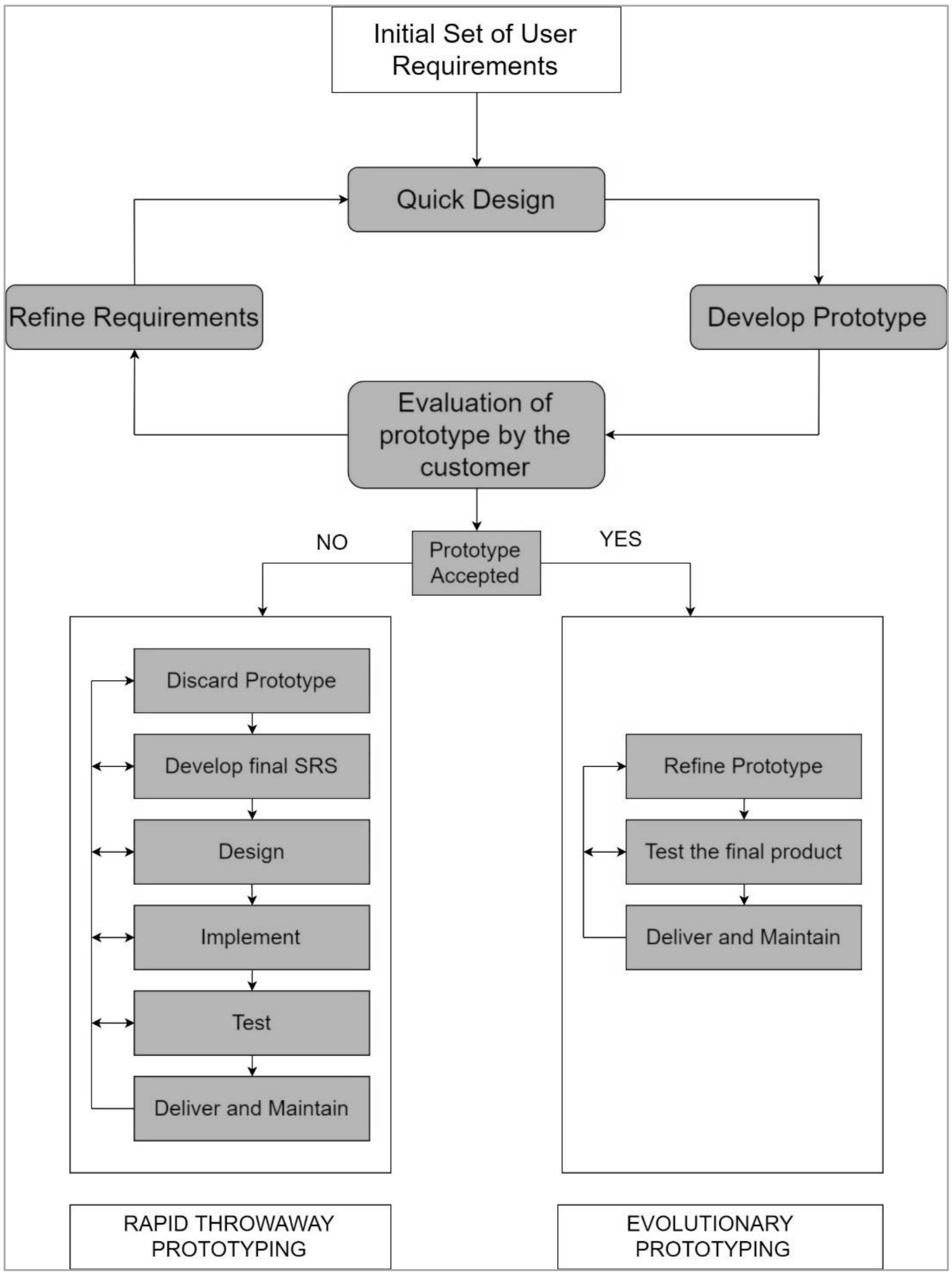
- **Usability:** The system will have an intuitive UI, accessible to both tech-savvy and non-technical users (e.g., parents).
- **Maintainability:** Code will follow best practices and be modular to facilitate future updates.
- **Scalability:** The application architecture will support scaling both vertically and horizontally to handle increased load.
- **Availability:** The system is expected to have 99.5% uptime during academic hours (9 AM – 6 PM).
- **Reliability:** The application will undergo testing for edge cases and will handle failures gracefully with meaningful error messages.

Appendix A: Glossary

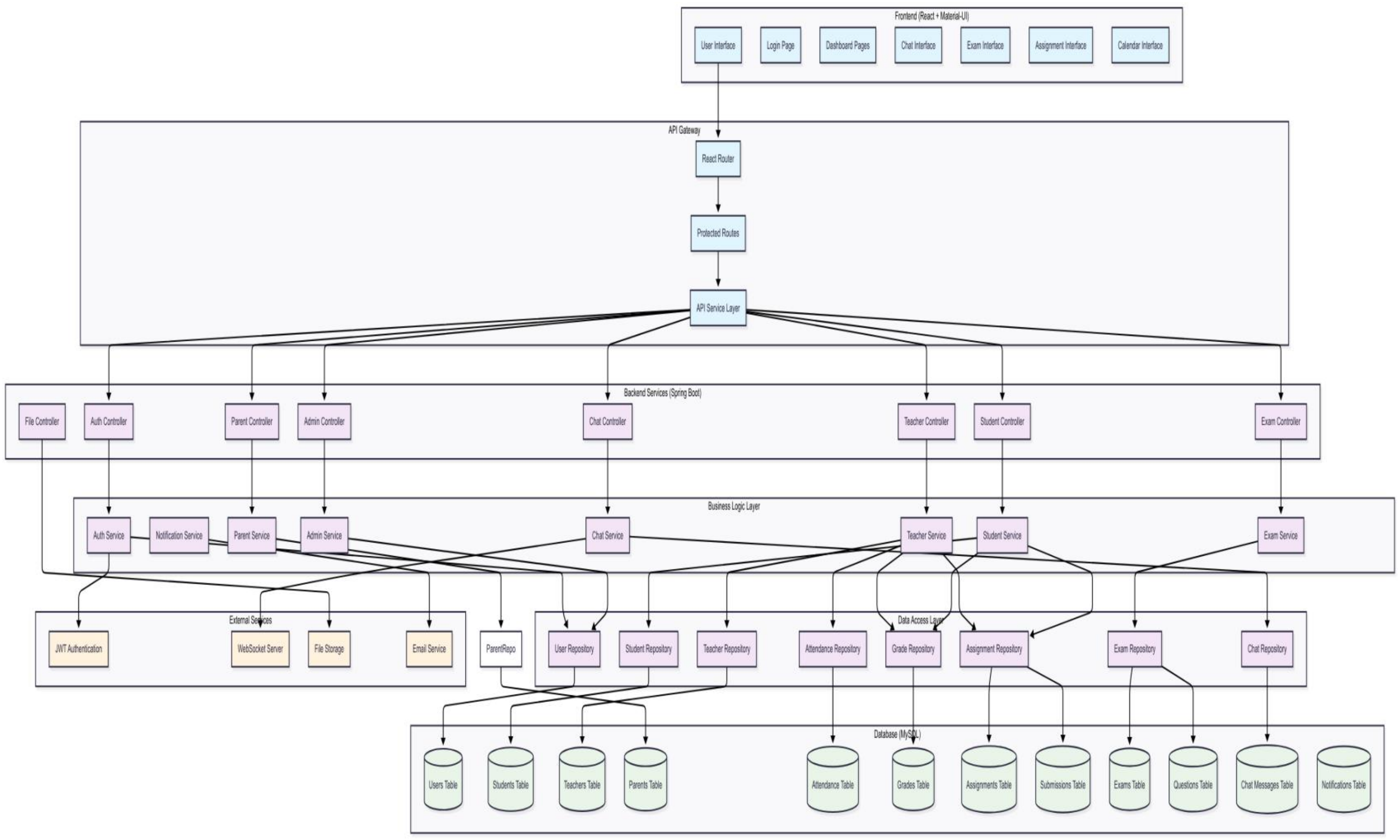
Sr. No.	Abbreviation	Full Form
1.	API	Application Programming Interface
2.	AWS	Amazon Web Service
3.	CLI	Command line Argument
4.	GB	Gigabyte
5.	HTML	Hypertext Markup Language
6.	HTTP / HTTPS	Hypertext Transfer Protocol / Hypertext Transfer Protocol Secure
7.	ID	Identification
8.	JS	JavaScript
9.	JWT	Java Web Token
10.	OS	Operating System
11.	RAM	Random Access Memory
12.	SQL	Structured Query Language
13.	SRS	Software Requirement Specification
14.	URL	Uniform Resource Locator
15.	ER	Entity Relationship

Appendix B: Analysis Models

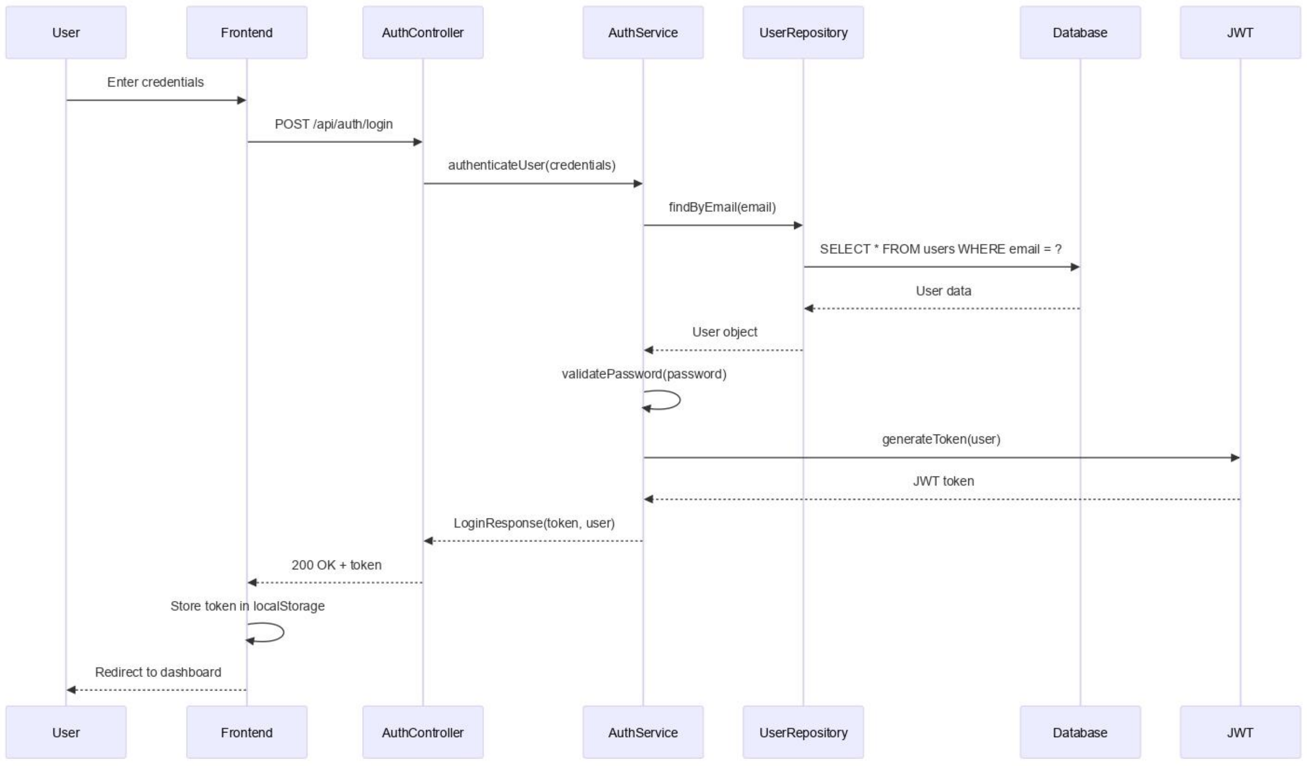
- **Software Development Approach in Our System**



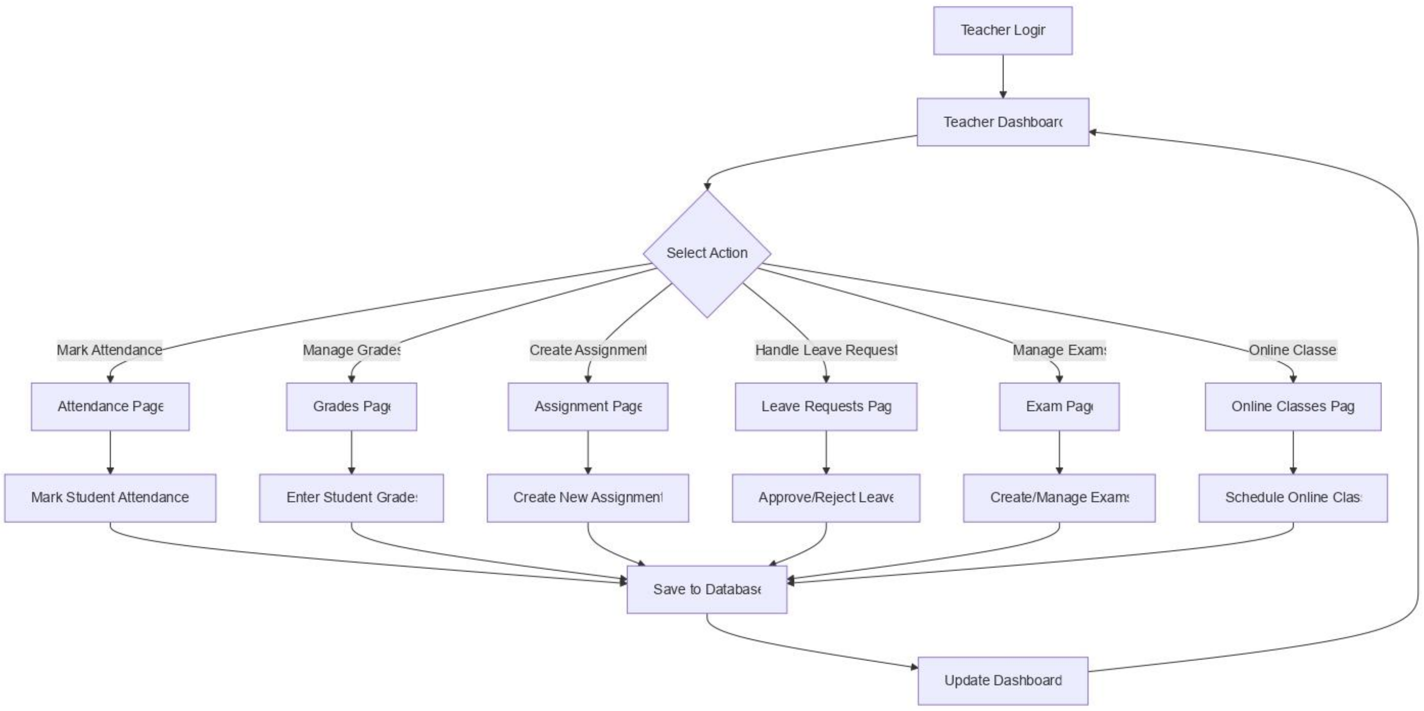
- **Complete System Architecture Flow Diagram:**



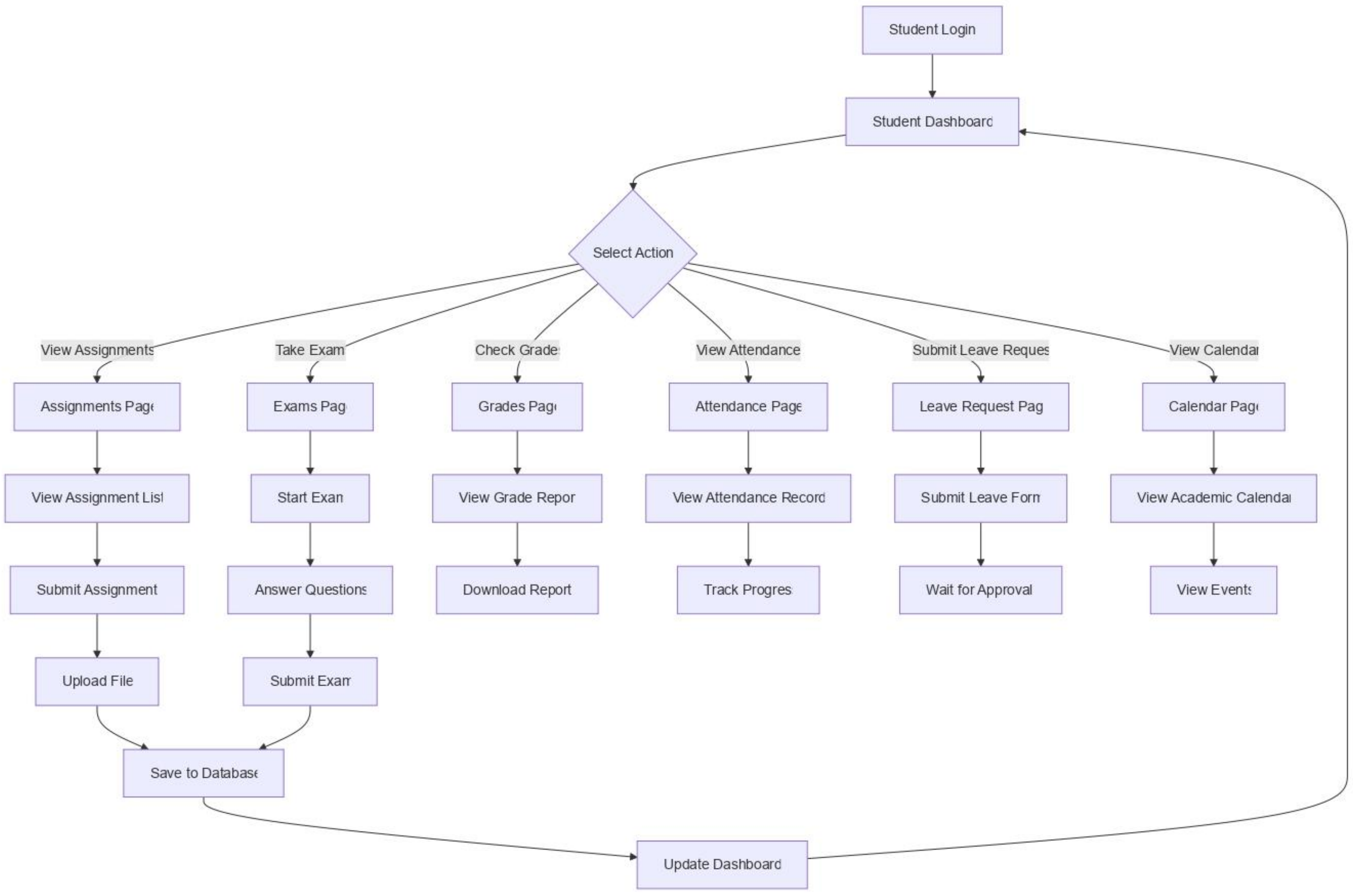
- **User Authentication Flow Diagram**



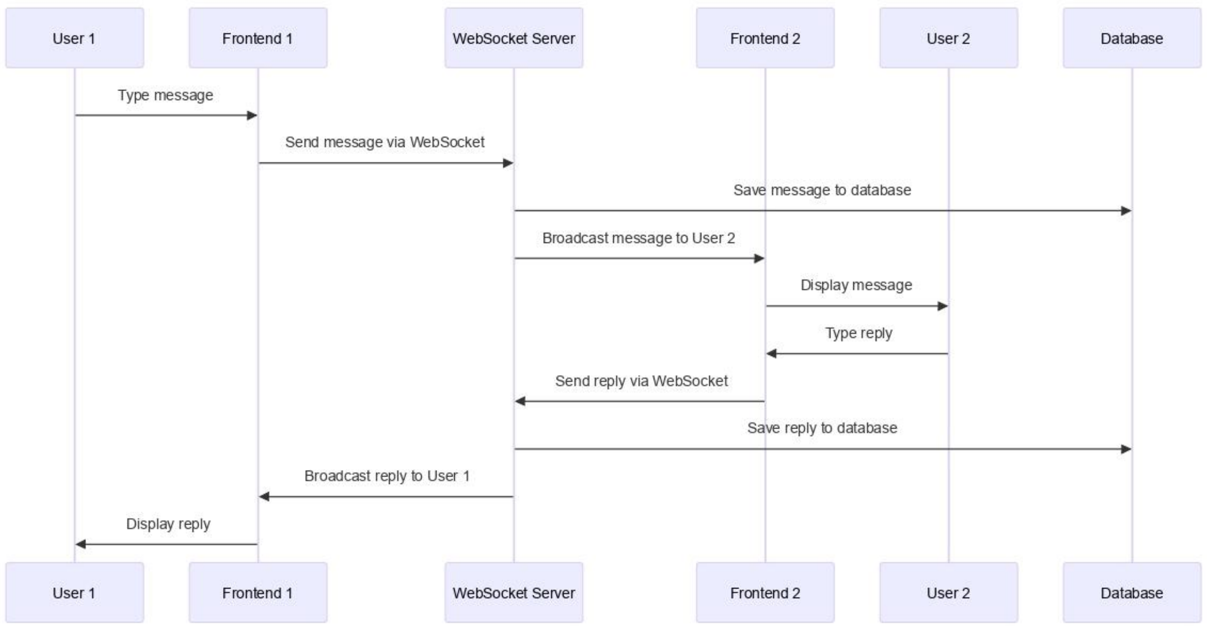
- ## Teacher Dashboard Flow Diagram



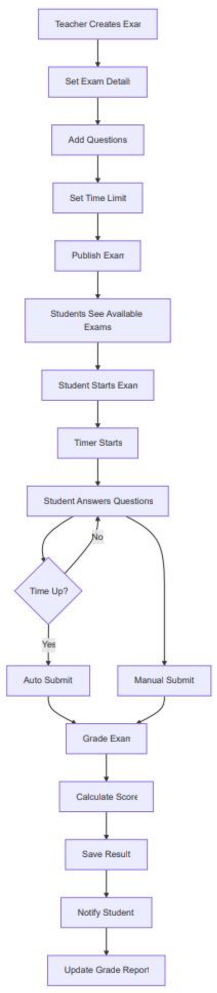
- ## Student Dashboard Flow Diagram



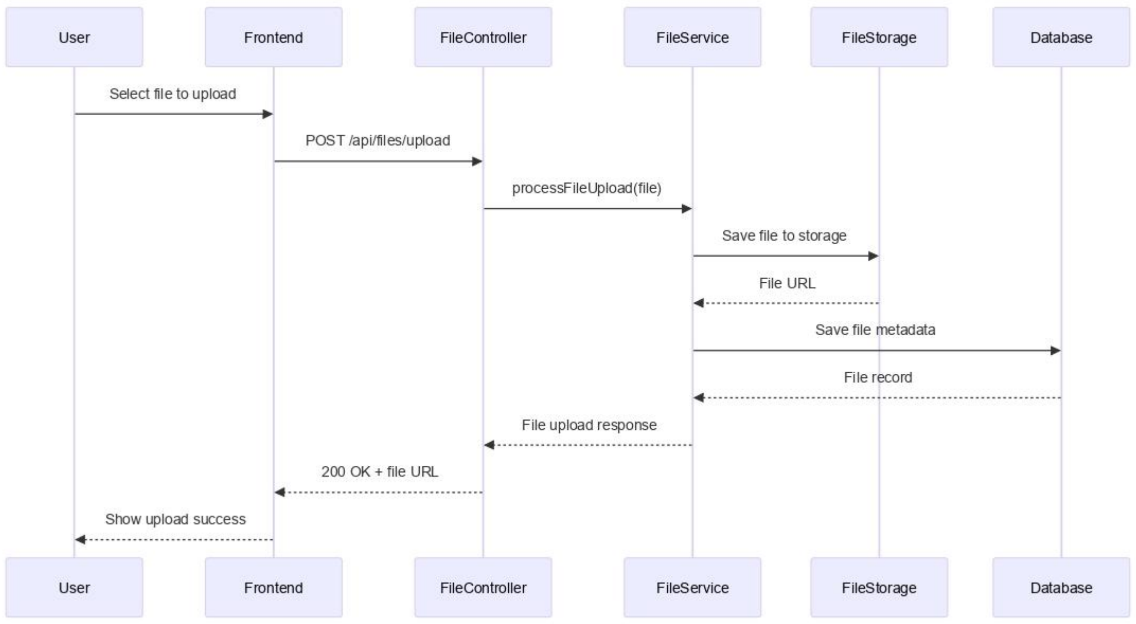
- **Real Time Chart Flow Diagram**



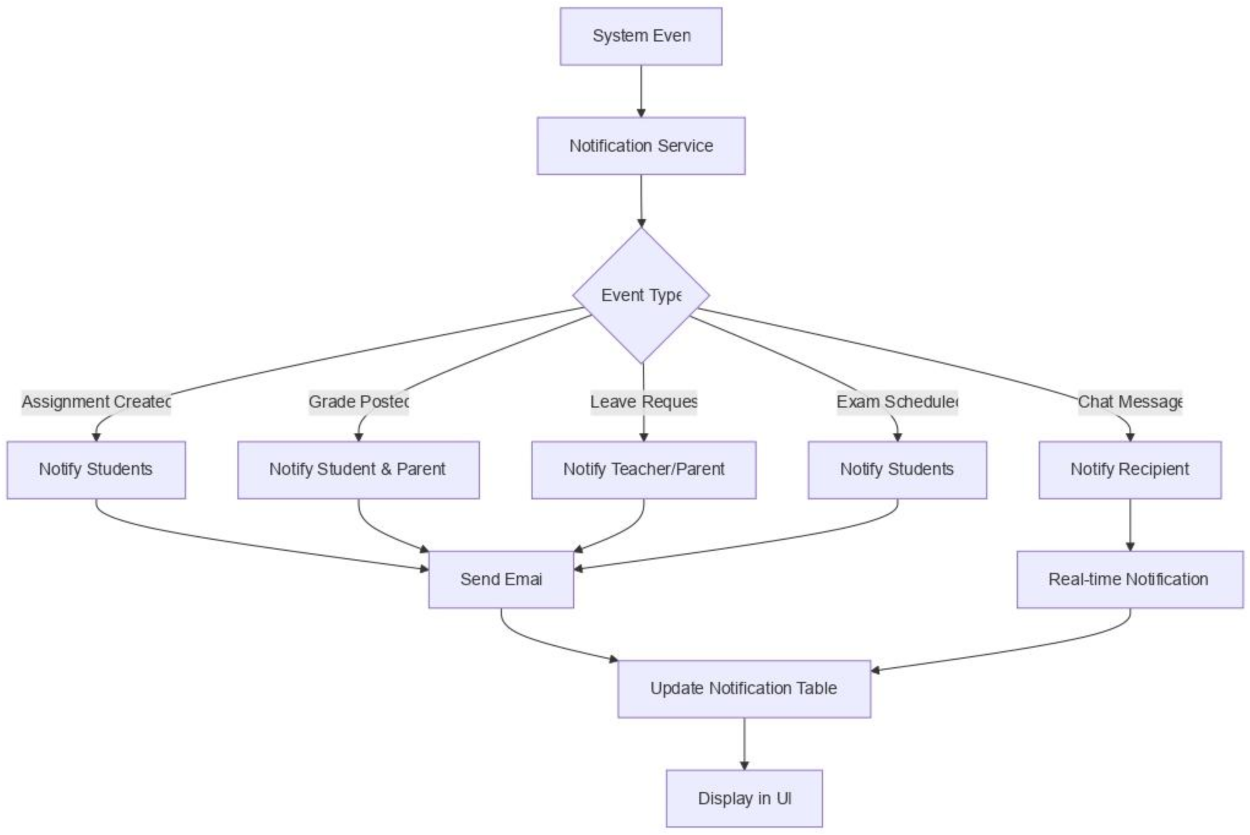
- **Exam Management Flow Diagram**



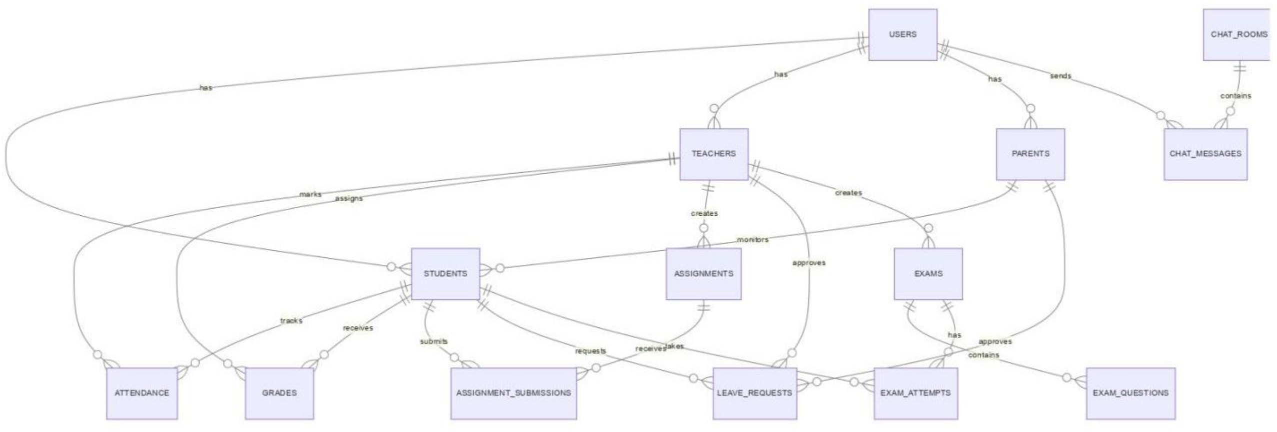
- ## File Upload Flow Diagram



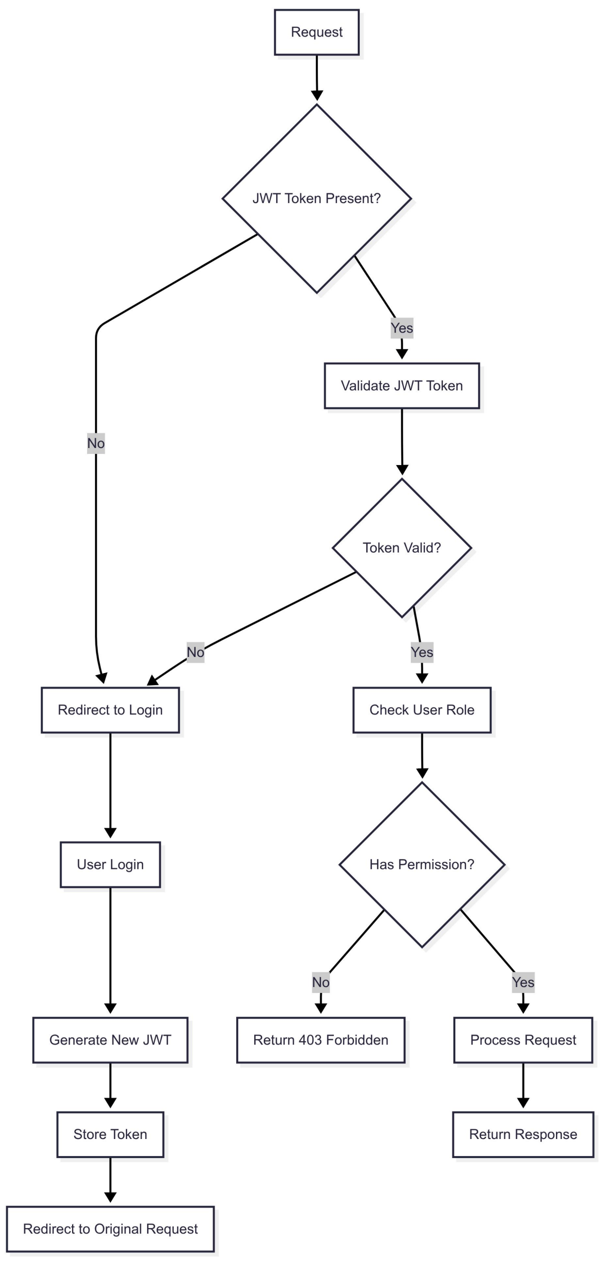
- ## Notification System Flow Diagram



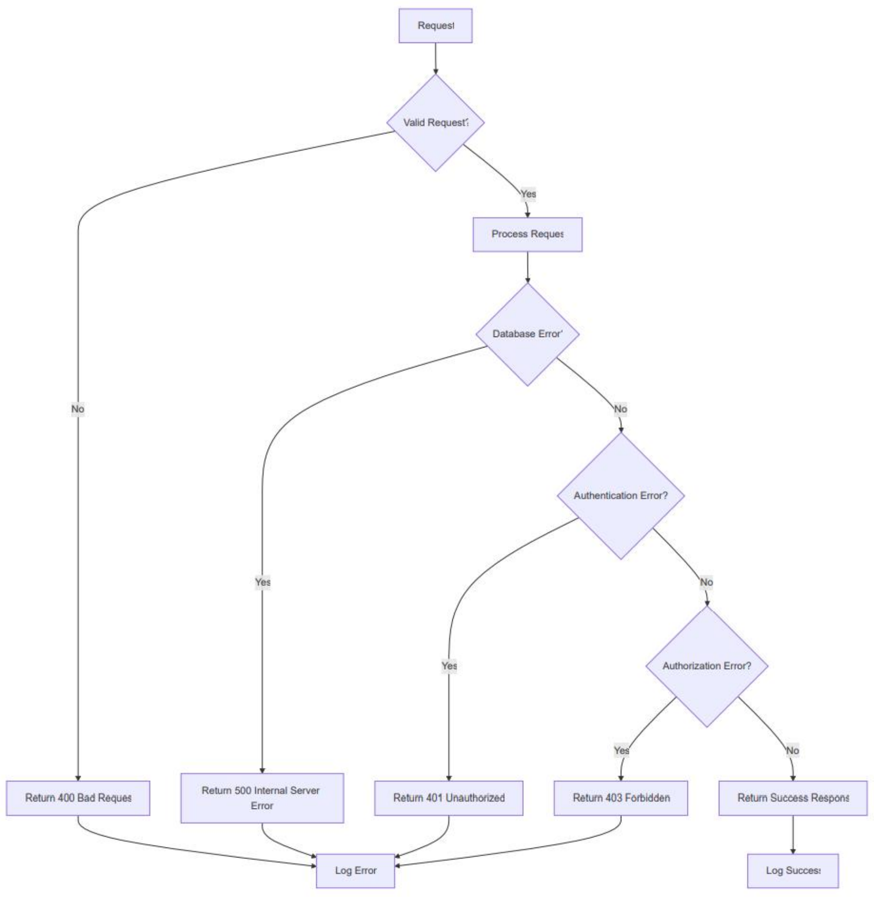
- ## Database Schema Relationships Diagram



- ## Security Flow Diagram



• Error Handling Flow Diagram


Request
Valid Request?
Yes
No
Process Reques
Database Error?
No
Yes
Authentication Error?
No
Yes
Authorization Error?
Yes
No
Return 400 Bad Reques
Return 500 Internal Server Error
Return 401 Unauthorized
Return 403 Forbidden
Return Success Respons
Log Error
Log Succes