FYPIFY

Course: SE323L- Software Construction and Development
Project Type: Open Ended Lab (OEL)

Submitted By:

Ahmed Javaid	2023-SE-09
Hafiz Abdullah	2023-SE-22
Ahmed Raza	2023-SE-24
Muhammad Ahad	2023-SE-31

Submitted To:

Sir Ali Raza

Dated: 29th Dec , 2025

Department of Computer Science

University of Engineering and Technology Lahore,

New Campus

Contents

FYPIFY

1 Introduction

1.1 Project Summary

FYPify is a comprehensive, web-based **Final Year Project (FYP) Management System** designed to modernize and automate the academic project life-cycle in universities. It addresses critical institutional challenges such as manual tracking, fragmented communication, and inconsistent evaluation processes by providing a centralized, transparent platform for all stakeholders.

The platform utilizes a robust and modern technology stack:

- **Backend:** Java 17, Spring Boot 3.x, Spring Security (JWT), and PostgreSQL.
- **Frontend:** Next.js 14 (React), TypeScript, and TailwindCSS.
- **Infrastructure:** Cloudinary for document storage, WebSockets for real-time updates, and Docker for containerization.

1.2 Core Functionality

FYPify implements **Role-Based Access Control (RBAC)** to provide tailored experiences for four primary users:

1. **Students:** Manage group formation, document submissions, and result tracking.
2. **Supervisors:** Oversee project progress, approve proposals, and provide feedback.
3. **Committee Members:** Evaluate projects using standardized digital rubrics.
4. **Administrators:** Manage users, configure committees, and generate audit reports.

By following a **Clean Layered Architecture**, FYPify ensures scalability and security, effectively streamlining the entire workflow from initial proposal submission to final graduation requirements.

2 Tools and Technologies

Technology	Version	Purpose
Java	21 (LTS)	Primary programming language
Spring Boot	3.4.0	Application framework for building production-ready applications
Maven	3.9+	Build automation and dependency management

Spring Boot Modules

Module	Purpose

Spring Web	RESTful API development, MVC architecture
Spring Data JPA	Object-Relational Mapping (ORM) with Hibernate
Spring Security	Authentication, authorization, and security features
Spring Validation	Request validation using Jakarta Bean Validation
Spring Mail	Email sending functionality
Spring WebSocket	Real-time bidirectional communication for notifications
Spring Actuator	Application monitoring and health checks

Security & Authentication

Technology	Version	Purpose
JWT (JSON Web Tokens)	0.12.6	Stateless authentication tokens
JJWT Library	0.12.6	JWT creation, parsing, and validation
BCrypt	Built-in	Password hashing algorithm

Database & Migrations

Technology	Version	Purpose
PostgreSQL	15+	Primary relational database
Flyway	Latest	Database schema migrations and version control
Hibernate	6.x	JPA implementation for ORM

Frontend Technologies

Technology	Version	Purpose
React	19.2.1	UI component library
Next.js	16.0.7	React framework with App Router, SSR, and file-based routing

TypeScript	5.x	Type-safe JavaScript superset

UI Component Library

Library	Version	Purpose
shadcn/ui	Latest	Pre-built accessible React components
Radix UI Primitives	Various	Unstyled, accessible UI component primitives
Lucide React	0.556.0	Icon library

Forms & Validation

Library	Version	Purpose
React Hook Form	7.68.0	Performant form state management
Zod	4.1.13	TypeScript-first schema validation
@hookform/resolvers	5.2.2	Zod integration with React Hook Form

File Handling

Library	Version	Purpose
React Hook Form	7.68.0	Performant form state management
Zod	4.1.13	TypeScript-first schema validation
@hookform/resolvers	5.2.2	Zod integration with React Hook Form

Cloud Services

Feature	Purpose
File Storage	Document uploads (PDFs, images)

CDN	Fast content delivery
Transformations	Image optimization and resizing

Email Service

Service	Purpose
SMTP Server	Email delivery (configurable - Gmail, SendGrid, etc.)
Thymeleaf Templates	HTML email templates

Development Tools

Tool	Purpose
Visual Studio Code	Primary code editor
Thymeleaf Templates	HTML email templates
IntelliJ IDEA	Java/Spring Boot development
Tool	**Purpose**
Maven	Java dependency management and builds
npm	Node.js package management
Turbo Pack	Next.js development bundler (faster than Webpack)
Visual Studio Code	Primary code editor
IntelliJ IDEA	Java/Spring Boot development
Maven	Java dependency management and builds
npm	Node.js package management
Turbopack	Next.js development bundler (faster than Webpack)
Git	Source code version control
GitHub	Repository hosting and collaboration

3 Design Patterns Used

This document explains the **Gang of Four (GOF) Design Patterns** implemented in the FYPify backend with references to specific files.

Pattern	Type
Singleton	Creational
Builder	Creational
Factory Method	Creational
Template Method	Behavioral
Chain of Responsibility	Behavioral
Strategy	Behavioral
Adapter	Structural
Repository	Architectural
DTO (Data Transfer Object)	Architectural

3.1 Singleton Pattern (Creational)

Purpose: Ensure a class has only one instance and provide global access to it.
Where Used: Spring's IoC container manages all beans as singletons by default.

File	Path
SecurityConfig	SecurityConfig.java
JwtAuthenticationFilter	JwtAuthenticationFilter.java
GlobalExceptionHandler	GlobalExceptionHandler.java
UserService	UserService.java

3.2 Builder Pattern (Creational)

Purpose: Construct complex objects step by step, separating construction from representation.
Where Used: Entity creation, DTO construction, API responses.

File	Path
ApiResponse	ApiResponse.java
ApiError	ApiError.java
SecurityConfig	SecurityConfig.java
UserService	UserService.java

3.3 Factory Method Pattern (Creational)

Purpose: Define an interface for creating objects, letting sub-classes decide which class to instantiate.
Where Used: Static factory methods for creating API responses and exceptions.

File	Path
ApiResponse	ApiResponse.java
BaseException	BaseException.java

3.4 Template Method Pattern (Behavioral)

Purpose: Define the skeleton of an algorithm, deferring specific steps to sub-classes.
Where Used: Security filters extending OncePerRequestFilter, exception hierarchy.

File	Path
JwtAuthenticationFilter	JwtAuthenticationFilter.java
BaseException	BaseException.java
GlobalExceptionHandler	GlobalExceptionHandler.java

3.5 Chain of Responsibility Pattern (Behavioral)

Purpose: Pass requests along a chain of handlers, where each handler decides to process or pass to the next.
Where Used: Spring Security filter chain, exception handler chain.

File	Path
SecurityConfig	SecurityConfig.java
JwtAuthenticationFilter	JwtAuthenticationFilter.java
GlobalExceptionHandler	GlobalExceptionHandler.java

3.6 Strategy Pattern (Behavioral)

Purpose: Define a family of algorithms, encapsulate each one, and make them interchangeable.
Where Used: Authentication providers, password encoding.

File	Path
SecurityConfig	SecurityConfig.java
JwtTokenProvider	JwtTokenProvider.java

3.7 Adapter Pattern (Structural)

Purpose: Convert the interface of a class into another interface clients expect.
Where Used: Converting exceptions to standardized API responses.

File	Path
GlobalExceptionHandler	GlobalExceptionHandler.java

3.8 Repository Pattern (Architectural)

Purpose: Mediate between domain and data mapping layers, acting like an in-memory collection.
Where Used: All data access operations via Spring Data JPA.

File	Path
User	modules/user/repository/UserRepository.java
Group	modules/group/repository/GroupRepository.java
Project	modules/project/repository/ProjectRepository.java
Submission	modules/submission/repository/SubmissionRepository.java

Committee	modules/committee/repository/CommitteeRepository.java

3.9 Pattern Categories Summary (Creational Patterns)

File	Path	Example
Singleton	Single instance	@Service,@Component classes
Builder	Complex object construction	ApiResponse, Entity classes
Factory Method	Object creation encapsulation	ApiResponse.success()

3.10 Behavioral Patterns

Pattern	Purpose	Example File
Chain of Responsibility	Request handling chain	Security Filters
Template Method	Algorithm skeleton	OncePerRequestFilter
Strategy	Interchangeable algorithms	PasswordEncoder

4 Functional Requirements

4.1 Functional Requirements Student (Leader + Member)

FR-ID	Functional Requirement	Actor	Priority	Description	Constraints / Rules
STU-01	User Registration & Login	Student	H	Student logs in using credentials	Based on role; JWT authentication
STU-02	View Profile	Student	M	Student can view personal info	Read-only
STU-03	Create Group	Student	H	Student creates a group with member emails	Must respect group_min_size & group_max_size
STU-04	Join Group	Student	H	Student accepts invite from group	Cannot join multiple groups
STU-06	Register Project	Leader	H	Leader submits title, abstract, domain,	Project → status = “PENDING_APPROVAL”

				proposed supervisors	
STU-07	View Group Project Status	Student	H	Student sees current project approval state	Both leader & members can view
STU-08	Upload Document	Leader	H	Leader uploads submission file	Only leader; versioned uploads
STU-09	Re-upload After Revision	Leader	H	Leader uploads new version after supervisor request	Creates new version number
STU-10	View All Submissions	Student	H	Students view documents, comments, statuses	Marks remain hidden until final release
STU-11	View Deadlines	Student	M	Students view deadlines per document	Read-only
STU-12	Receive Notifications	Student	H	Student receives system notifications	WS or polling
STU-13	View Final Results	Student	H	Student views results ONLY after release	Access blocked before release

4.2 Functional Requirements (Supervisor)

FR-ID	Functional Requirement	Actor	Priority	Description	Constraints
SUP-01	View Assigned Projects	Supervisor	H	See list of groups assigned post-approval	Assigned by FYP Committee
SUP-02	View Student Submissions	Supervisor	H	See latest versions submitted	Including file preview
SUP-03	Review Submission	Supervisor	H	Add comments, approve or request revision	Status change stored
SUP-04	Give Supervisor Marks	Supervisor	H	Score document (0–100)	Hidden from students
SUP-05	Request Revision	Supervisor	H	Send revisions to leader with comments	Triggers notification
SUP-06	Approve Submission	Supervisor	H	Approving locks submission (if deadline passed)	Auto-lock
SUP-07	Track Progress	Supervisor	M	View submission completeness per group	Read-only

4.3 Functional Requirements (Evaluation Committee)

FR-ID	Functional Requirement	Actor	Priority	Description	Constraints
EVAL-01	View Locked Submissions	Evaluator	H	View documents ready for evaluation	Only for LOCKED_FOR_EVAL
EVAL-02	View Submission Details	Evaluator	H	Preview file, supervisor comments	Marks from supervisor hidden
EVAL-03	Score Submission	Evaluator	H	Assign committee score per document	Stored per evaluator
EVAL-04	Save Draft Evaluation	Evaluator	M	Save partial evaluation without finalizing	Draft not counted
EVAL-05	Finalize Evaluation	Evaluator	H	Submit final score & comments	Triggers FYP Committee notification
EVAL-06	View Evaluation History	Evaluator	M	See their past evaluations	Only own evaluations

4.4 Functional Requirements (FYP Committee)

FR-ID	Functional Requirement	Actor	Priority	Description	Constraints
FYP-01	View Pending Projects	FYP Member	H	List of submitted projects awaiting approval	Status → "PENDING_APPROVAL"
FYP-02	Approve/Reject Project	FYP Member	H	Approve project and assign supervisor	Required step
FYP-03	Set Deadlines Per Document	FYP Member	H	Choose due dates for required doc types	Used for locking process
FYP-04	Monitor Submissions & Evaluations	FYP Member	M	See progress and pending evaluations	Read-only
FYP-05	Compute Final Results	FYP Member	H	Combine supervisor + evaluator scores using weights	Produces final result object
FYP-06	Release Final Results	FYP Member	H	Publish results to students	Not reversible unless special flow
FYP-07	Approve Result Release (Committee)	FYP Member	H	Multiple member approval (if configured)	Optional feature

4.5 Functional Requirements (Admin)

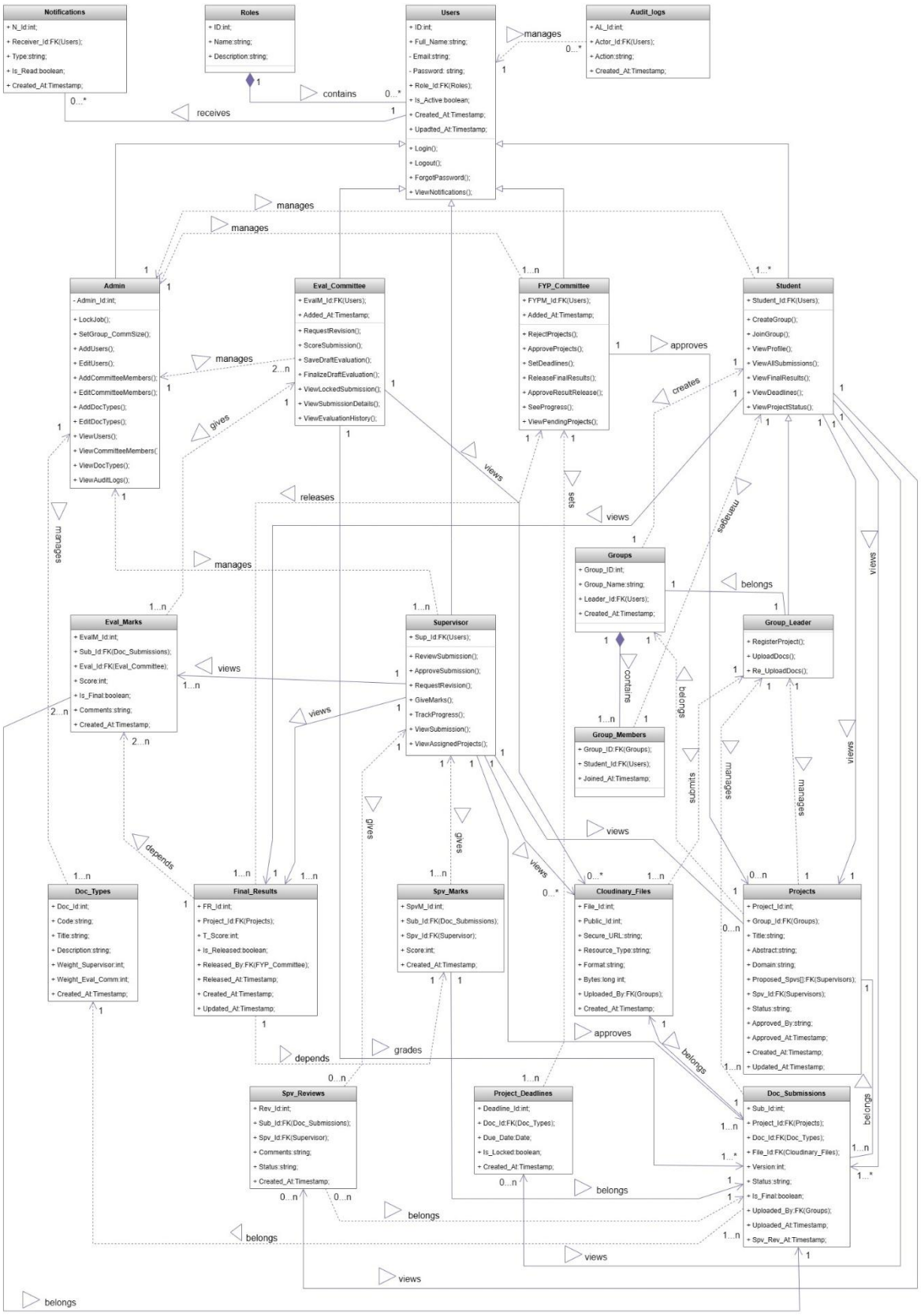
FR-ID	Functional Requirement	Actor	Priority	Description	Constraints
ADM-01	Manage Users	Admin	H	Create, update, delete users; assign roles	Must maintain audit logs
ADM-02	Manage Committees	Admin	H	Add/remove FYP Committee and Evaluation Committee members	Committees must always have at least 1 member
ADM-03	Manage Document Types	Admin	H	Add/edit doc types with weights	Total weights must sum correctly
ADM-04	Manage System Settings	Admin	H	Group size settings, deadline	Stored centrally
ADM-05	View Audit Logs	Admin	M	View all system actions	Read-only
ADM-06	Trigger Backup / Restore	Admin	L	Backup system data	Sensitive action

4.6 System / BackGround Jobs

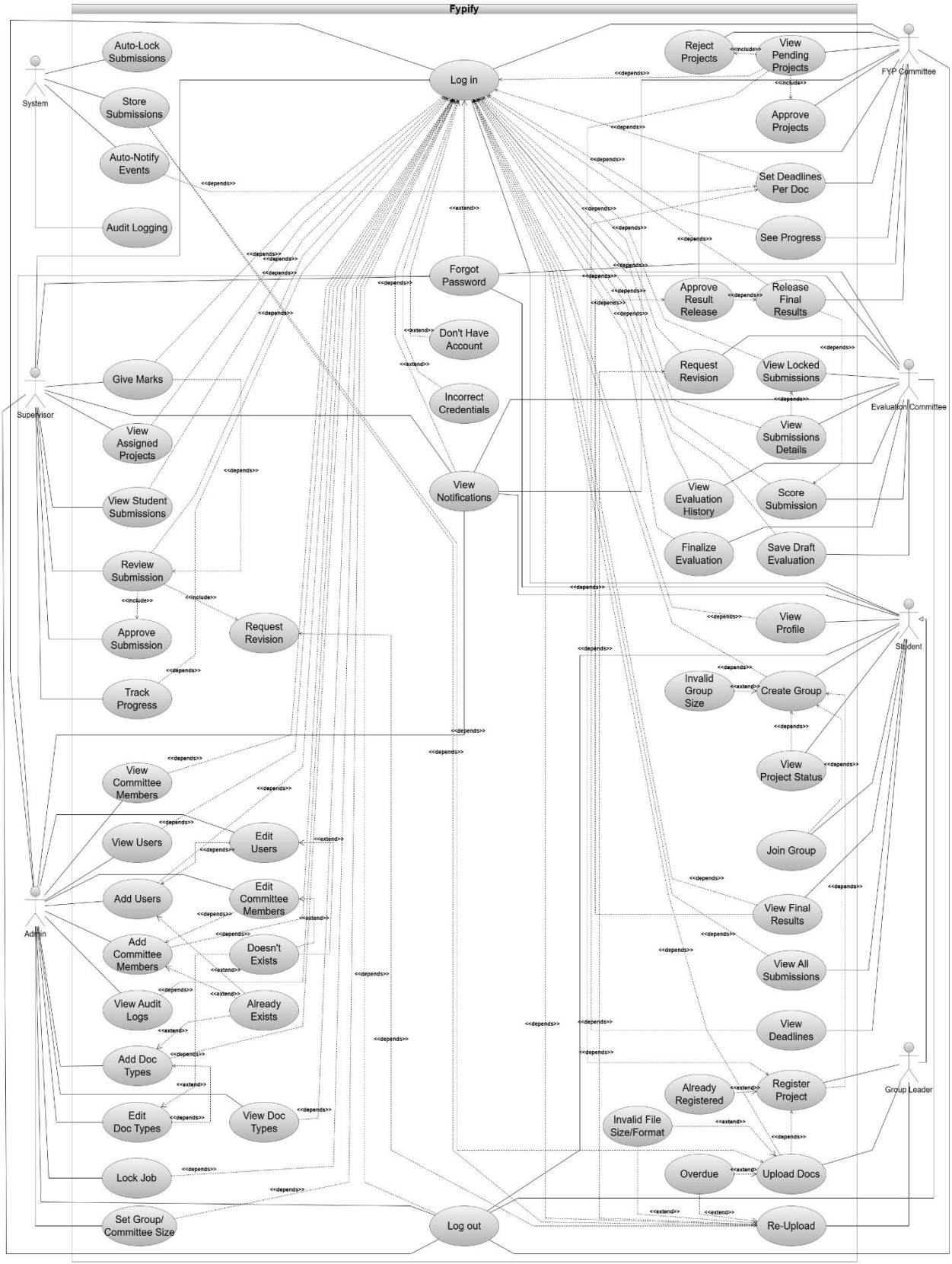
FR-ID	Functional Requirement	Actor	Priority	Description	Constraints
SYS-01	Auto-Lock Submissions	System	H	Lock document after deadline to evaluators	No upload allowed after lock
SYS-02	Auto-Notify Events	System	H	Send notifications for all important events	Works via logs + triggers
SYS-03	Cloud File Storage	System	H	Store submissions in Cloudinary securely	Return secured URL
SYS-04	Audit Logging	System	H	Log all critical actions	Immutable logs
SYS-05	Role-Based Access Control	System	H	Ensure all actions respect permissions	Enforced in backend

5 DIAGRAMS

5.1 Class Diagrams

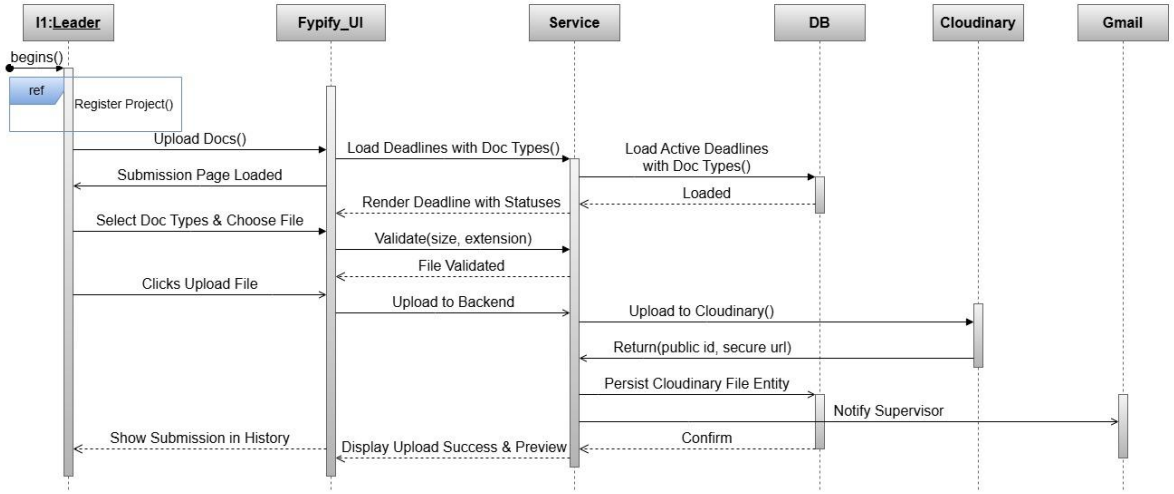


5.2 Use-Case Diagram

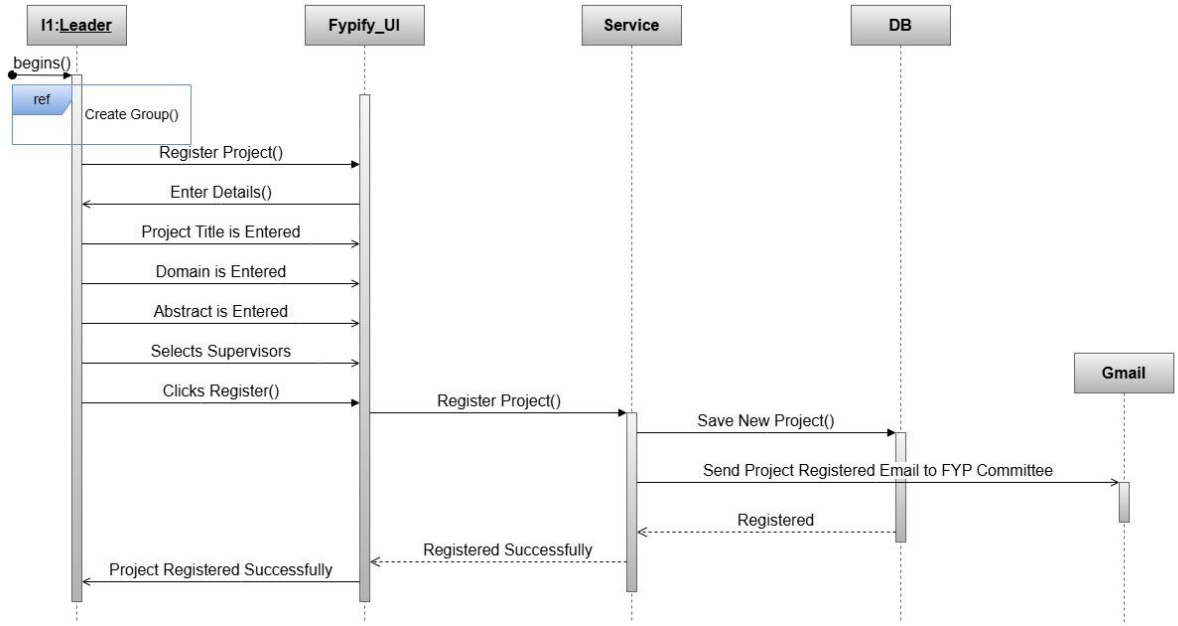


5.3 Sequence Diagrams

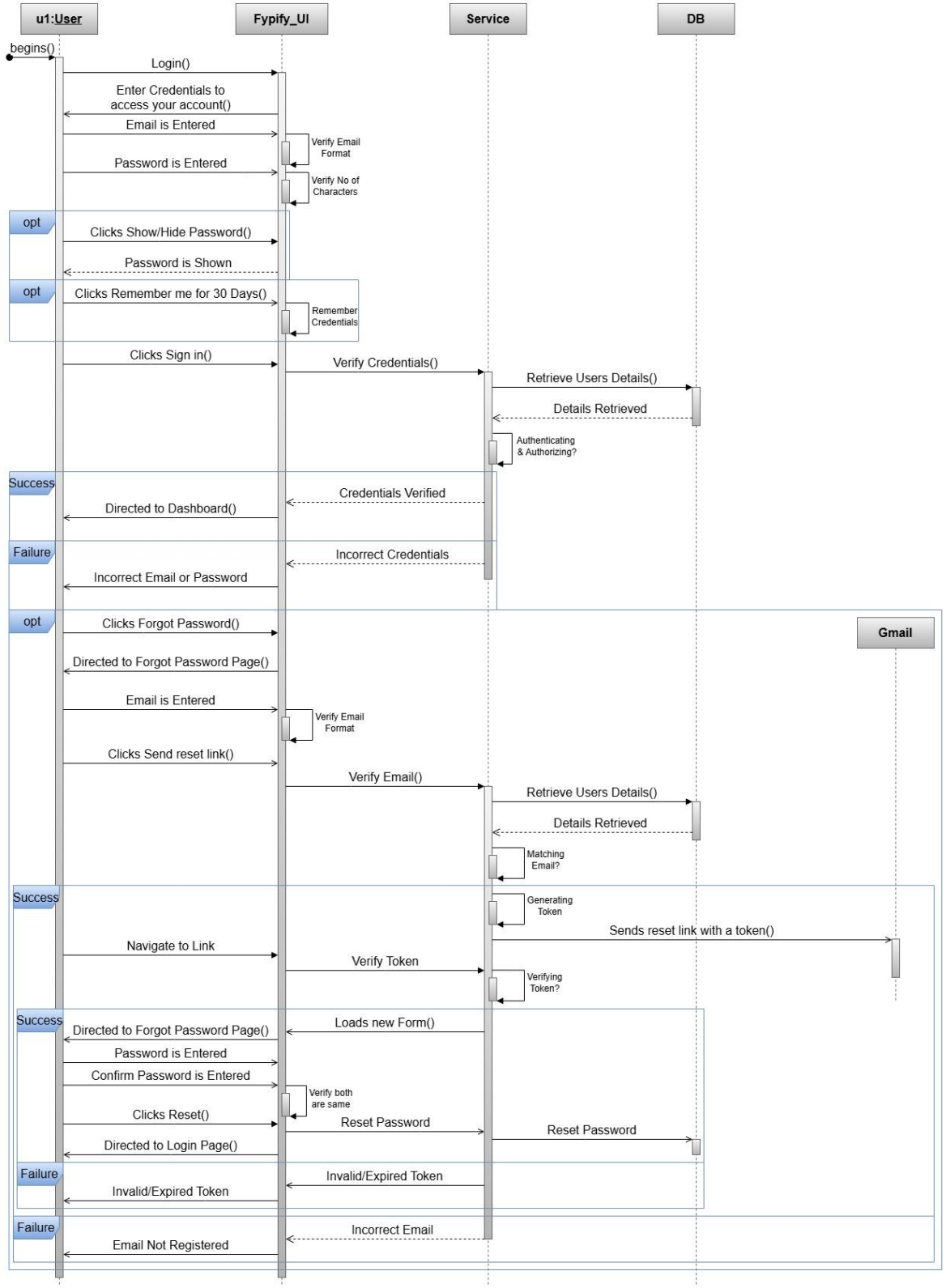
5.3.1 Upload Document



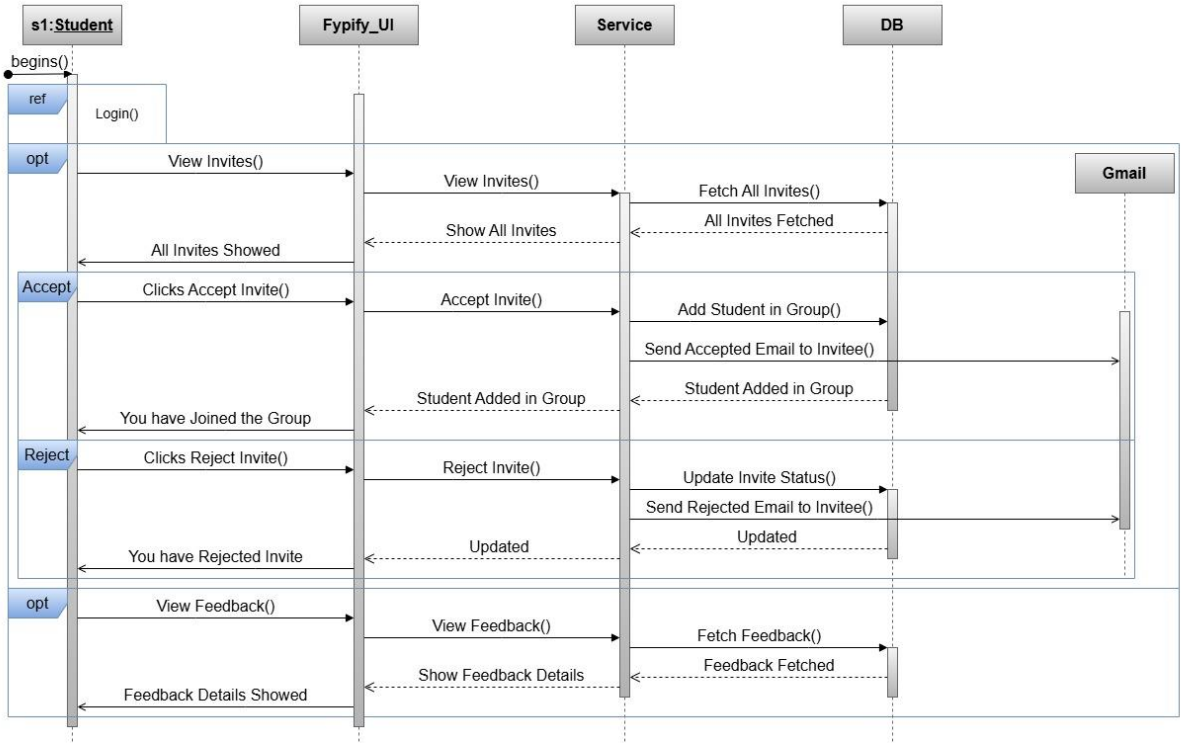
5.3.2 Register Project



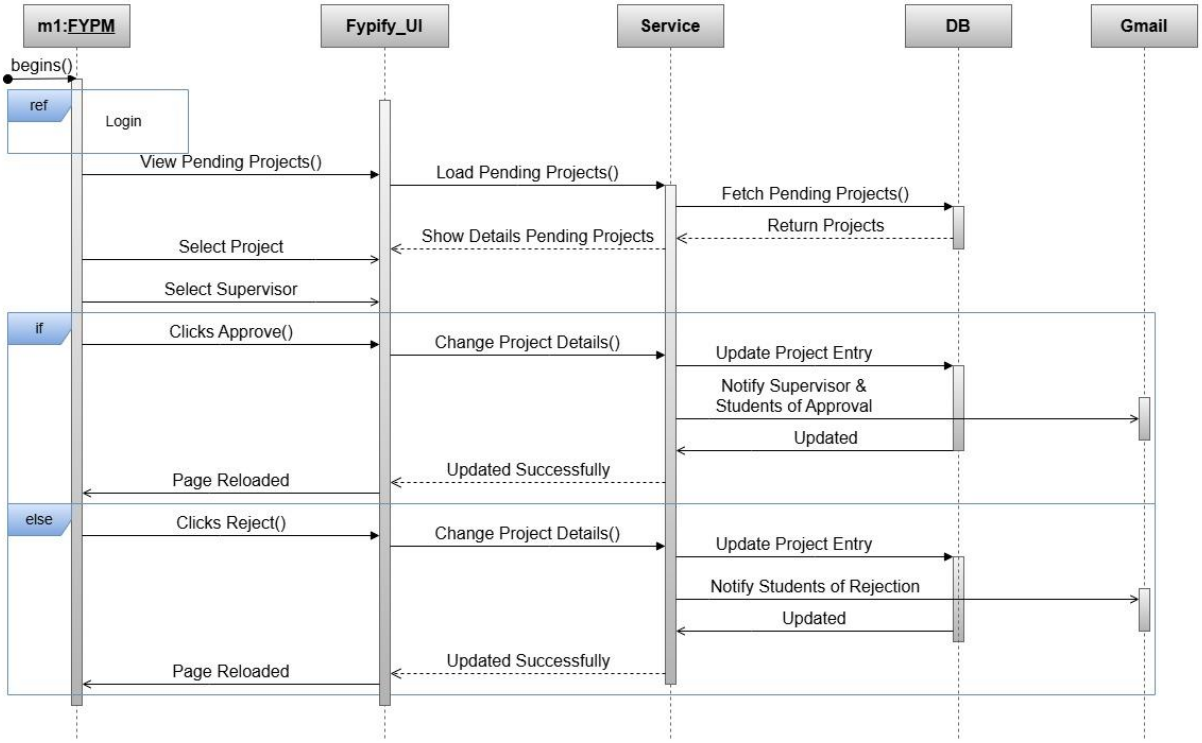
5.3.3 Login


u1:User
Fypify_UI
Service
DB
begins()
Login()
Enter Credentials to access your account()
Email is Entered
Verify Email Format
Password is Entered
Verify No of Characters
opt
Clicks Show/Hide Password()
Password is Shown
opt
Clicks Remember me for 30 Days()
Remember Credentials
Clicks Sign in()
Verify Credentials()
Retrieve Users Details()
Details Retrieved
Authenticating & Authorizing?
Success
Credentials Verified
Directed to Dashboard()
Failure
Incorrect Credentials
Incorrect Email or Password
opt
Clicks Forgot Password()
Gmail
Directed to Forgot Password Page()
Email is Entered
Verify Email Format
Clicks Send reset link()
Verify Email()
Retrieve Users Details()
Details Retrieved
Matching Email?
Success
Generating Token
Sends reset link with a token()
Navigate to Link
Verify Token
Verifying Token?
Success
Loads new Form()
Directed to Forgot Password Page()
Password is Entered
Confirm Password is Entered
Verify both are same
Clicks Reset()
Reset Password
Reset Password
Directed to Login Page()
Failure
Invalid/Expired Token
Invalid/Expired Token
Failure
Incorrect Email
Email Not Registered

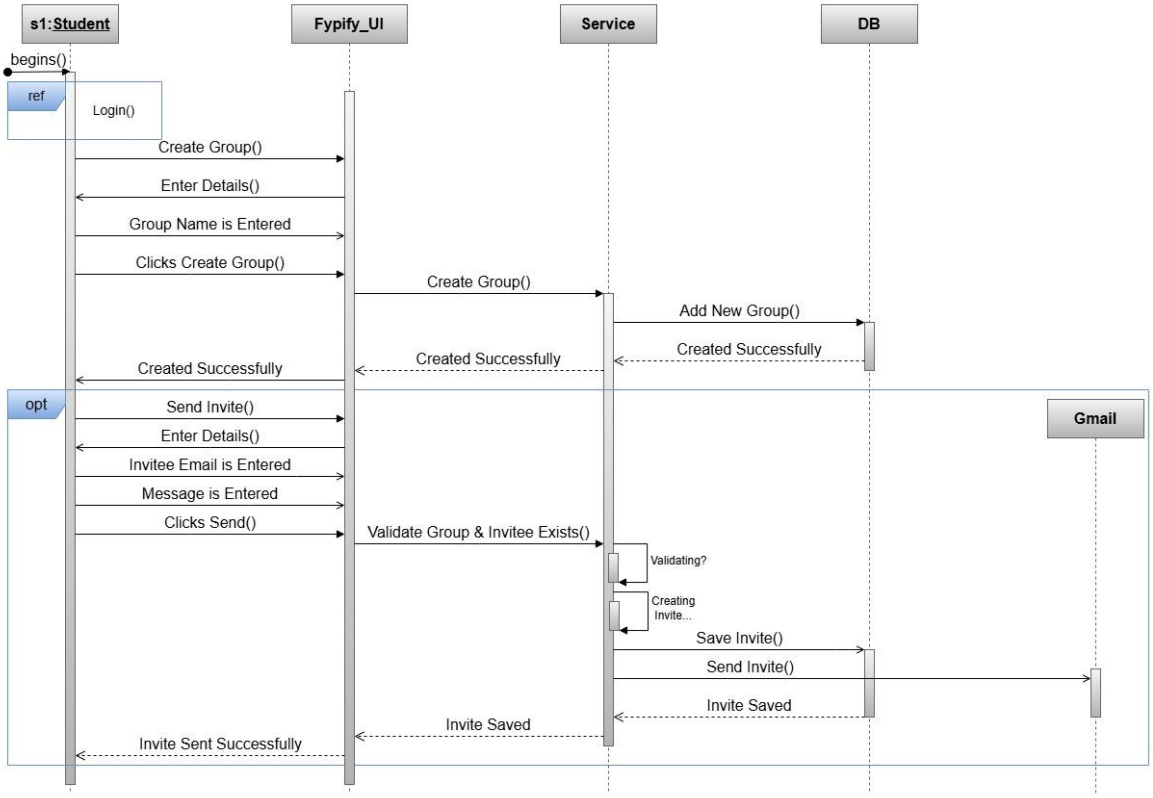
5.3.4 Accept /Reject Invite



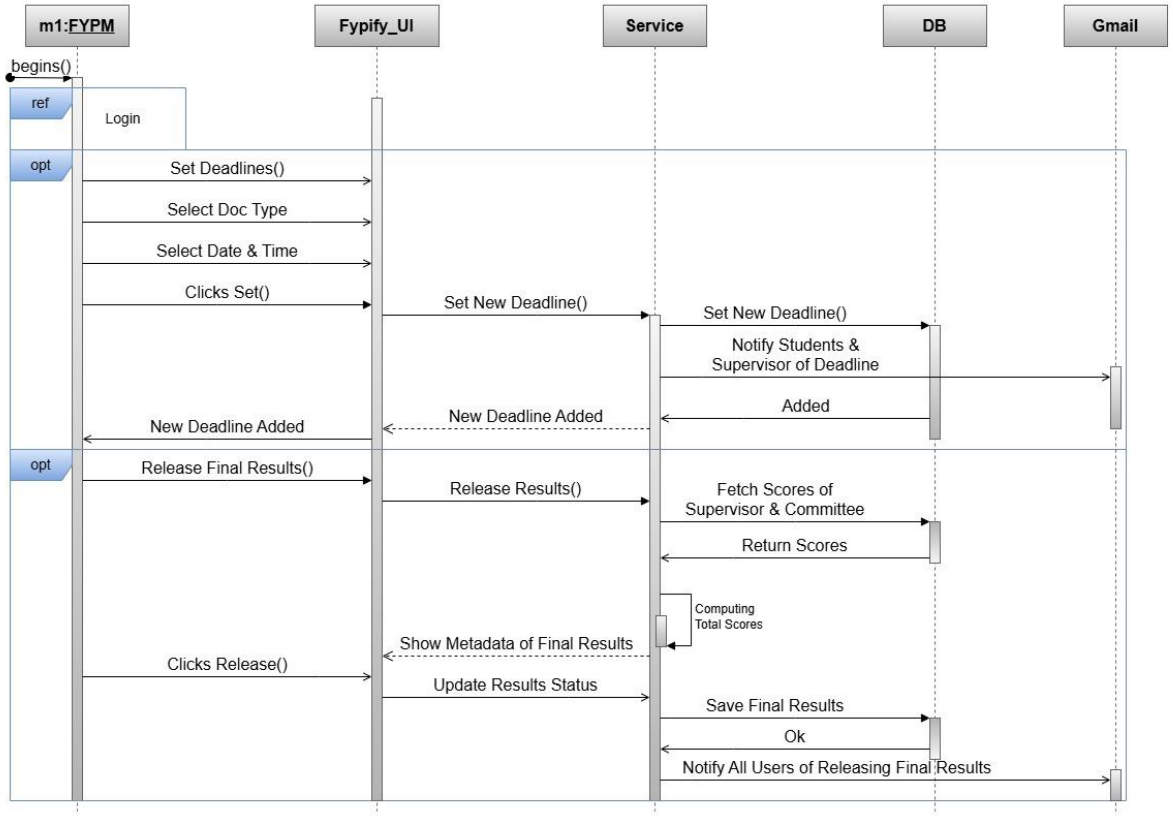
5.3.5 Approve Project



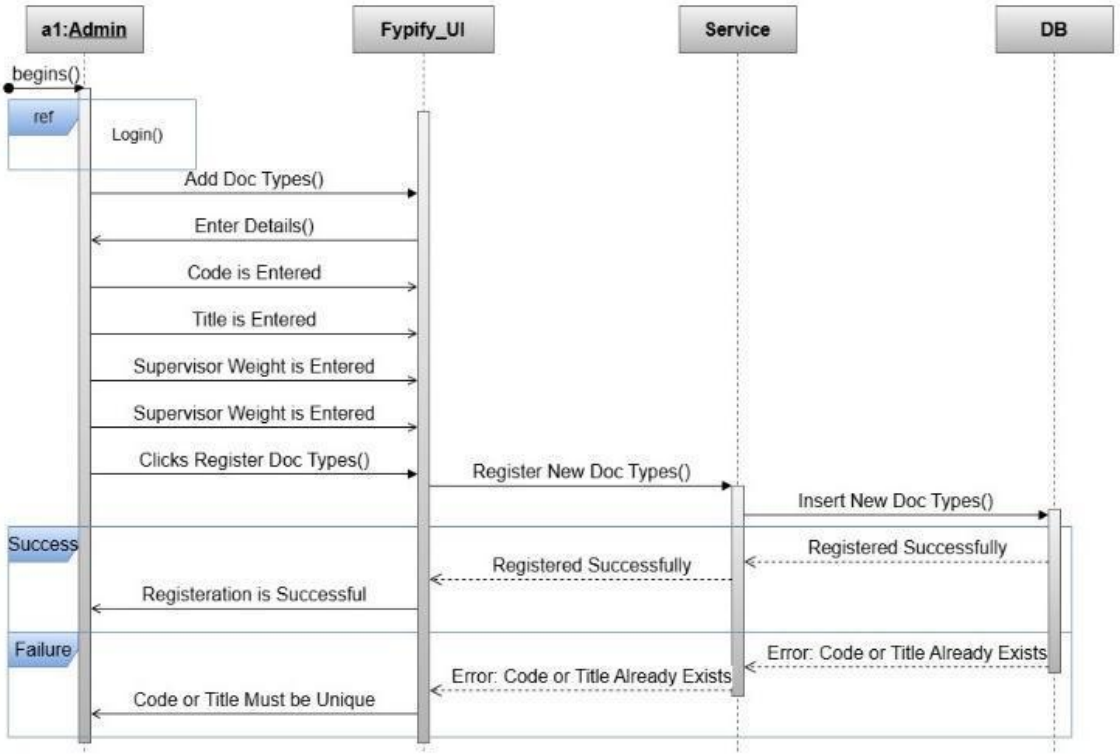
5.3.6 Create Group



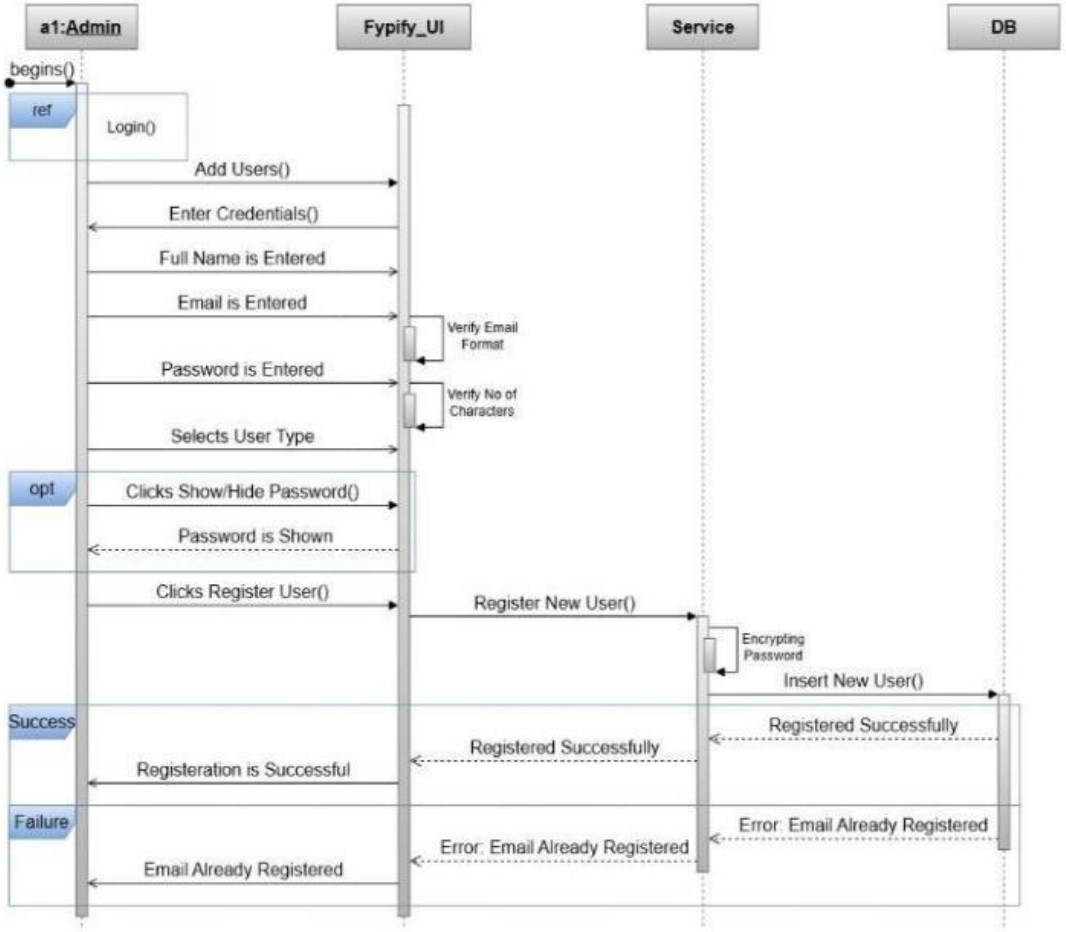
5.3.7 Add Deadline



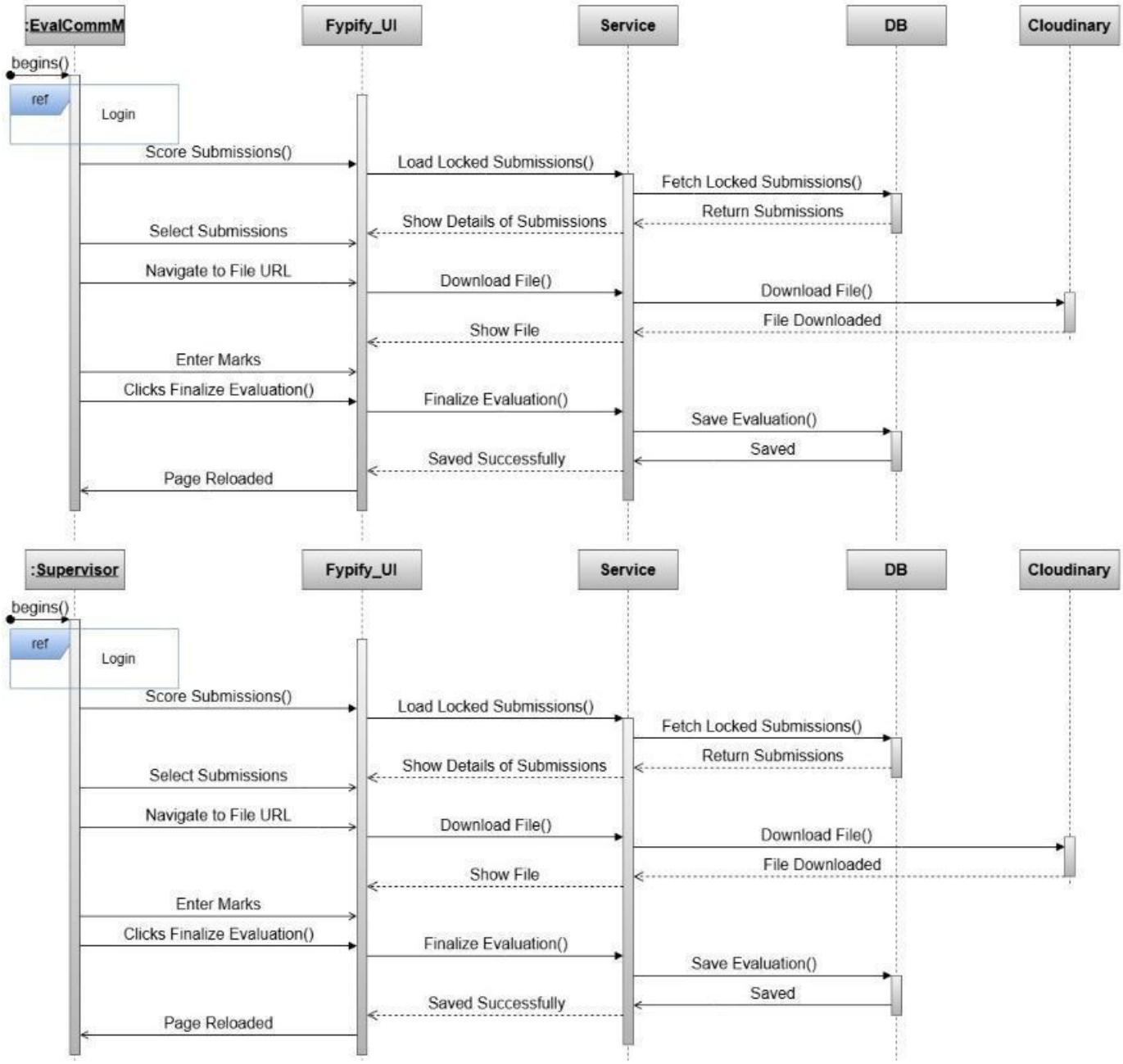
5.3.8 Add Document Type



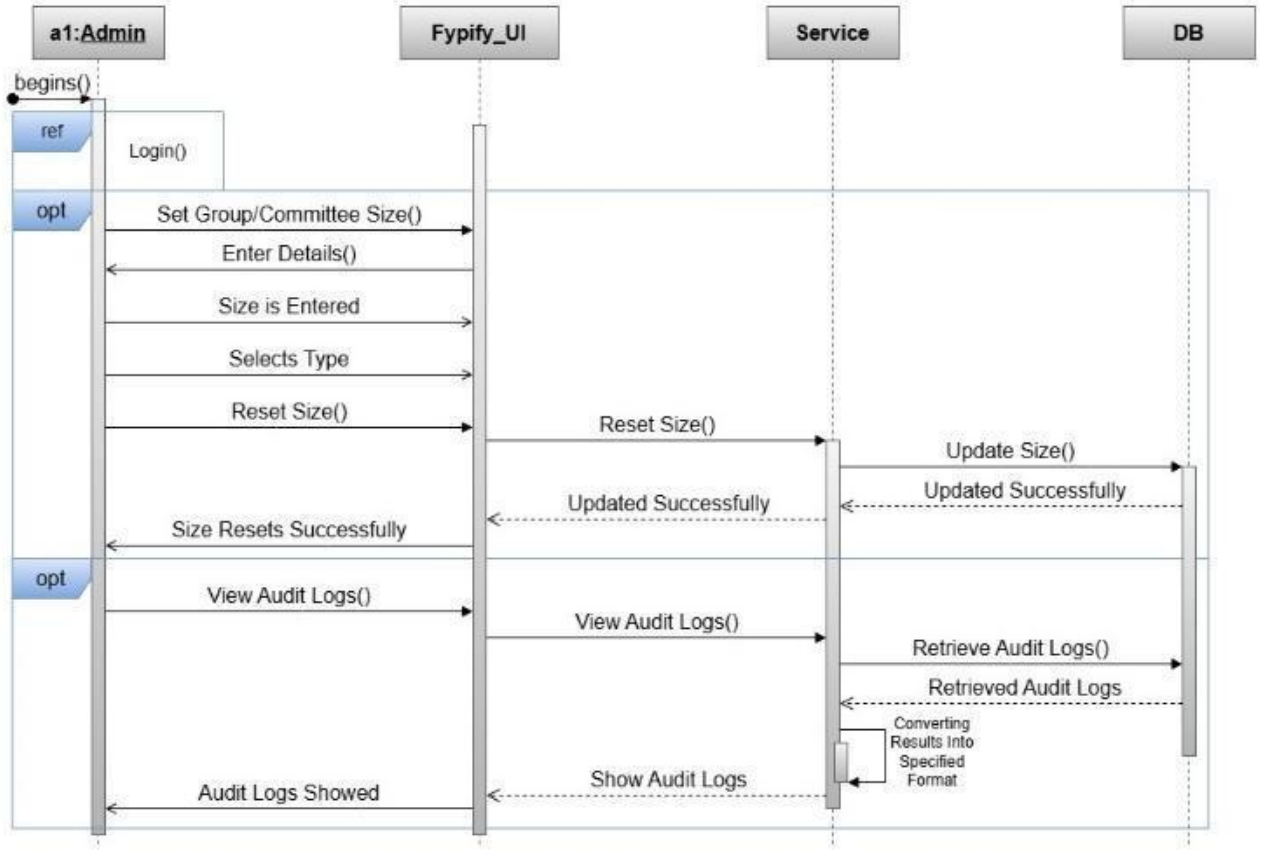
5.3.9 Add User



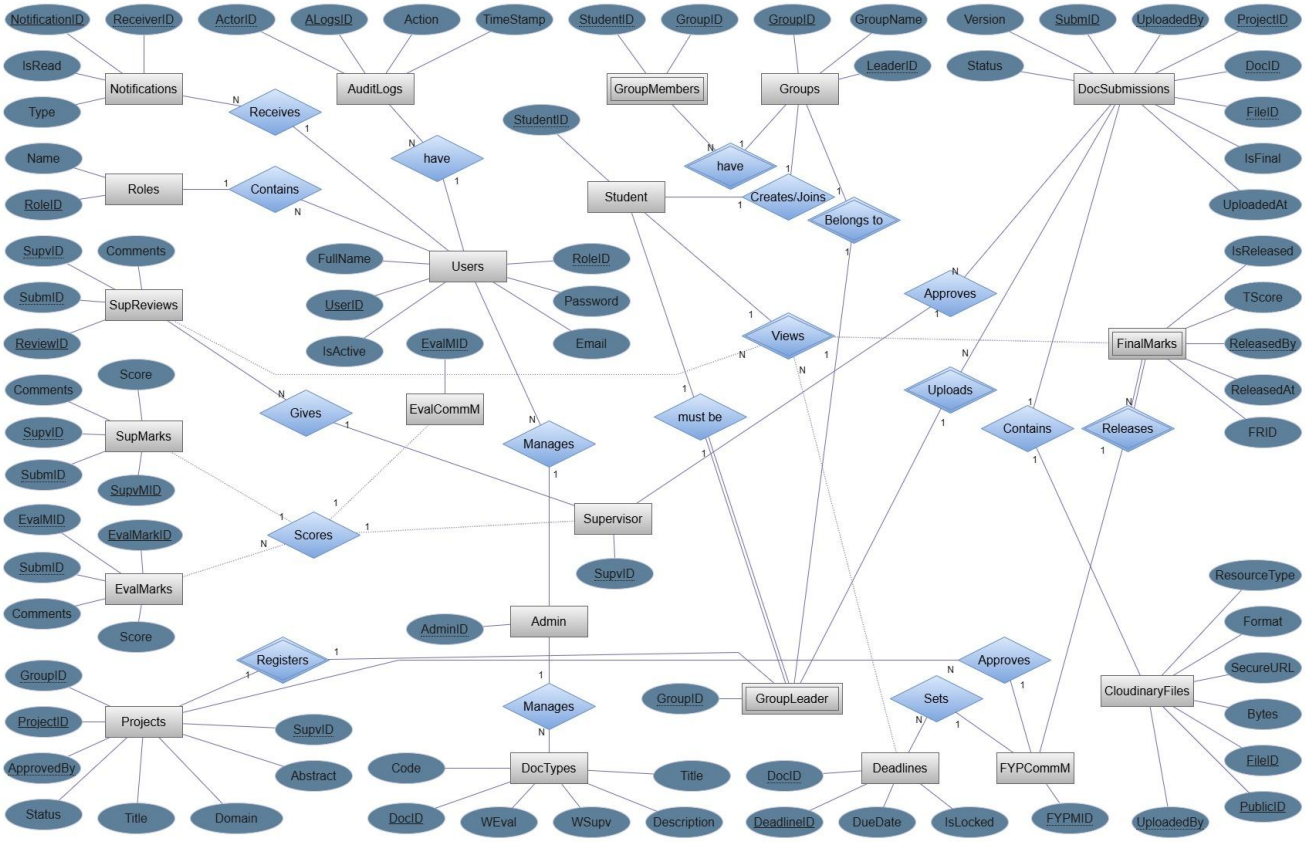
5.3.10 Evaluate Submission



5.3.11 Manage Groupe Size

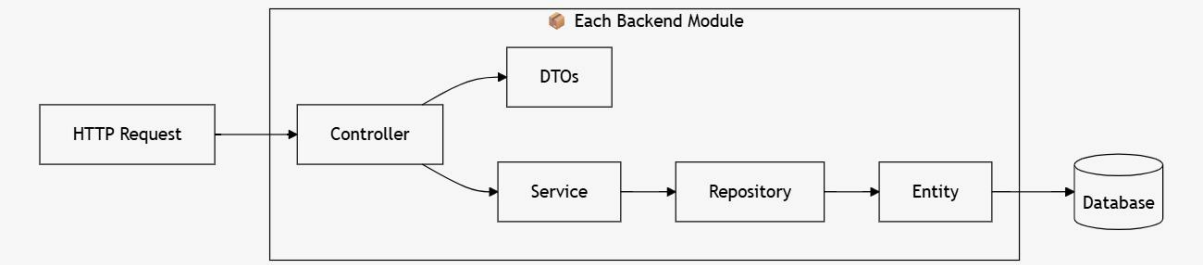


5.4 ERD Design

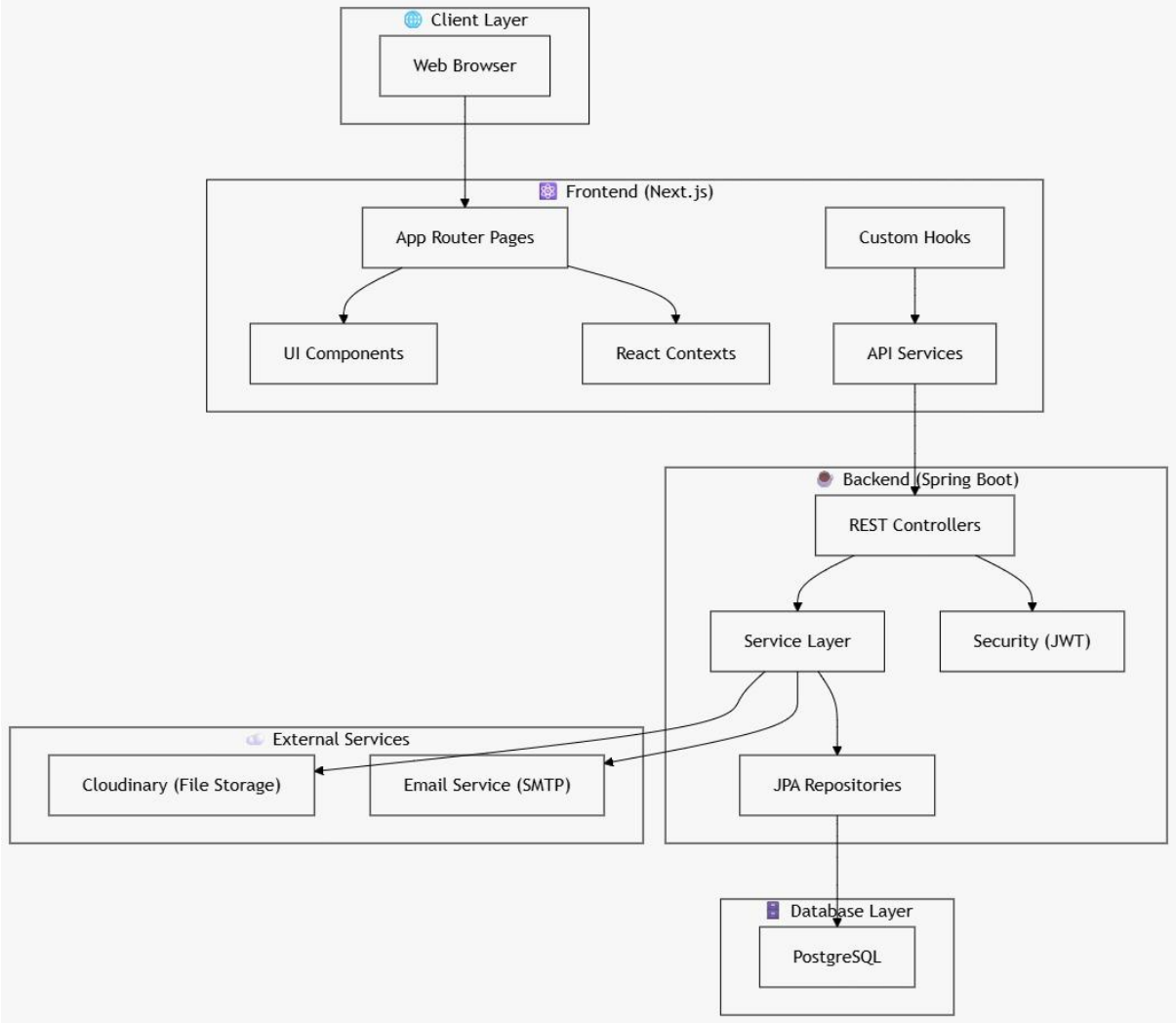


5.5 Architecture Diagrams

5.5.1 Module Layered Diagram



5.5.2 High-Level System Architecture



5.5.3 Security Architecture