Online Notes Sharing System

# Complete Project Documentation

## Software Engineering Diagrams & Analysis

**Project Overview**

The Online Notes Sharing System is a comprehensive web application designed to facilitate academic collaboration among students. The system allows students to upload, browse, and download academic notes while providing administrators with robust content management capabilities. Built using Flask backend with

SQLite database, the application ensures secure authentication, file management, and quality control through an approval workflow.

**Key Features:**

User registration and authentication

Notes upload with metadata management Browse, search, and download functionality Administrative review and approval system Secure file storage and handling

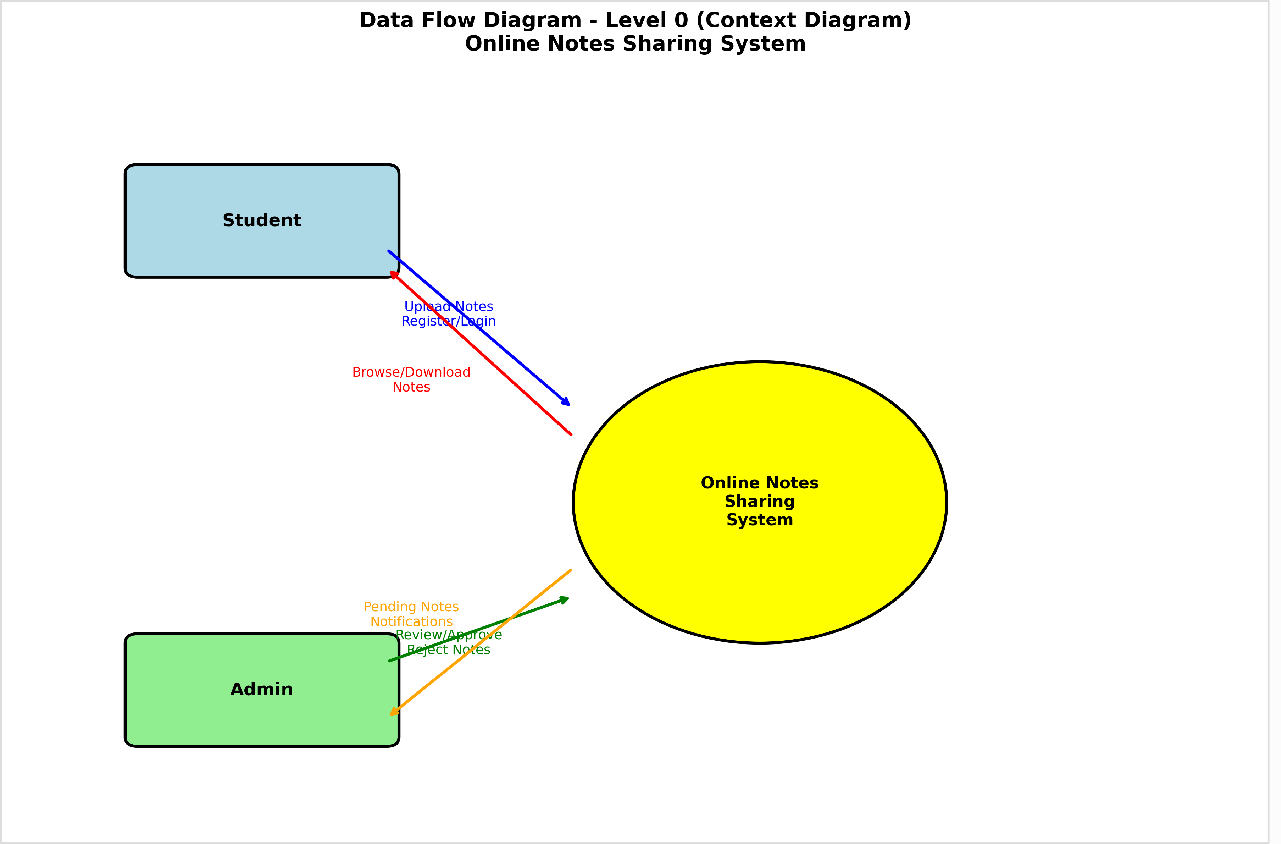
# Table of Contents

* Data Flow Diagrams (Level 0 and Level 1)
* Event Table
* System Architecture
* Use Case Diagrams
* Class Diagram
* Entity-Relationship (E-R) Diagram
* Schema Design
* State Machine Diagram
* Activity Diagram
* System Flow Chart

# Data Flow Diagrams

**Level 0 Data Flow Diagram**

The Level 0 DFD provides a high-level view of the Online Notes Sharing System, showing the primary data flows between external entities (Students and Admins) and the system as a single process.



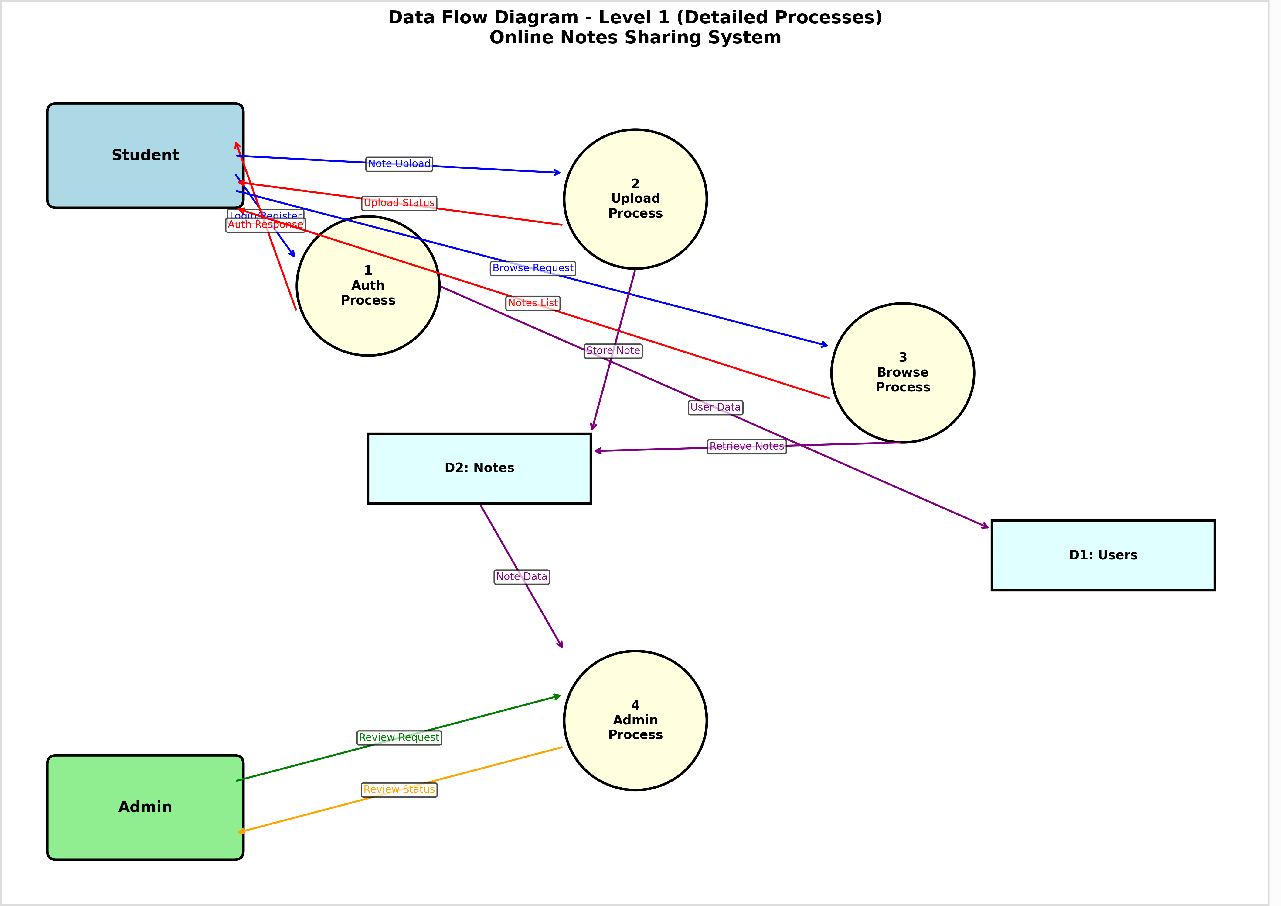
**Key Points:**

Shows external entities: Students and Admins

Illustrates main data flows: note uploads, downloads, and administrative actions Demonstrates the system boundary and external data stores

**Level 1 Data Flow Diagram**

The Level 1 DFD breaks down the system into detailed processes, showing how data flows between authentication, note management, file storage, and administrative functions.



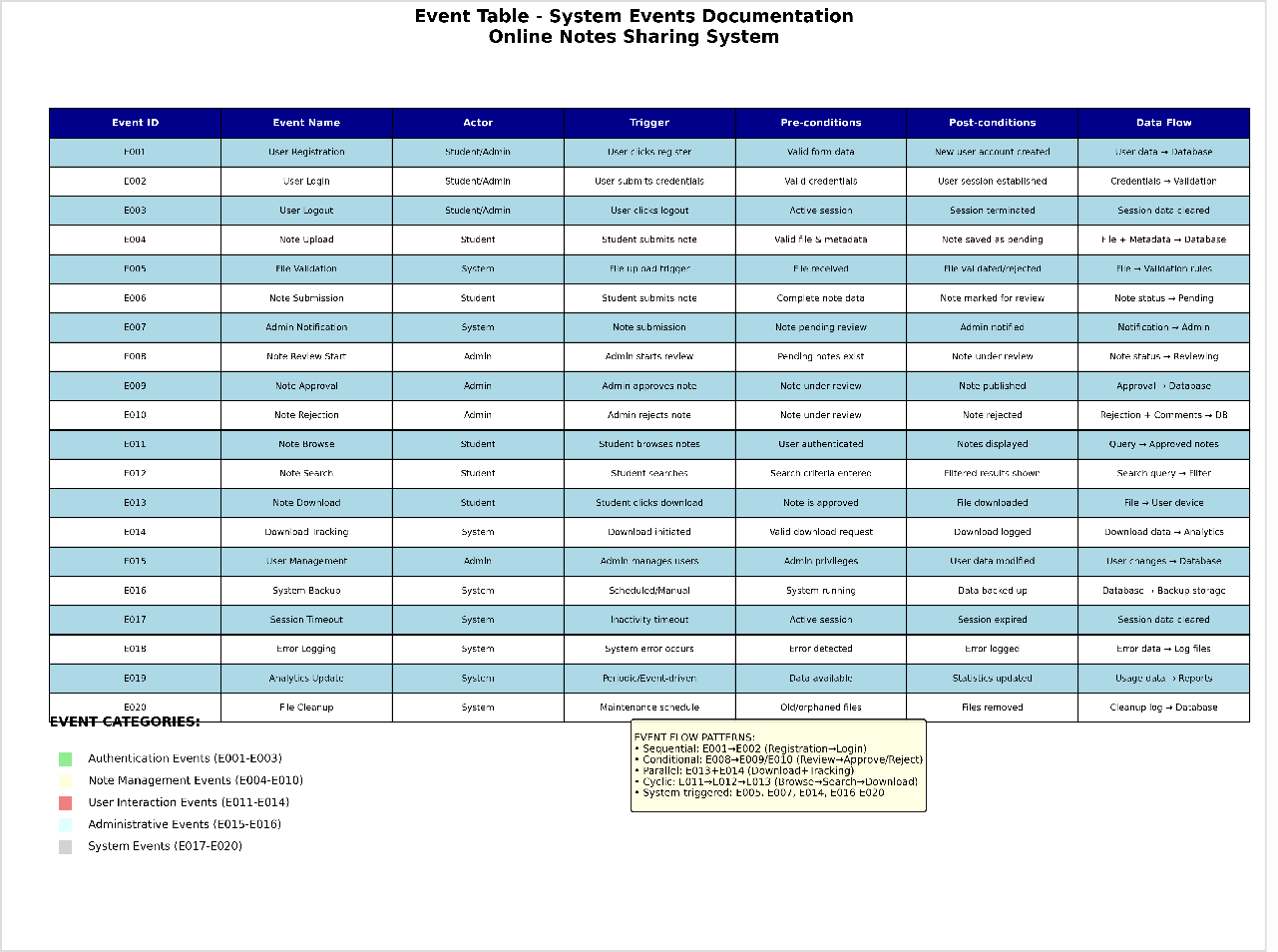
**Key Points:**

Detailed breakdown of system processes

Shows internal data stores: User Database, Notes Database, File Storage Illustrates process interactions and data transformations

Demonstrates the approval workflow and authentication processes

# Event Table



The Event Table documents all significant events that occur within the Online Notes Sharing System, including user actions, system responses, and administrative activities.

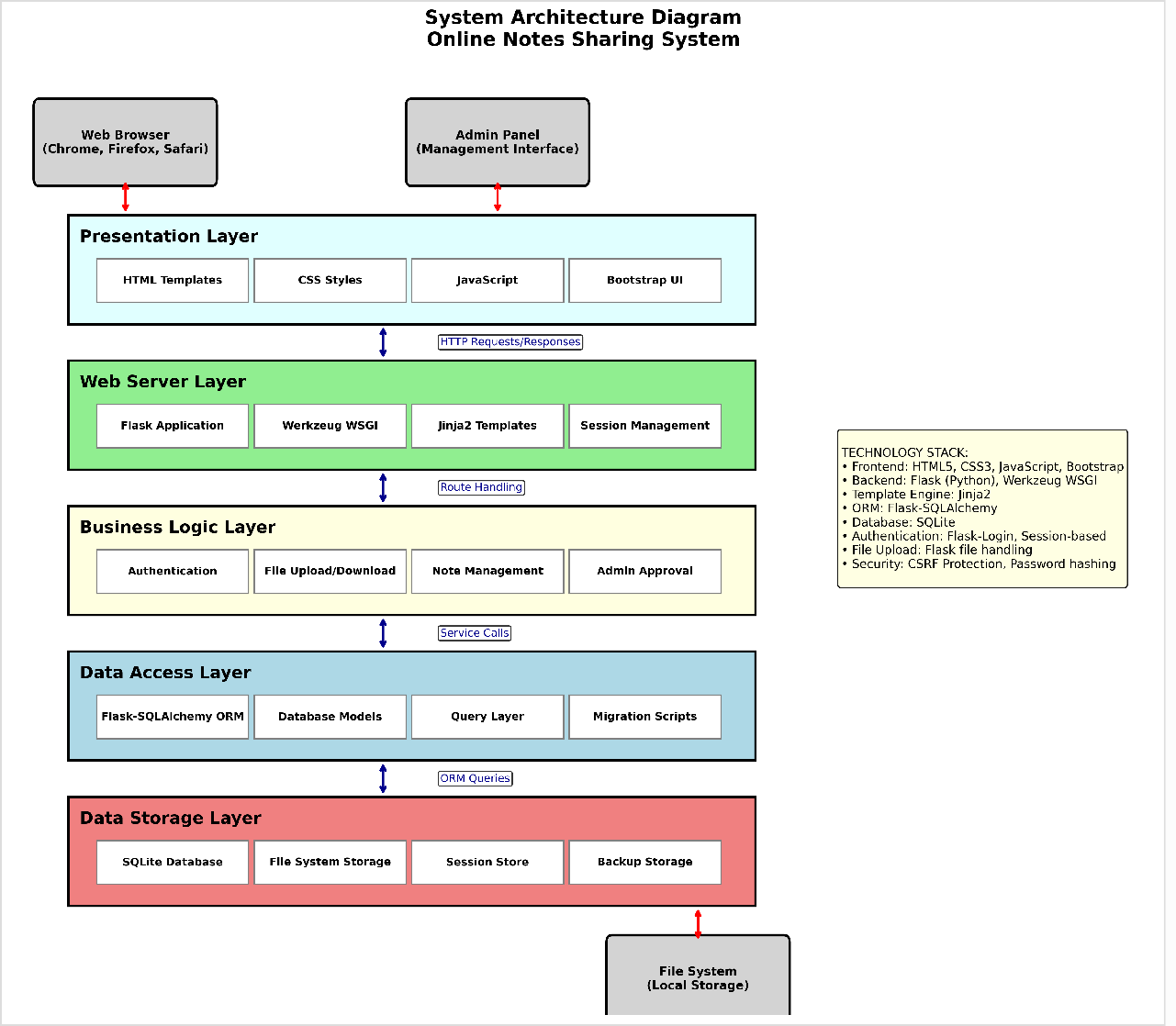
**Key Points:**

Comprehensive list of system events and triggers

Defines event sources, descriptions, and system responses Maps user actions to system processes

Provides foundation for system behavior specification

# System Architecture



The System Architecture diagram illustrates the overall structure of the Online Notes Sharing System, showing the relationships between different layers and components of the application.

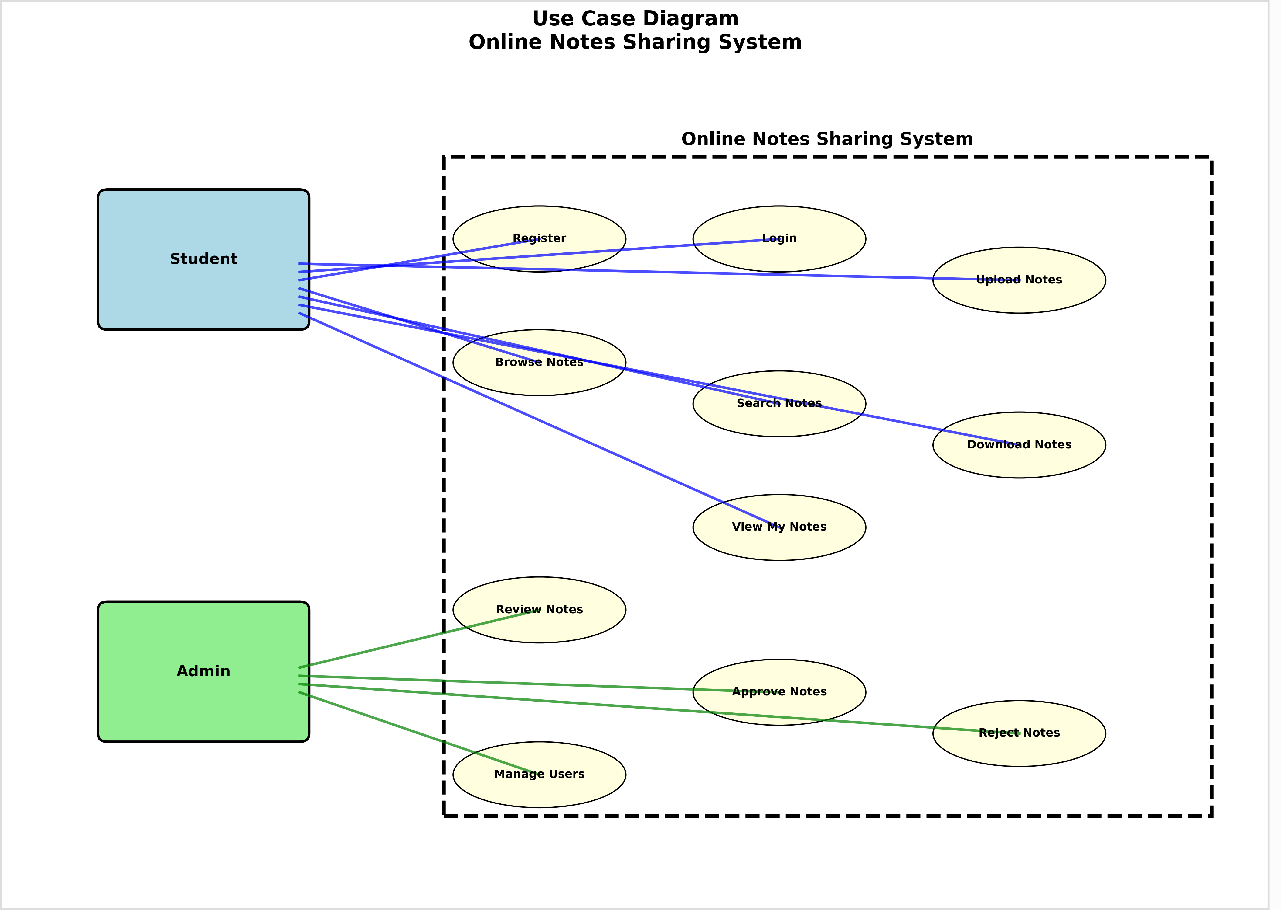
**Key Points:**

Three-tier architecture: Presentation, Application, and Data layers Frontend: HTML, CSS, JavaScript for user interface

Backend: Flask framework with Flask-SQLAlchemy and Flask-CORS Database: SQLite for data persistence

Clear separation of concerns and modular design

# Use Case Diagrams



The Use Case Diagram identifies the actors in the system and their interactions with various functionalities. It provides a clear view of what different types of users can accomplish within the system.

**Key Points:**

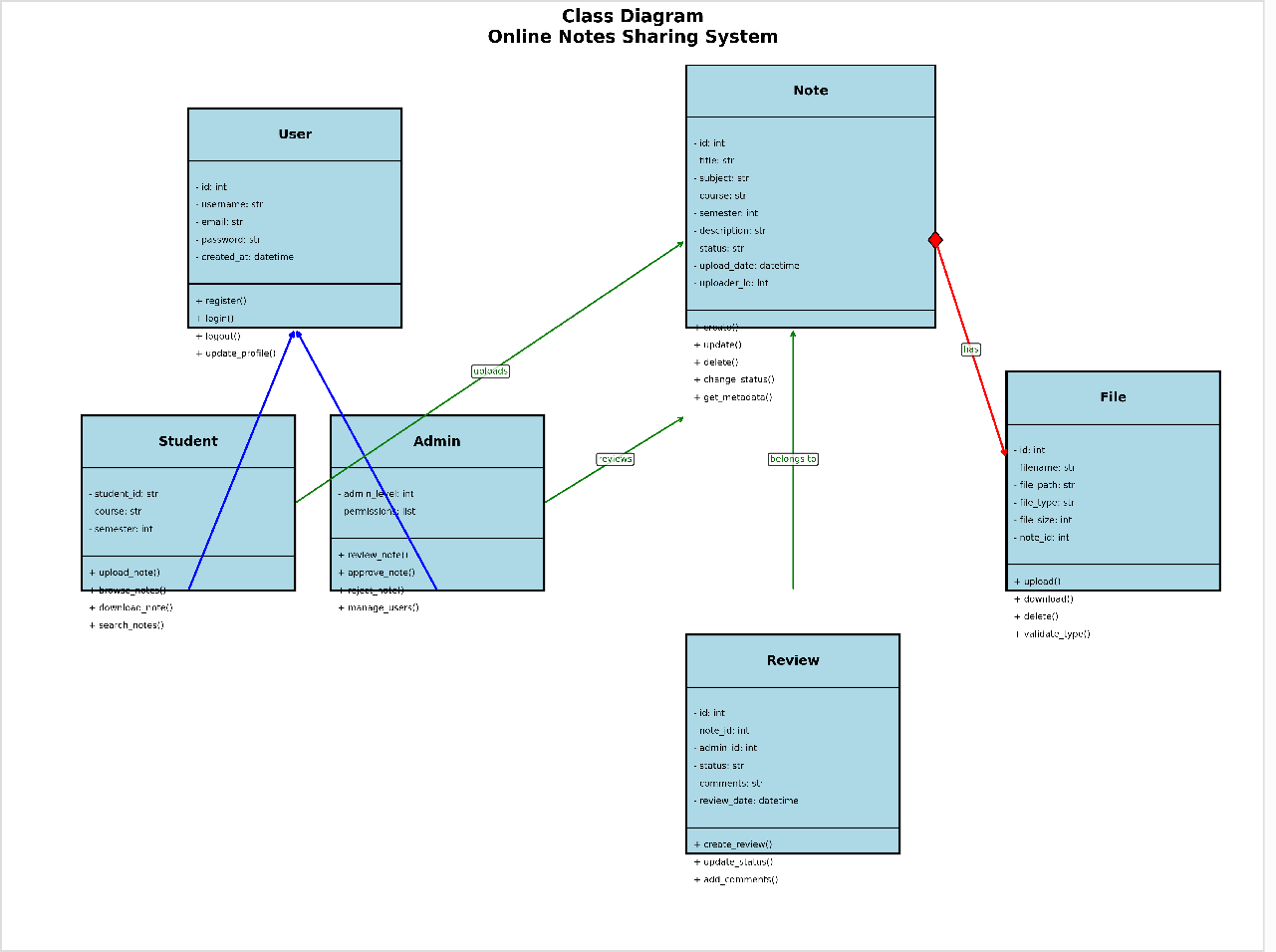
Two primary actors: Student and Admin

Student use cases: Register, Login, Upload Notes, Browse Notes, Download Notes

Admin use cases: Login, Review Notes, Approve/Reject Notes, Manage Users Shows system boundary and external relationships

Illustrates role-based access control

# Class Diagram



The Class Diagram represents the object-oriented design of the system, showing classes, their attributes, methods, and relationships. This diagram serves as the blueprint for the system's implementation.

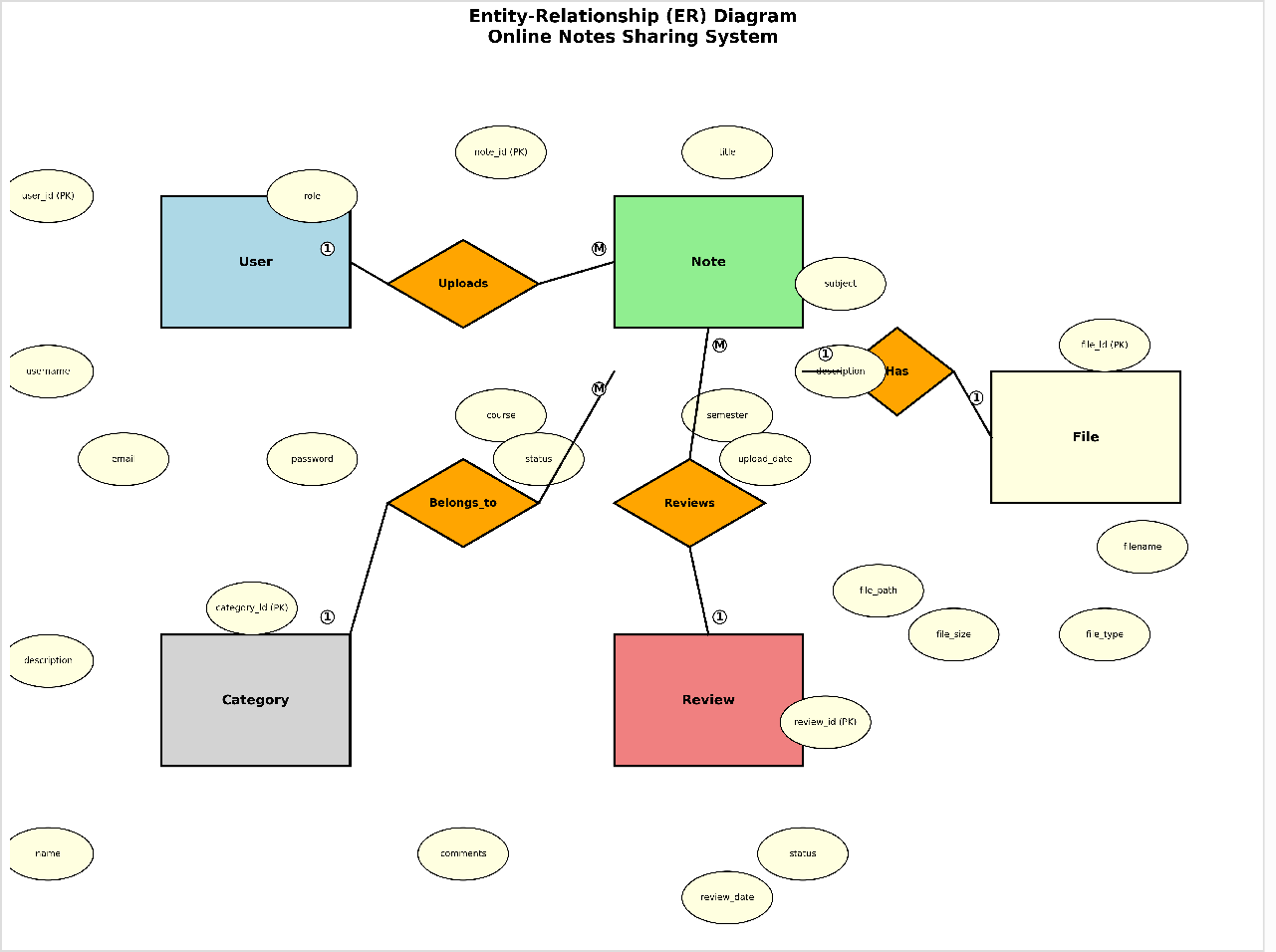
**Key Points:**

Core classes: User, Student, Admin, Note, File

Inheritance relationship between User and Student/Admin classes Association relationships showing data connections

Detailed attributes and methods for each class Encapsulation principles with private and public members

# Entity-Relationship (E-R) Diagram



The E-R Diagram models the database structure by showing entities, their attributes, and the relationships between them. This diagram is crucial for understanding data organization and integrity constraints.

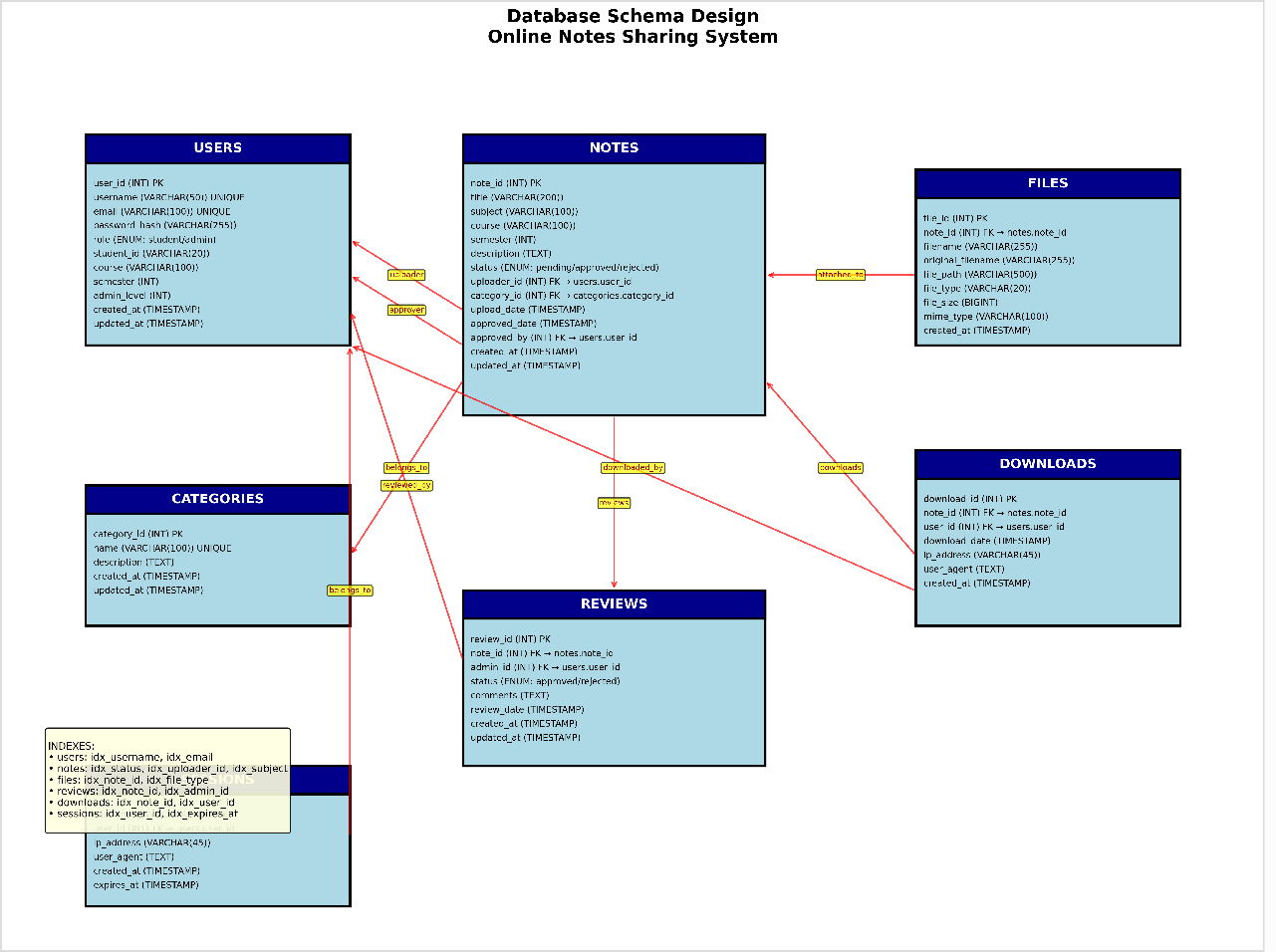
**Key Points:**

Primary entities: Users, Notes, Files

One-to-many relationship between Users and Notes One-to-one relationship between Notes and Files

Attributes include primary keys, foreign keys, and descriptive data Supports referential integrity and data consistency

# Schema Design



The Schema Design provides the detailed database structure with specific table definitions, column specifications, data types, constraints, and relationships required for implementation.

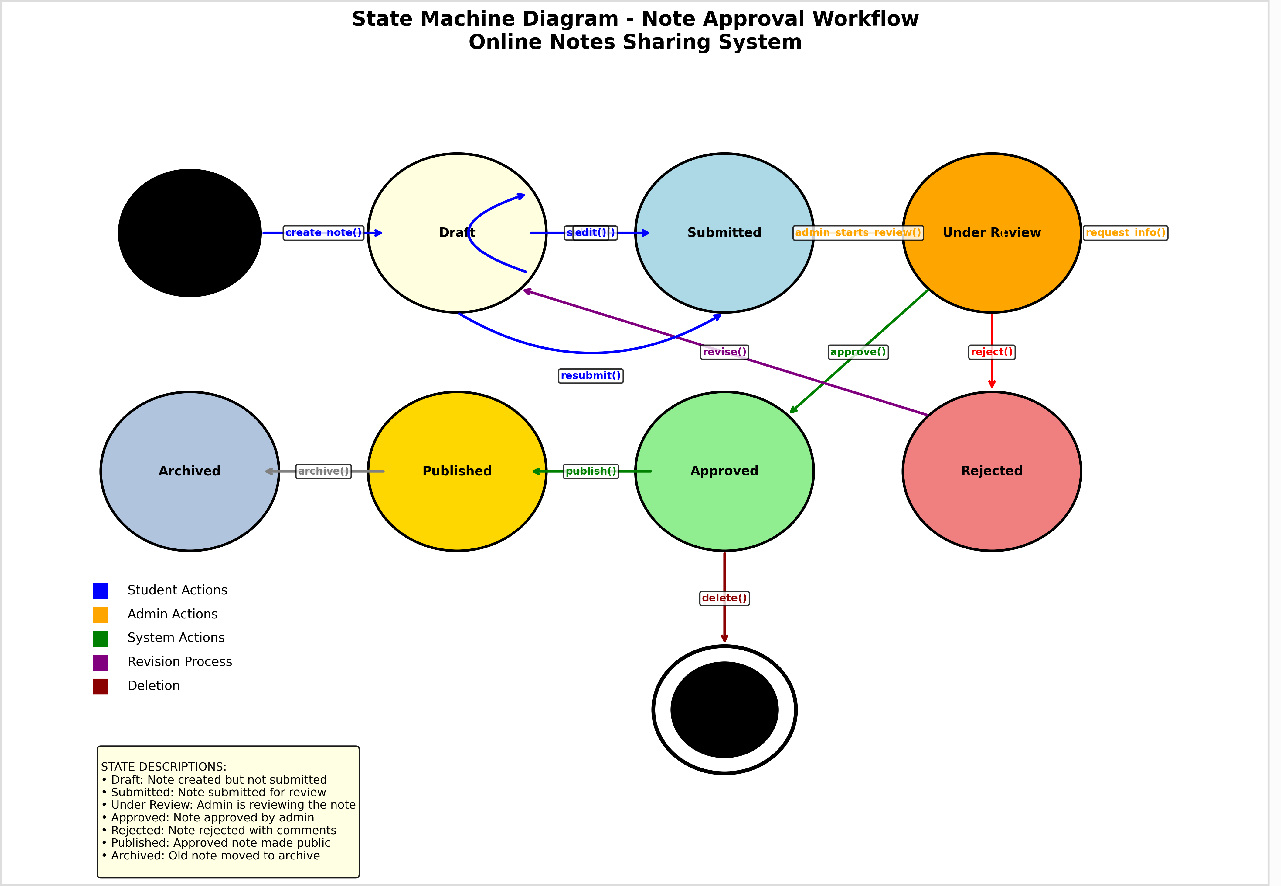
**Key Points:**

Detailed table structures with column names and data types Primary key and foreign key relationships

Constraints for data validation and integrity

Indexes for optimized query performance Normalization to reduce data redundancy

# State Machine Diagram



The State Machine Diagram illustrates the different states that notes can exist in throughout their lifecycle, from initial upload through the approval process to final publication or rejection.

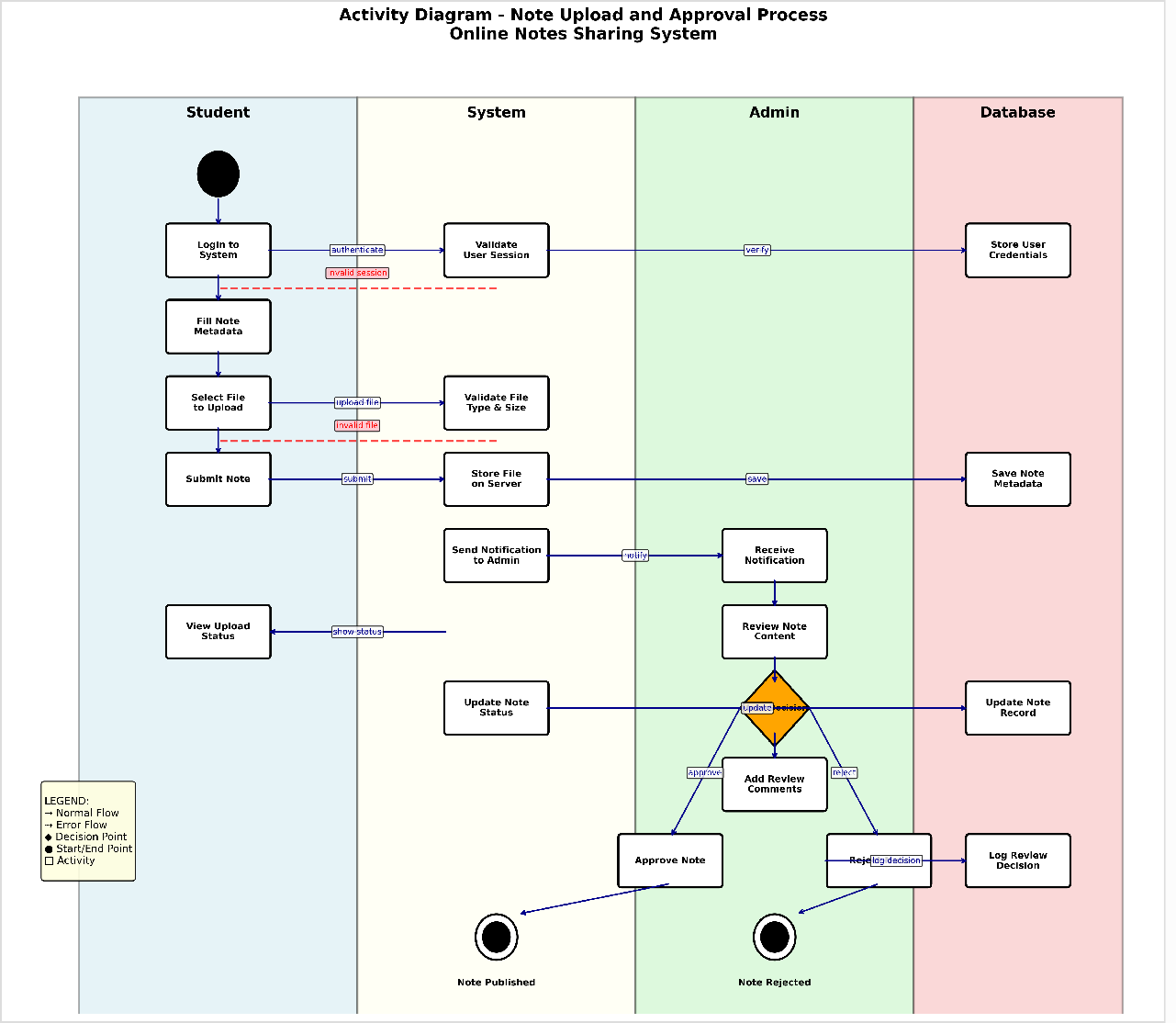
**Key Points:**

States: Pending, Under Review, Approved, Rejected Transitions triggered by admin actions

Initial state upon note upload

Final states for approved and rejected notes Guards and conditions for state transitions

# Activity Diagram



The Activity Diagram shows the workflow and business processes within the system, illustrating the sequence of activities from note upload through administrative review to final disposition.

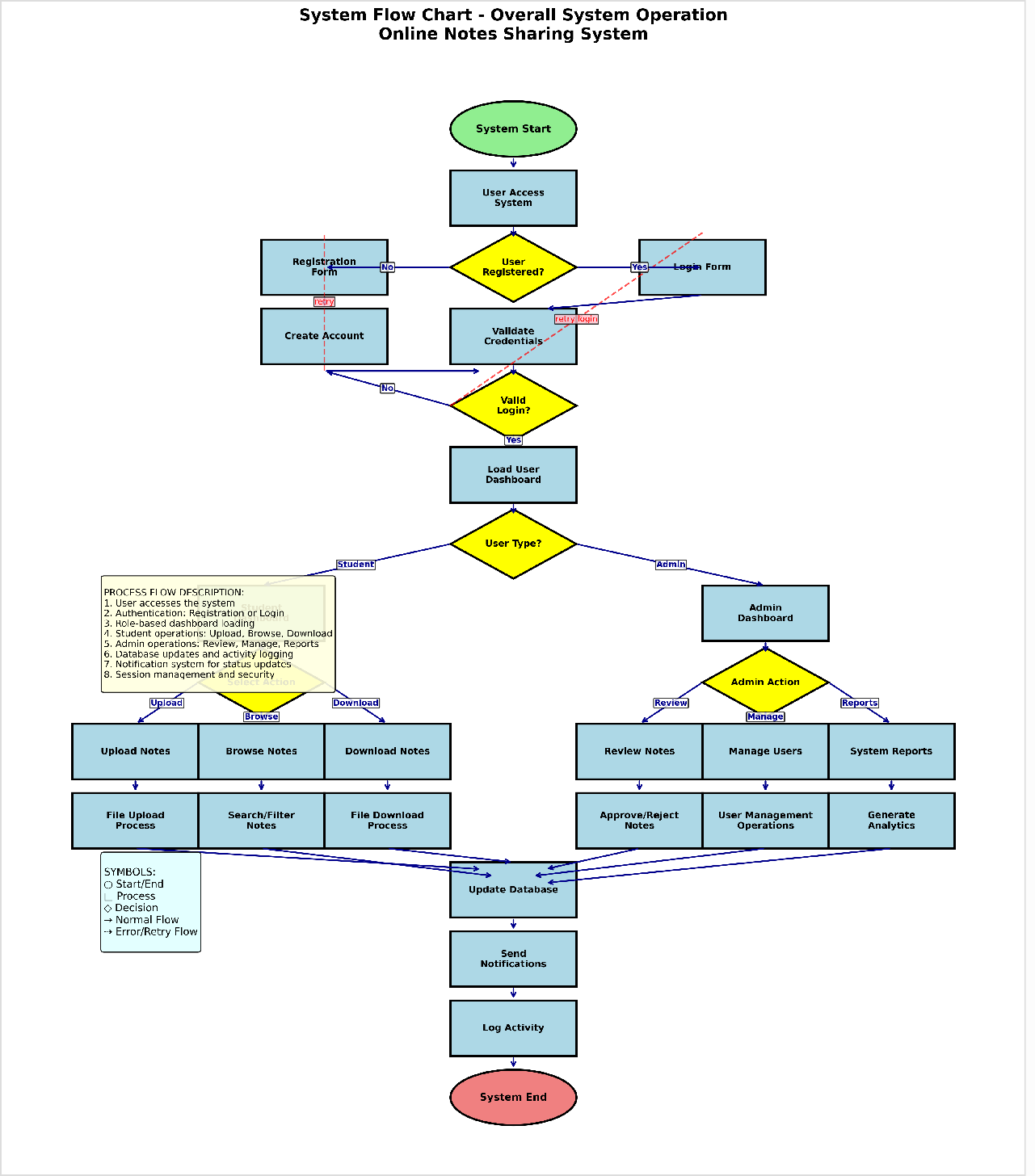
**Key Points:**

Complete workflow from note upload to publication Decision points and parallel activities

Swim lanes showing responsibilities

Exception handling and alternative flows Synchronization and merge points

# System Flow Chart



**Key Points:**

Overall system operation flow

User authentication and authorization processes Main functional workflows

Error handling and validation steps

The System Flow Chart provides a high-level overview of the entire system operation, showing the main processes, decision points, and data flow from user interaction to system response.

System entry and exit points

# Conclusion

This comprehensive documentation provides a complete view of the Online Notes Sharing System through various software engineering diagrams. Each diagram serves a specific purpose in understanding different aspects of the system - from high-level architecture to detailed database design. Together, these diagrams form a solid foundation for system development, maintenance, and future enhancements.

The systematic approach demonstrated through these diagrams ensures that all stakeholders have a clear understanding of the system's functionality, structure, and behavior, facilitating effective communication and successful project implementation.