

**CEBU INSTITUTE OF TECHNOLOGY
UNIVERSITY**

COLLEGE OF COMPUTER STUDIES

Software Design Description

for

ALLY: Leveraging Digital Platforms to Enhance Access to Legal Aid

Signature

Change History

Authors	Date	Description of Version	Version
Agramon, Vicci Louise Enriquez, Piolo Frances L. Largoza, Darwin Darryl Jean Malagapo, Nathan Rener S. Pangalay, Jerjen Res T.	March 22, 2025	Initial Draft	1.0
Agramon, Vicci Louise Enriquez, Piolo Frances L.	April 2, 2025	Revisions	2.0
Enriquez, Piolo Frances L.	May 12, 2025	Updated Version	2.1



Preface



Table of Contents

1.	INTRODUCTION	6
1.1.	PURPOSE	6
1.2.	SCOPE	6
1.3.	DEFINITIONS AND ACRONYMS	8
1.4.	REFERENCES	8
2.	ARCHITECTURAL DESIGN	9
3.	DETAILED DESIGN	10
	<i>Module 1</i>	<i>10</i>
	<i>Module 2</i>	<i>29</i>

1. Introduction

1.1. Purpose

ALLY is a digital legal platform designed to facilitate engagement, case-matching, and preliminary consultations between clients and legal professionals. This document provides a detailed explanation of the system's functionality, purpose, and its role in mediating interactions between lawyers and clients. It is intended to serve as a comprehensive guide for the following stakeholders:

- **Stakeholders:** Partners and investors involved in the development and deployment of the software.
- **Law Firms and Legal Practitioners:** Primary users of the system who will leverage its features to connect with clients and manage cases.
- **Engineering Team:** Developers responsible for implementing the system's functionalities based on this document.
- **Users:** Your everyday people that want to know about certain legal processes and laws.
- **QA Test Teams:** Testers who will validate each function outlined in this document to ensure system reliability and performance.

1.2. Scope

The proposed system, ALLY, is designed to bridge the gap between potential clients and legal professionals by providing a digital platform that simplifies the process of finding, consulting, and collaborating with lawyers. The system aims to make legal services more accessible, efficient, and secure. Below are the key capabilities of the software:

Client-Lawyer Matching

An AI-powered matching system connects clients with the most suitable legal professionals based on their legal needs, location, and case type.

Lawyers can manage their profiles, expertise, and availability through a dedicated dashboard, ensuring accurate and up-to-date information for matching.

Virtual Organization of Legal Processes

Clients can securely upload and manage legal documents, enabling lawyers to access case files remotely.

A centralized document repository ensures all relevant information is stored in one location, reducing the need for physical paperwork and improving efficiency.

Preliminary AI consultation

The AI-powered preliminary legal consultation analyzes case details by comparing them to past cases, assessing the likelihood of legal action, and estimating potential success rates based on similar case outcomes.

It provides users with insights on whether pursuing legal action is advisable and suggests possible legal strategies. By leveraging data patterns and legal trends, the system offers an informed starting point before formal legal consultation.

On-the-Spot Online Consultations

A temporary messaging system allows clients to seek immediate legal advice without lengthy appointment scheduling. Conversations are designed to be temporary, ensuring privacy and security by not retaining long-term communication history.

Case Tracking and Updates

Clients receive automated email updates about the progress of their legal matters, notifying them of important developments or required actions. A unique case tracking ID is assigned to each case, enabling clients to independently check their case status through the platform.

Educational Resources

A comprehensive **library** of legal resources, including FAQs, templates, and self-help guides, is available to educate users about their rights and legal processes. Resources are accessible offline, ensuring availability even without an internet connection.

Security and Privacy

End-to-end encryption ensures the confidentiality of all user data and communications. Multi-factor authentication (MFA) and secure login mechanisms protect user accounts from unauthorized access.

1.3. Definitions and Acronyms

Ally - Anonymous Legal Liaison for You

Ally - A software system that facilitates engagement, case-matching, and preliminary legal consultation to clients.

A.I - Artificial Intelligence

A.I - Technology that allows computers to do tasks that usually needs human intervention/

Database – A structured collection of data stored electronically.

Spring boot – A framework that is highly suited for backend development and uses Java as its native language

SQL - Structured Query Language

SQL - A language used to manipulate and control data.

SQLite - lightweight, serverless, self-contained relational database management system (RDBMS).

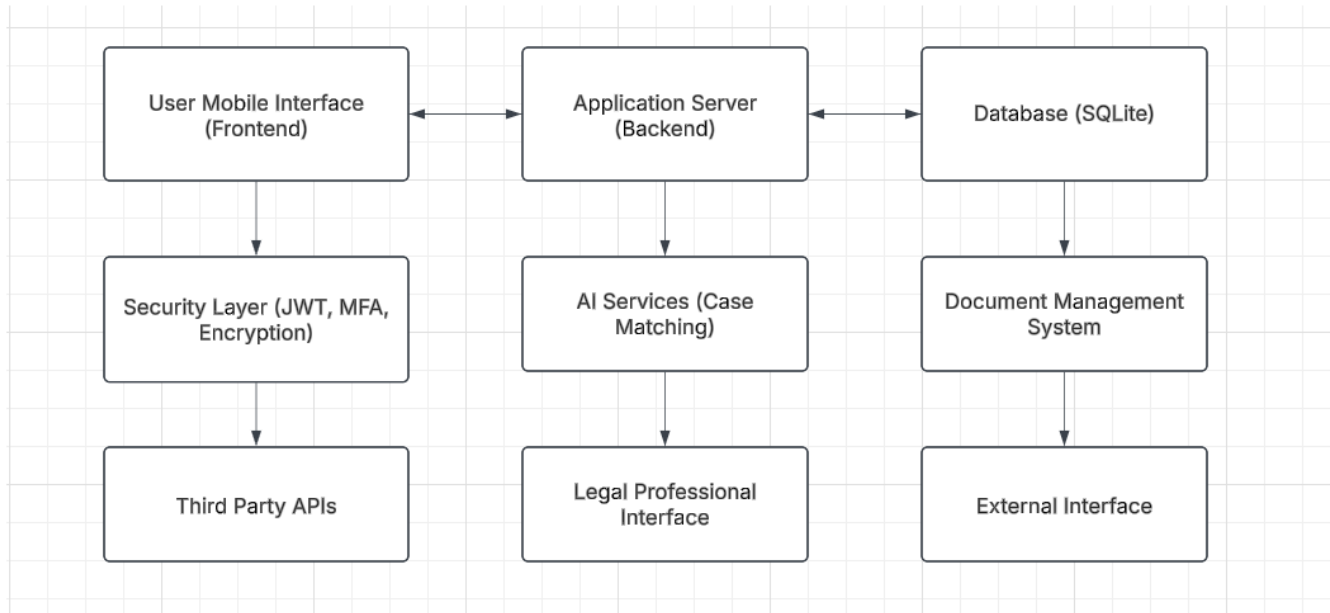
Backend – server-side of a software that handles system functionality.

Frontend – Visual aspect of a software that provides the UI/UX design of the system. What the users see and interact with.

1.4. References

Problem Statement	<u>Problem Exploration for Capstone.docx</u>	A concise document that identifies the problem, the target audience, and the impact of the issue, providing a clear foundation for the project.
Project Proposal	<u>CapstoneProjectProposal.docx</u>	A high-level document that outlines the proposed solution, including objectives, scope, methodology, and expected outcomes.
Software Requirements Specification	<u>Team23-Capstone-SRS.docx</u>	A detailed document that defines the system's functionalities, constraints, and technical requirements, serving as a guide for development.

2. Architectural Design



3. Detailed Design

Module 1 : User Management System

User Registration and Account Management

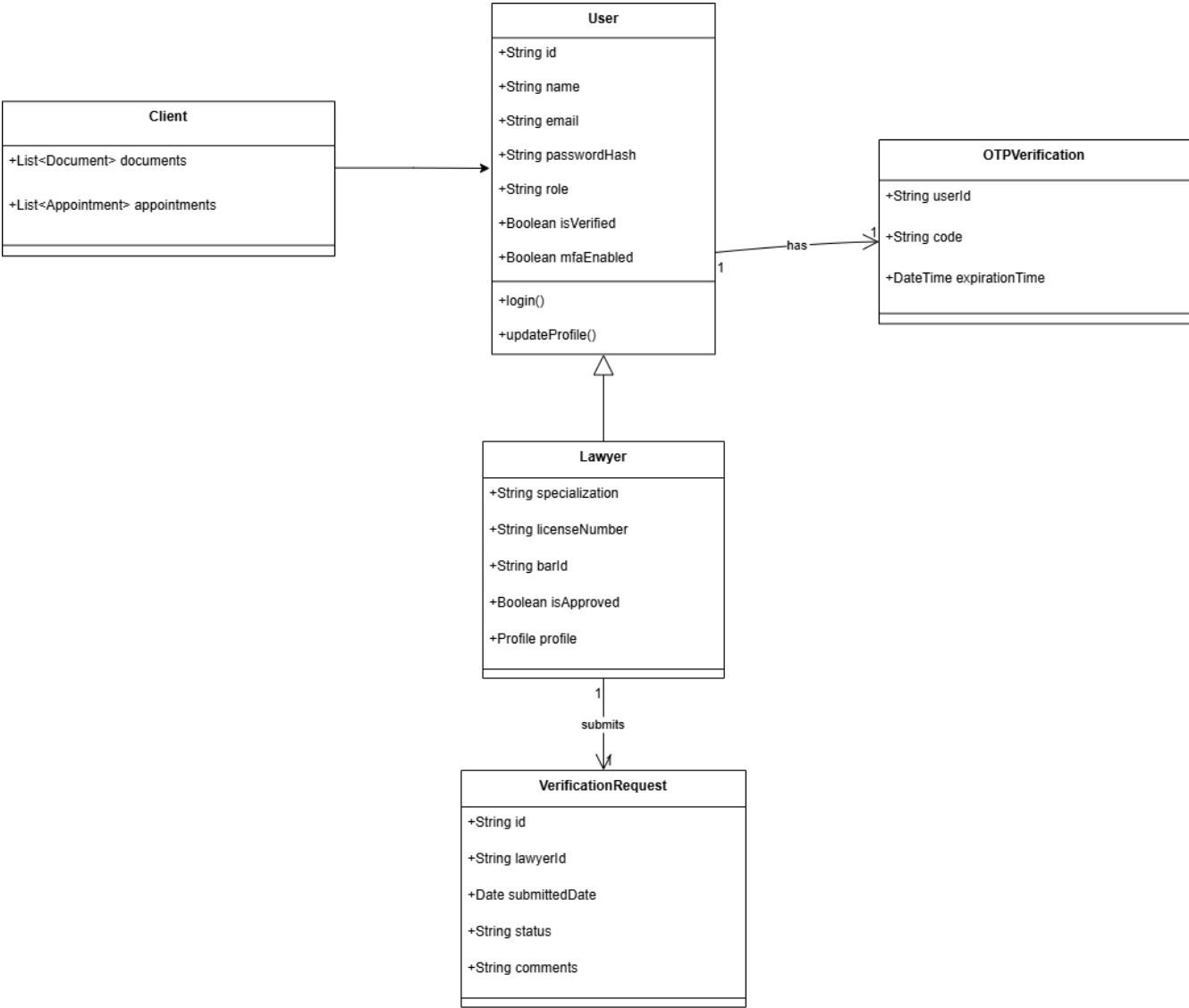
- User Interface Design
- Frontend Components

Component Name	Description and Purpose	Component Type/Format
Client Registration Form	Allows new clients to input their information to create an account.	Web Form (HTML, CSS, JavaScript)
Lawyer Registration Form	Allows new lawyers to input their information and credentials for account creation and verification.	Web Form (HTML, CSS, JavaScript)
Account Dashboard (Client)	Displays client-specific information, access to documents, appointments, and messaging.	Web Page/Dashboard
Account Dashboard (Lawyer)	Displays lawyer-specific information, client cases, appointments, messaging, and profile management.	Web Page/Dashboard
Lawyer Profile Management	Allows lawyers to set and update their professional profile information.	Web Form/Page
Login Form	Allows existing users (Clients, Lawyers, Admins) to authenticate and access the system.	Web Form / Page
Admin Verification Interface	Allows administrators to view pending lawyer registrations, review submitted credentials, and approve or reject lawyer accounts.	Web Page / Dashboard (Admin)

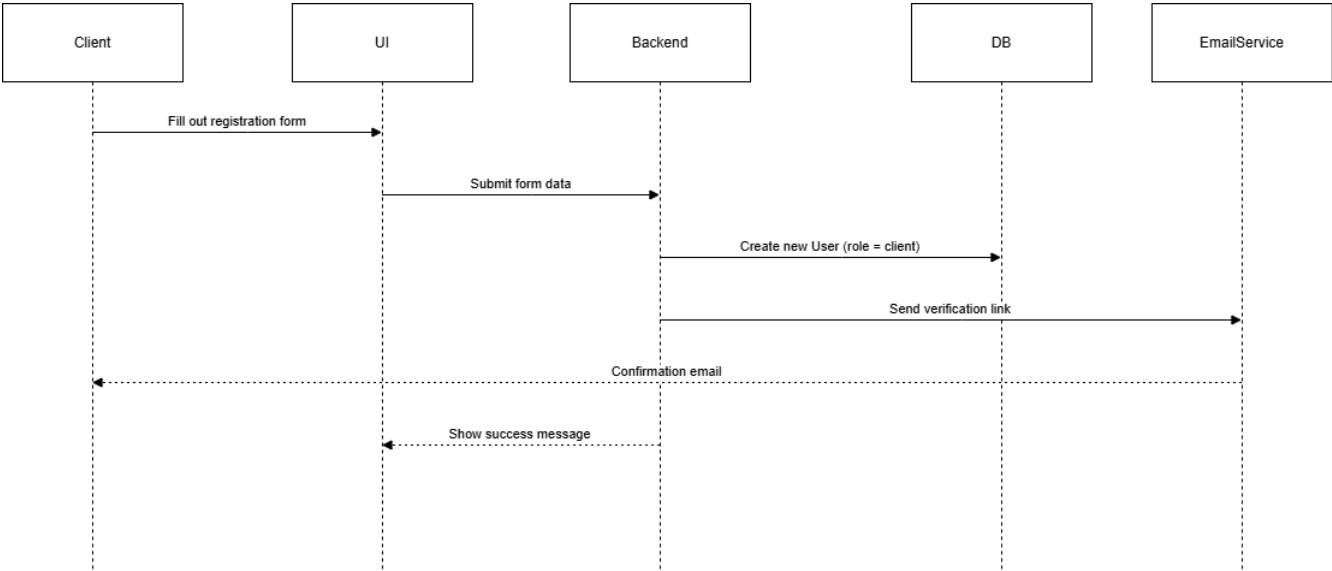
- Backend Component

Component Name	Description and Purpose	Component Type/Format
Registration Handler (Client)	Processes client registration, validates data, creates new accounts in the database.	API Endpoint / RESTful API
Registration Handler (Lawyer)	Processes lawyer registration and credential verification; creates lawyer accounts.	API Endpoint / RESTful API
Account Management Service	Handles login, authentication, profile updates for both clients and lawyers.	API Endpoint / Service
Admin Verification Service	Enables admin review and approval of lawyer credentials before activation.	API Endpoint / Service
Login Handler	Authenticates users based on provided credentials (email/password), manages sessions, and handles Multi-Factor Authentication (MFA) where applicable.	API Endpoint / Service
Email Service	Manages the sending of emails for various system functions, primarily for account email verification and potentially other notifications.	Backend Service / Utility

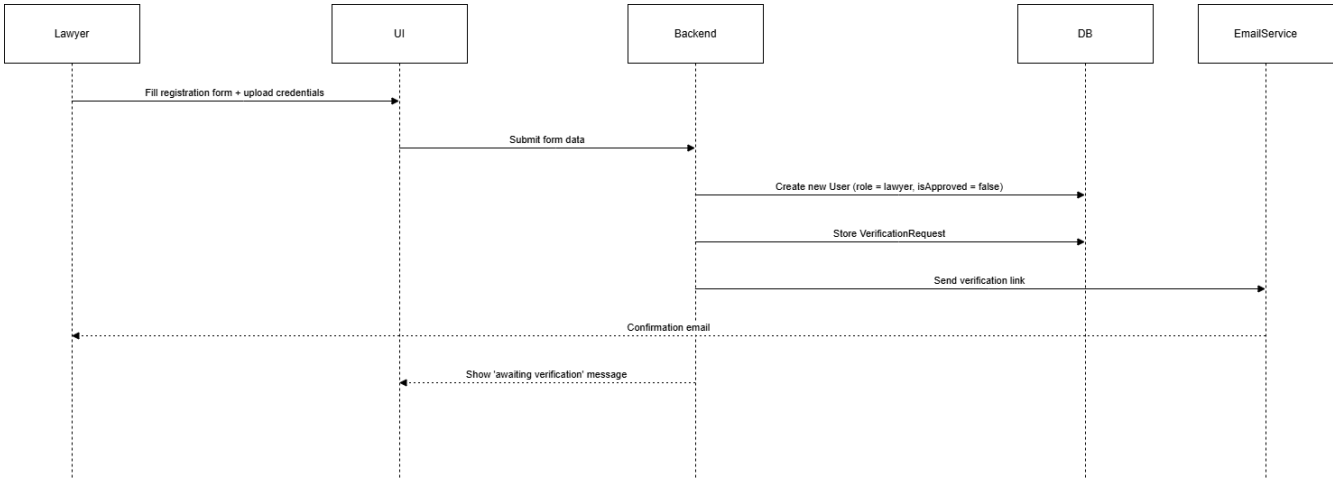
- Object-Oriented Components
 - Class Diagram



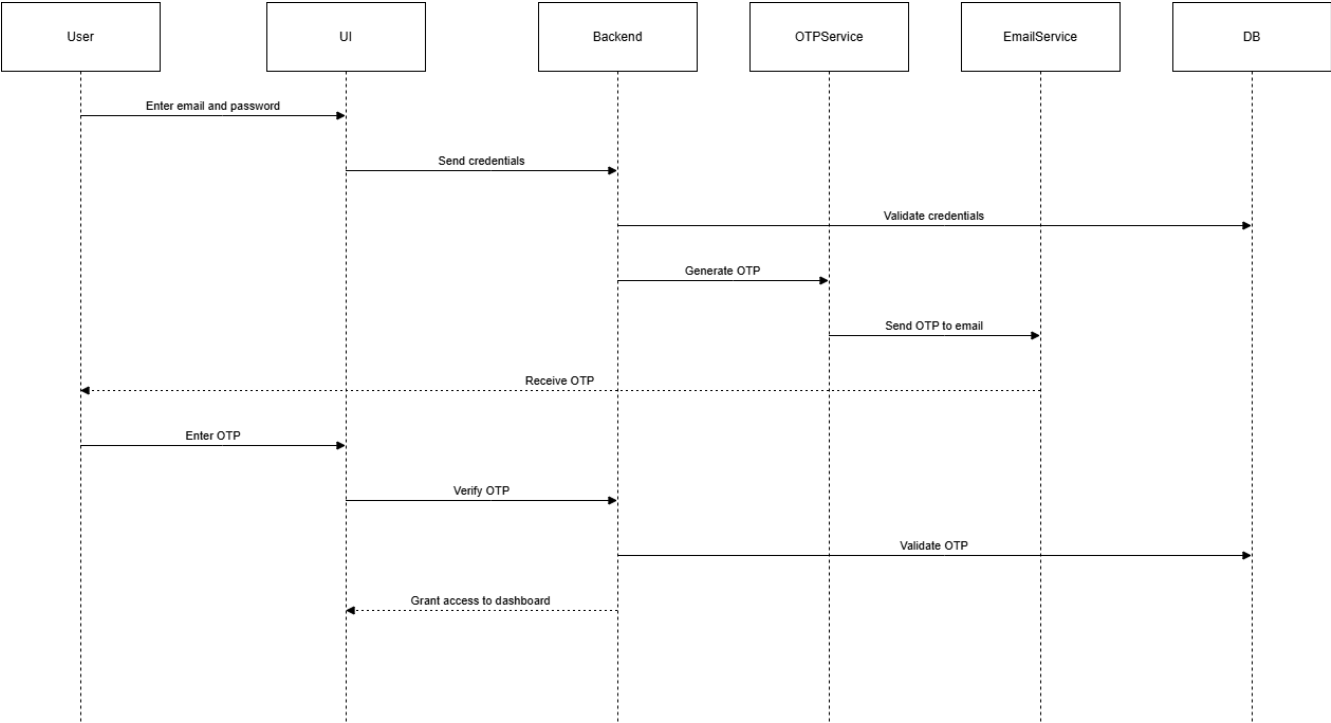
- Sequence Diagram
- Transaction 1.1: Register Client



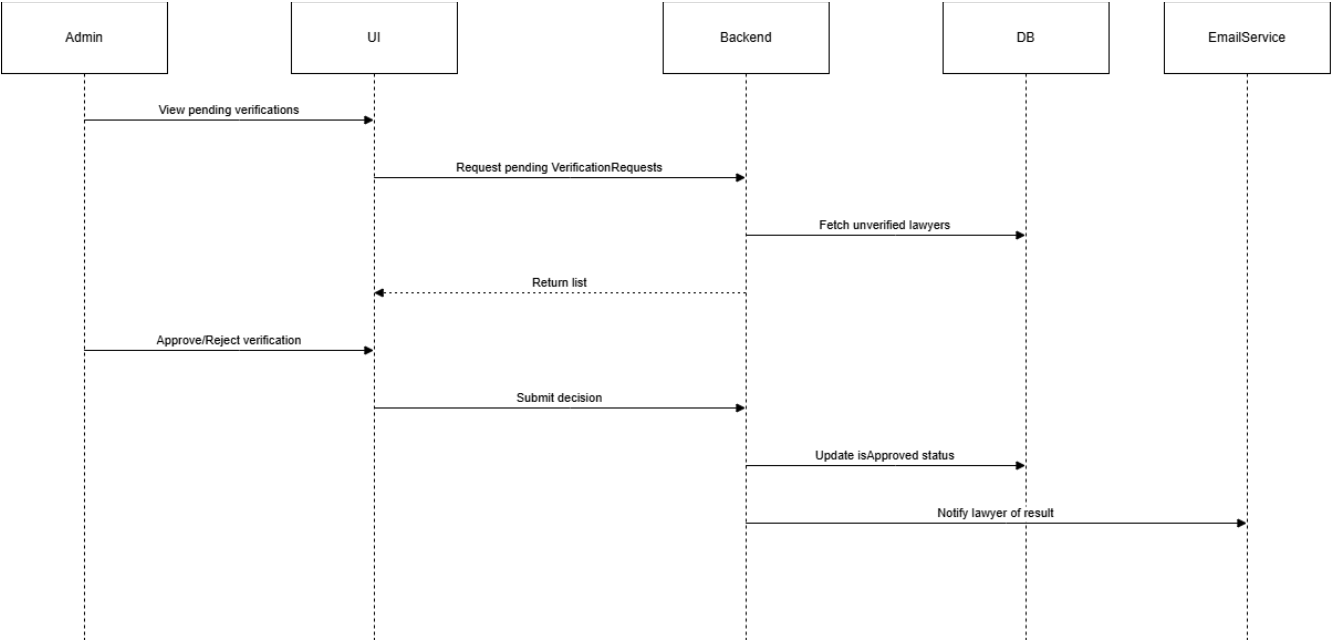
• Transaction 1.2: Register Lawyer



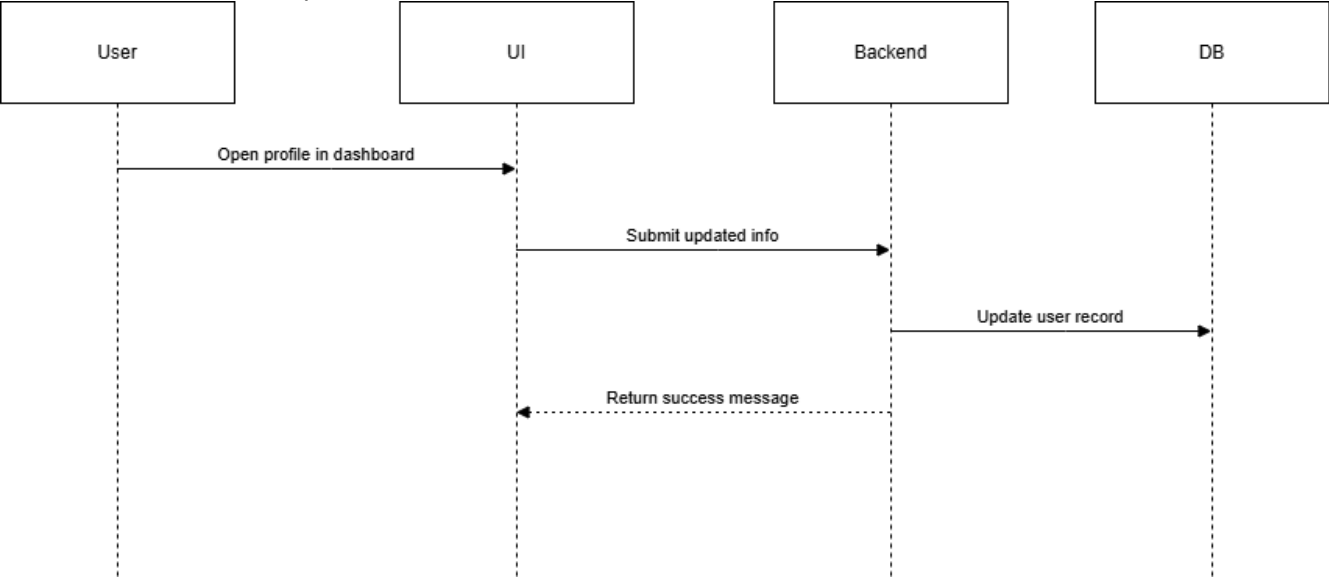
Transaction 1.3: Verify Lawyer



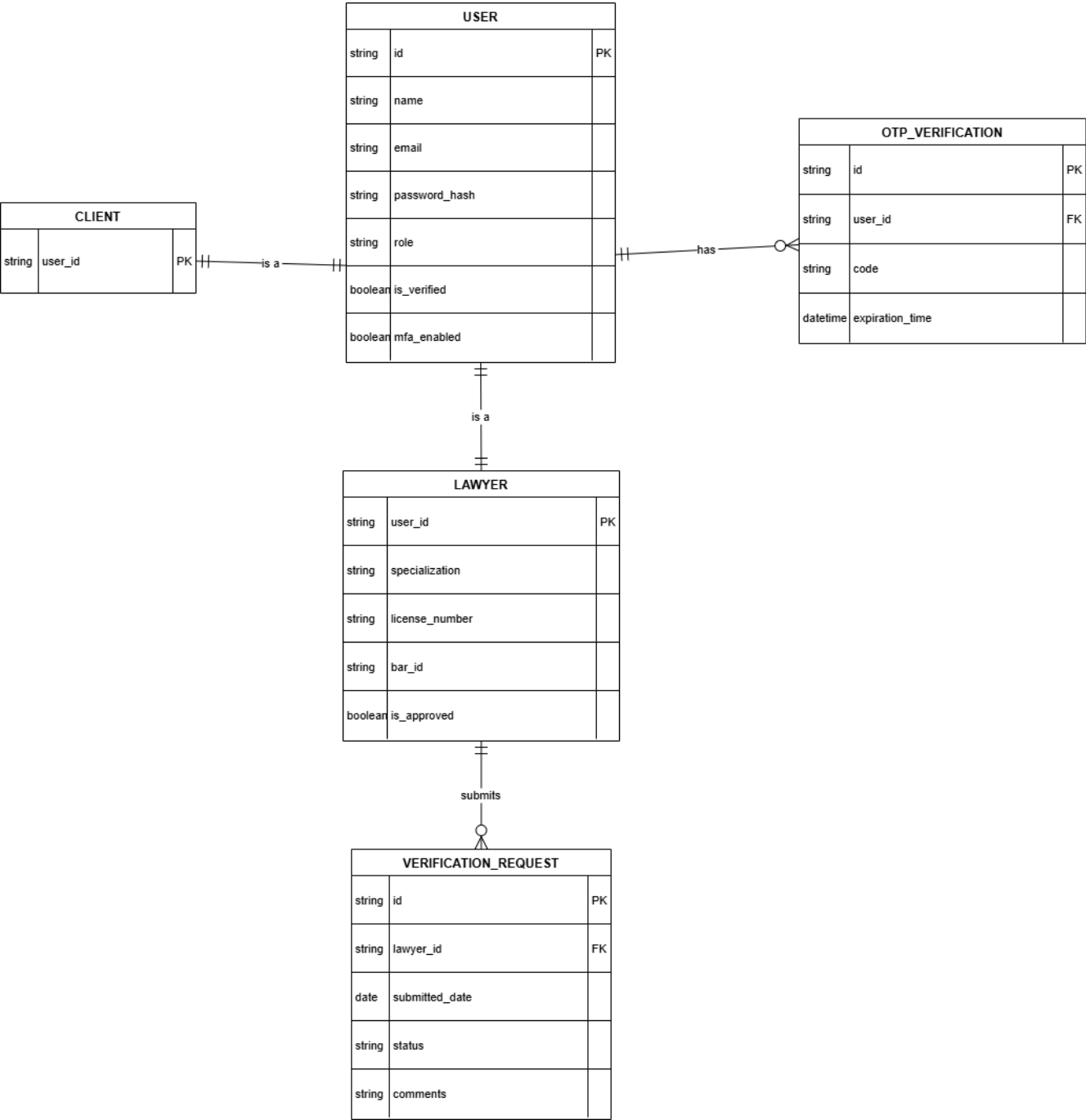
Transaction 1.4: Login



• Transaction 1.5: Update Profile



- Data Design
 - ERD or schema



Module 2 : Anonymous - ALLY AI Consultation

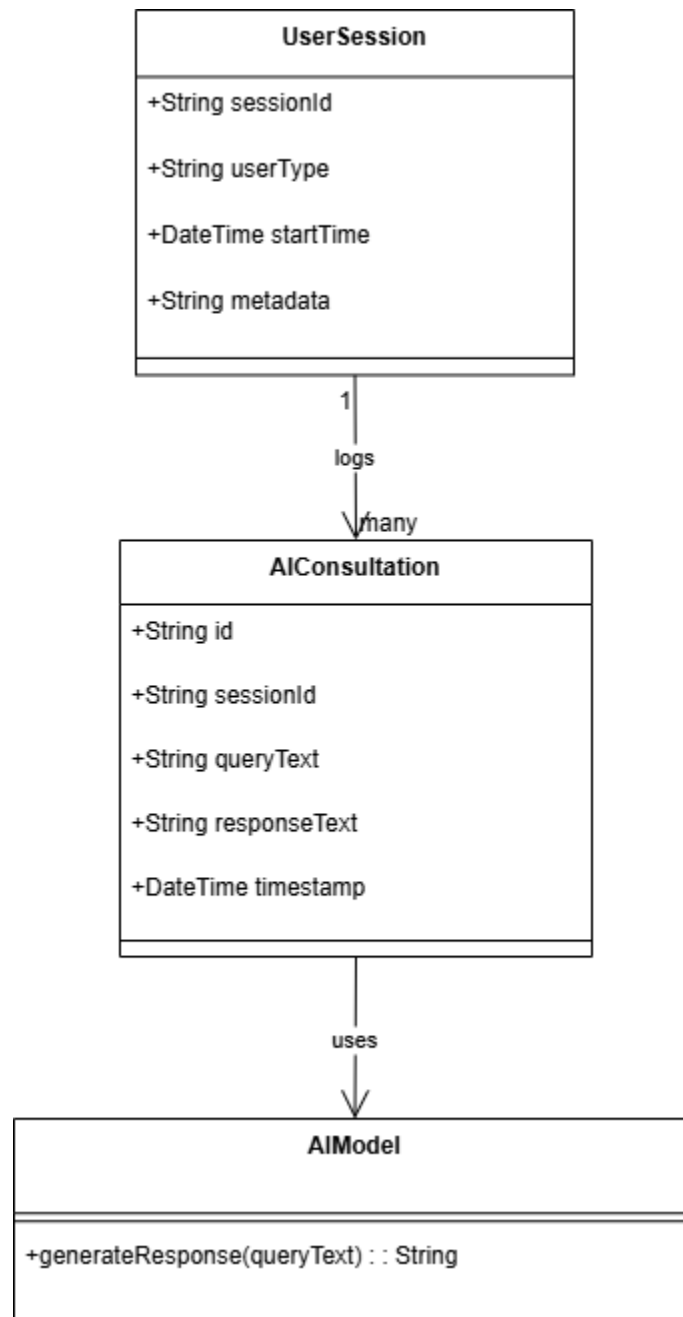
- User Interface Design
 -
- Front-end component(s)

Component Name	Description and Purpose	Component Type/Format
ALLY AI Chat Interface	Provides an interactive chatbot-style interface for users to ask legal questions and receive AI-generated responses.	Mobile interface / Chat Widget (Kotlin)
Session ID Tracker	Handles anonymous session identity (token or UUID) for tracking without personal login.	Kotlin Service /Local Storage
Query submission UI	Captures the user's legal concern input before sending it to the backend.	Text Input / Submit Button
AI Response Display	Dynamically shows the AI-generated legal insights or advice to the user.	Text Display / Scrollable View

- Back-end component(s)

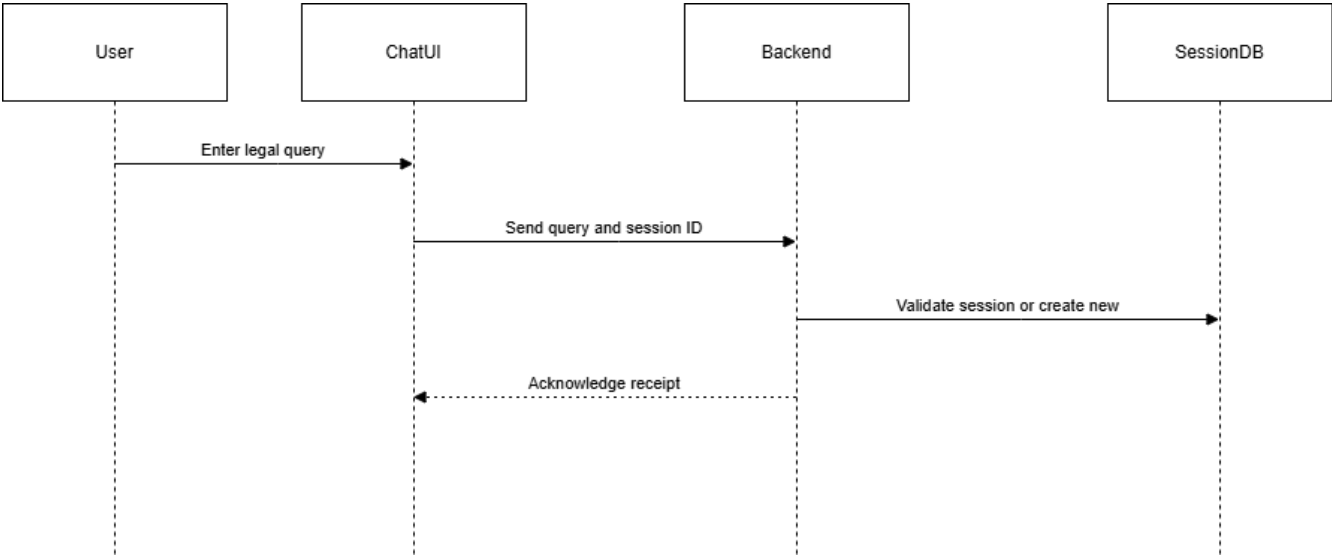
Component Name	Description and Purpose	Component Type/Format
AI Consultation Service	Receives user queries, invokes the AI model (e.g., NLP/LLM), and returns legal information or insights.	API Endpoint / Backend Service
AI Interaction Logger	Records AI interactions for quality improvement, debugging, and feedback tracking.	Backend Service
AI Model Interface	Interfaces with an AI/LLM to generate legal insights from the submitted query.	
Session Manager	Tracks anonymous user sessions with unique IDs to maintain continuity across chat interactions.	

- Object-Oriented Components
 - Class Diagram

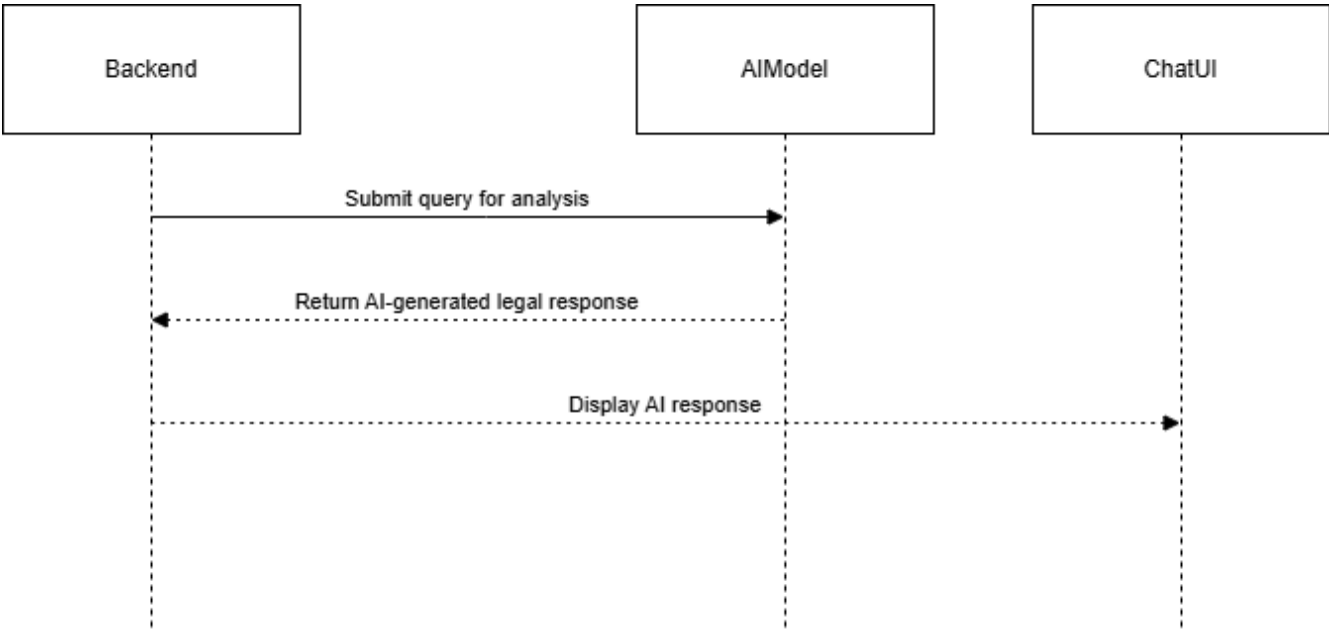


○ Sequence Diagram

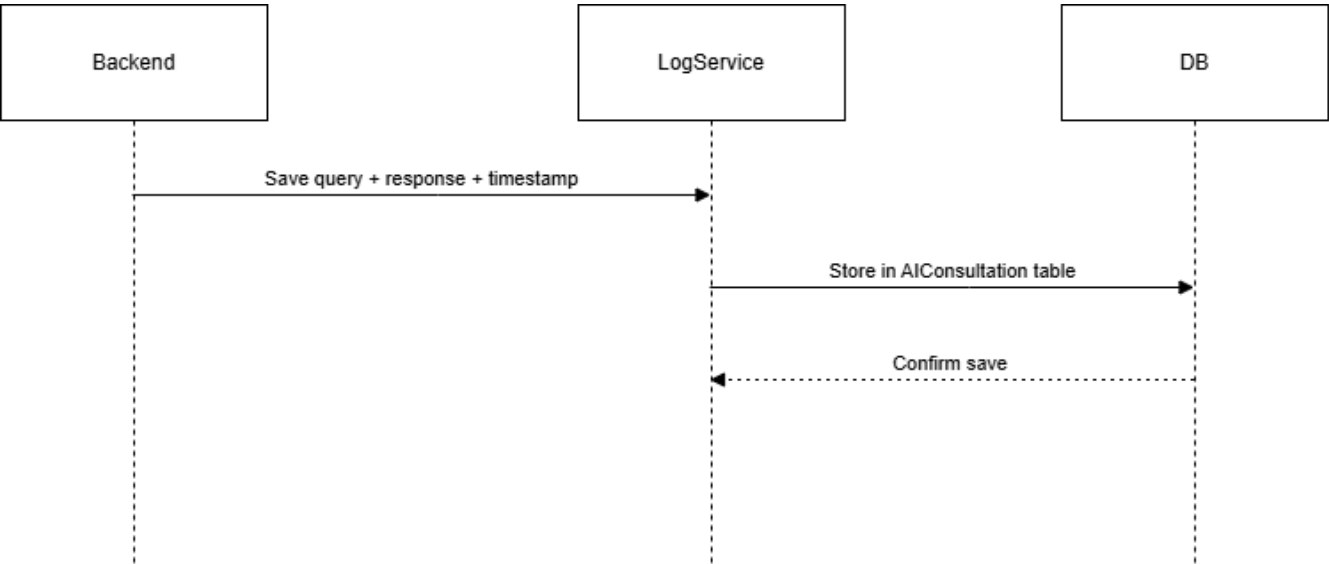
● Transaction 2.1: Submit Legal Query



● Transaction 2.2: Received AI Response

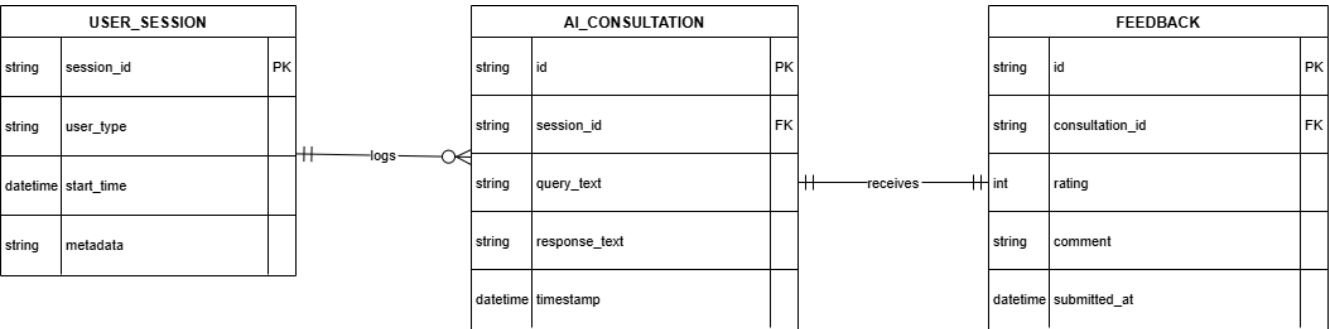


Transaction 2.3: Log Interaction



Data Design

- ERD or schema



Module 3: Lawyer Matching and Selection

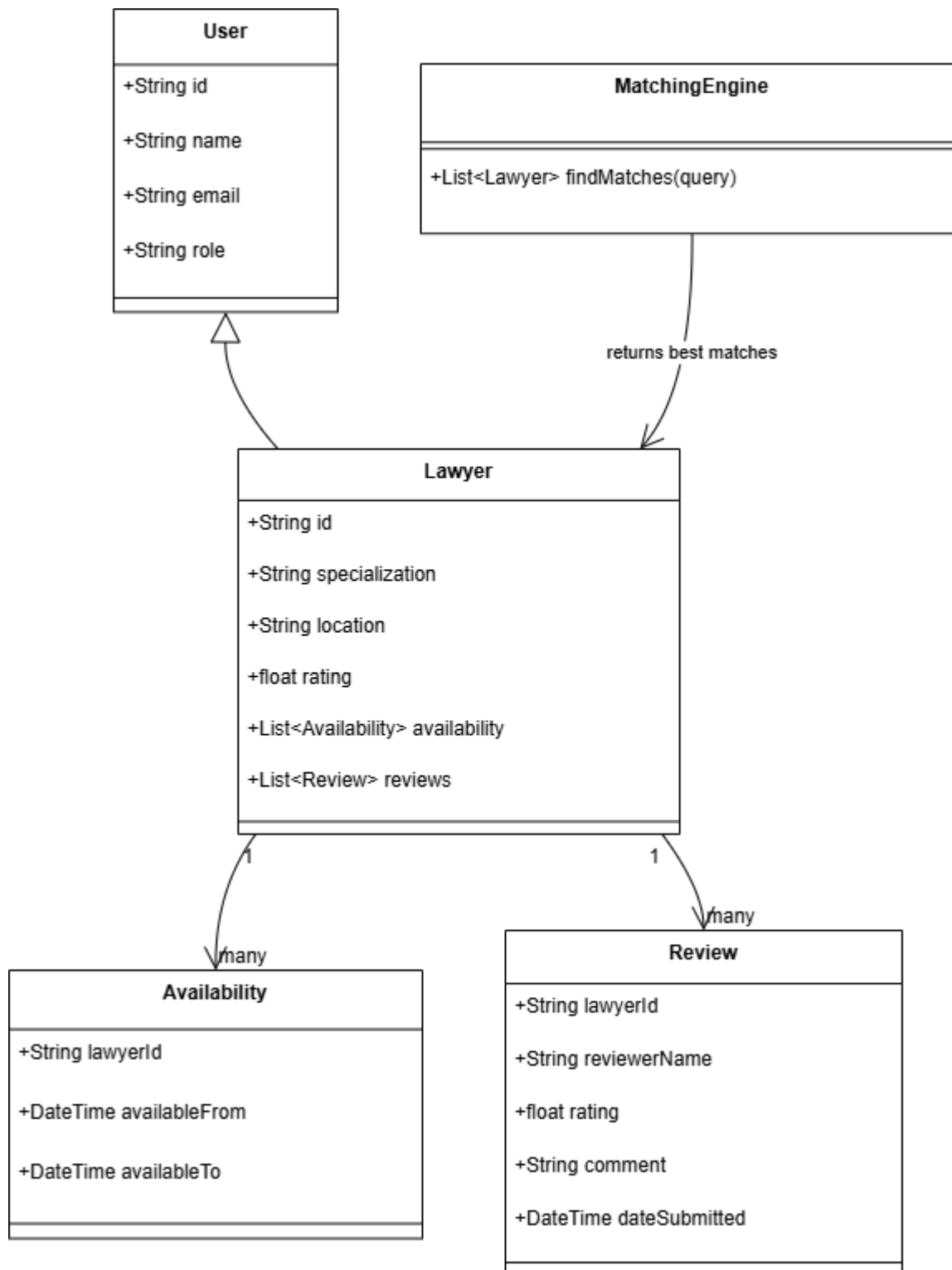
- User Interface Design
 -
- Front-end component(s)

Component Name	Description and Purpose	Component Type/Format
Lawyer Search UI	Allows users to search or filter available lawyers based on expertise, availability, and location.	Web Page (HTML, CSS, JavaScript)
Lawyer Profile View	Displays detailed lawyer information such as biography, specialization, rating, and booking options.	Web Page (HTML, CSS, JavaScript)
AI Match Submission Form	Users submit legal need for AI matching.	Form UI
AI Match Results Display	Displays AI-recommended lawyers sorted by score	Card Grid / Results List

- Back-end component(s)

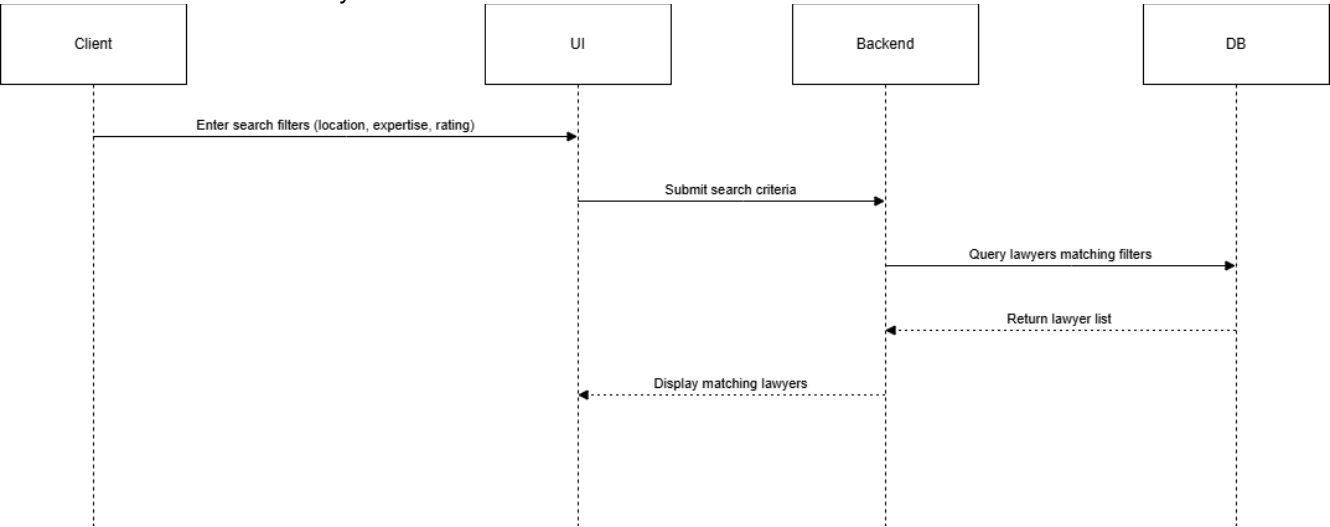
Component Name	Description and Purpose	Component Type/Format
Lawyer Search Service	Handles user search/filtering queries and retrieves relevant lawyer profiles from the database.	API Endpoint / Service
Lawyer Profile Service	Fetches and compiles detailed lawyer info, reviews, and availability.	Springboot API + DB
AI Matching Engine	Applies logic (rule-based logic) to suggest best-fit lawyers.	Service Component
Match History Logger	Optionally logs match sessions for analytics or audit	Backend Logger

- Object-Oriented Components
 - Class Diagram

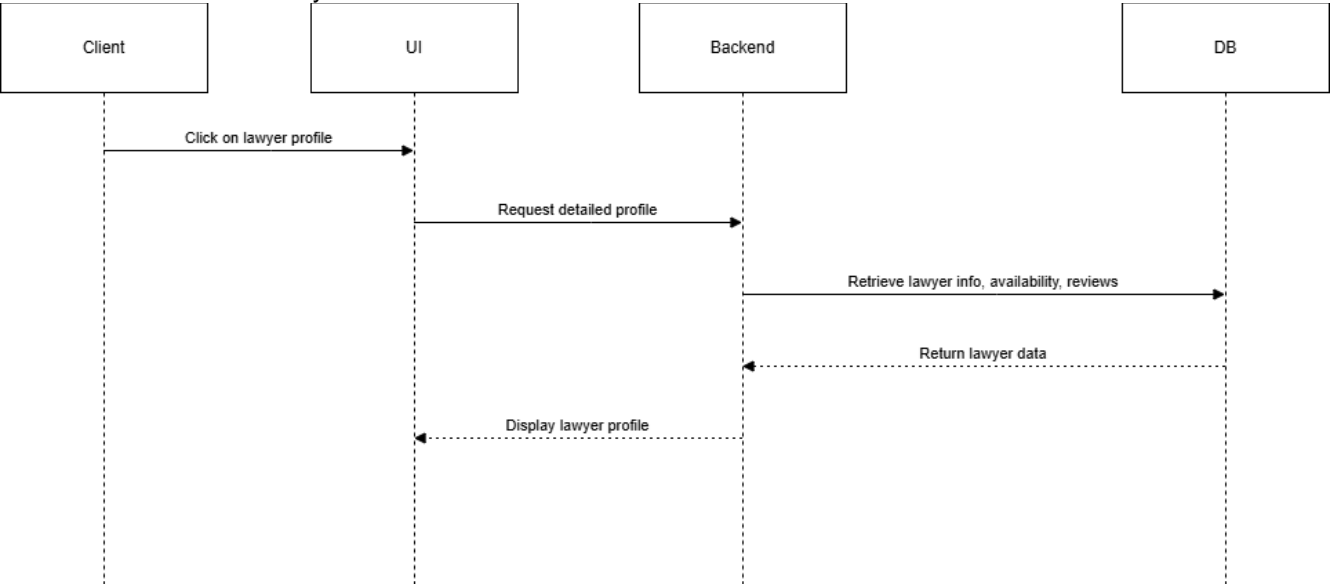


○ Sequence Diagram

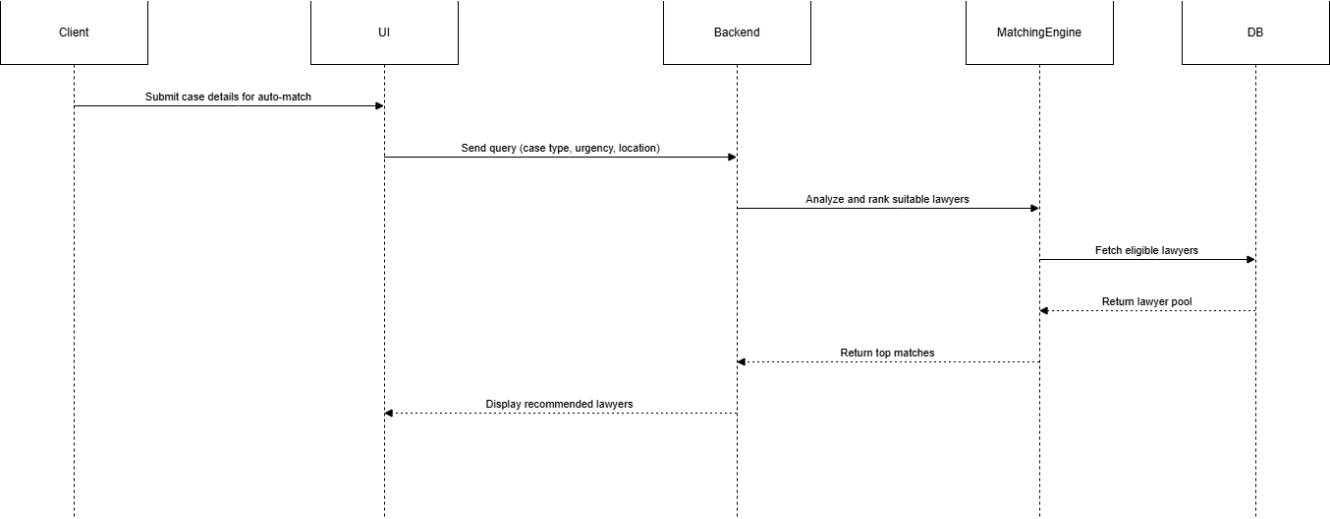
• Transaction 3.1: Search Lawyer



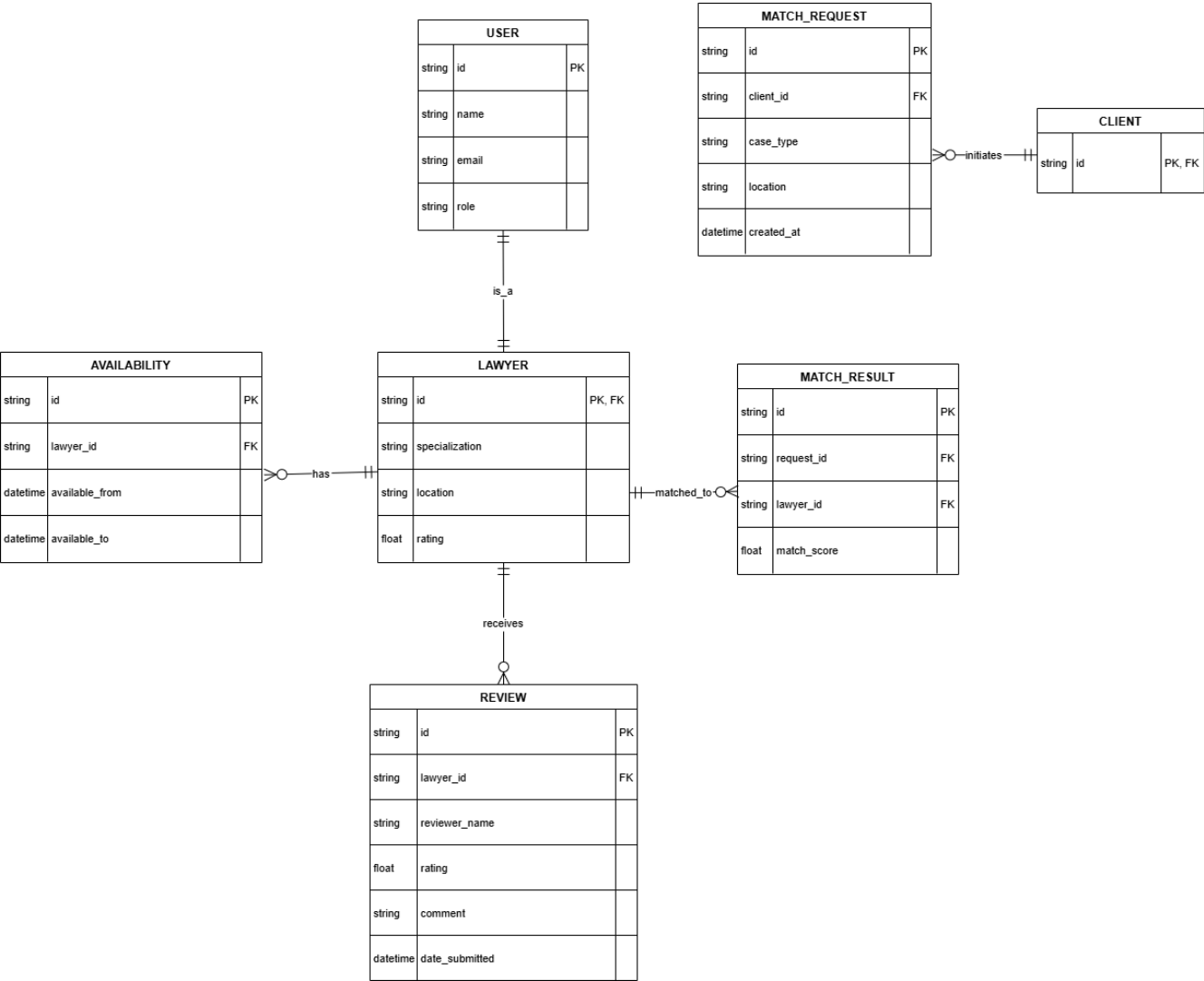
Transaction 3.2: View Lawyer Profile



Transaction 3.3: Match via AI



- Data Design
 - ERD or schema



Module 4: Document Submission (Registered Clients)

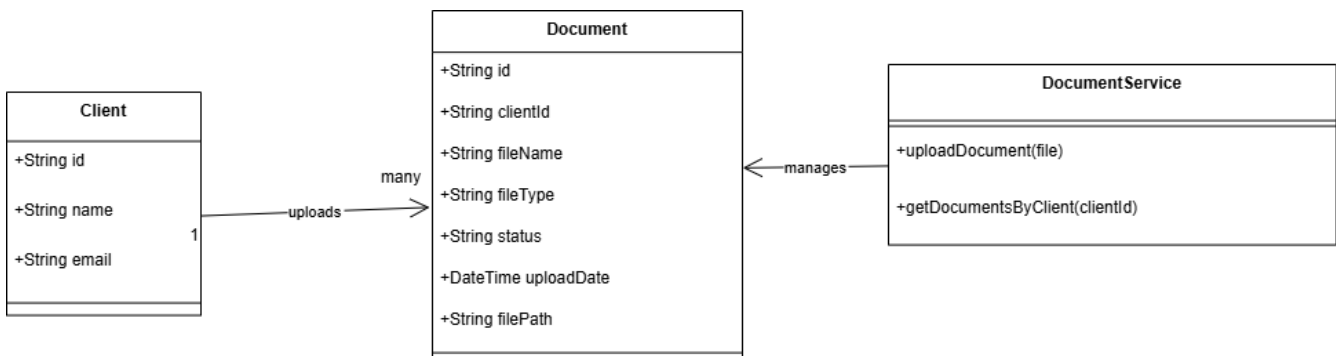
- User Interface Design
- Front-end component(s)

Component Name	Description and Purpose	Component Type/Format
Document Upload Interface	Allows clients to upload legal documents for lawyer review.	Web Form (HTML, CSS, JavaScript)
Documents List View	Displays all previously uploaded documents by the client, with status and download links.	Web Form (HTML, CSS, JavaScript)
Upload Status Notification	Provides feedback on upload. (success, error, file type restrictions, etc.)	Toast / Modal

- Back-end component(s)

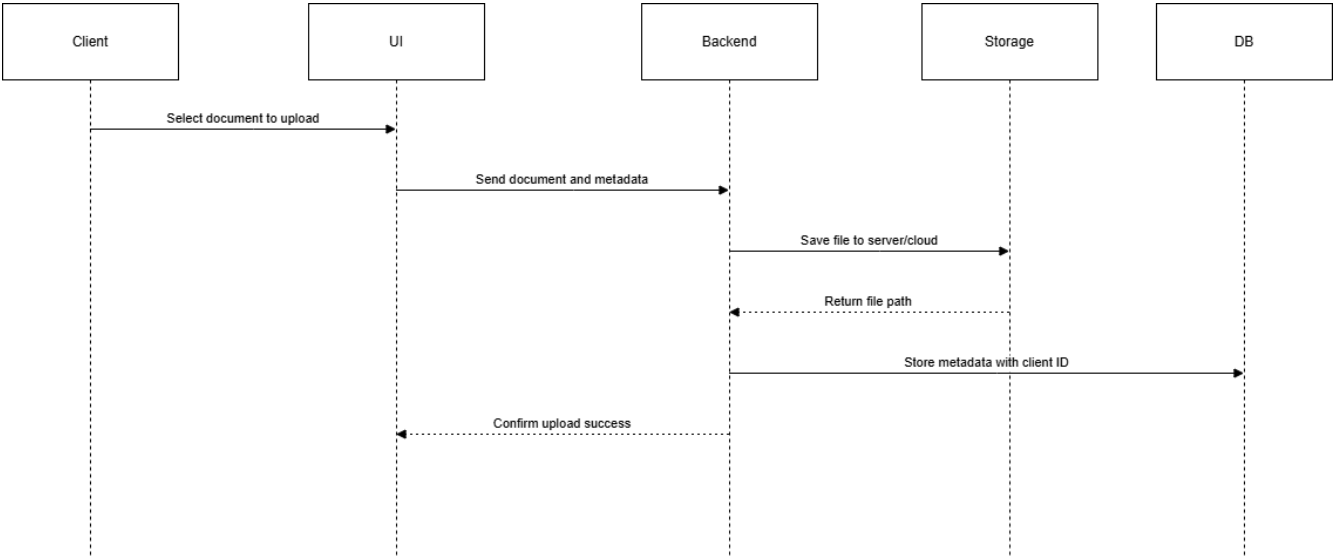
Component Name	Description and Purpose	Component Type/Format
Document Upload Service	Handles incoming file uploads, performs validation, and stores documents securely.	API Endpoint / Service
Document Metadata Manager	Stores and retrieves document records tied to a client.	JPA Repository
File Storage Engine	Save files to secure location (cloud or local path)	Firebase storage
Document Status Updater	Updates status of documents (e.g., pending, review, approved)	API Service

- Object-Oriented Components
 - Class Diagram

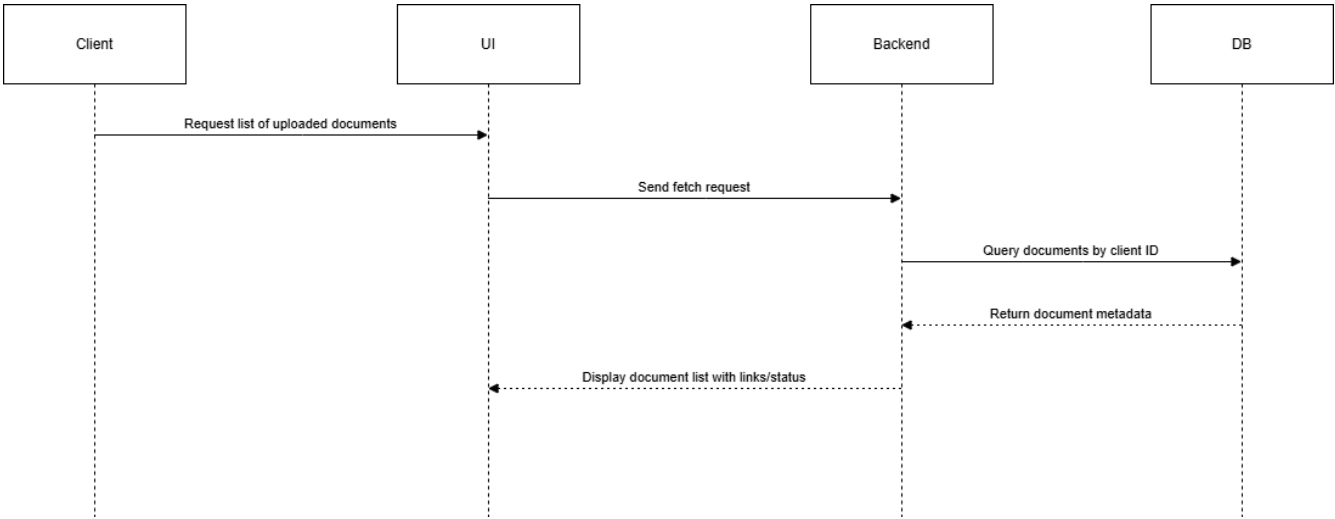


- Sequence Diagram

- Transaction 4.1: Upload Document

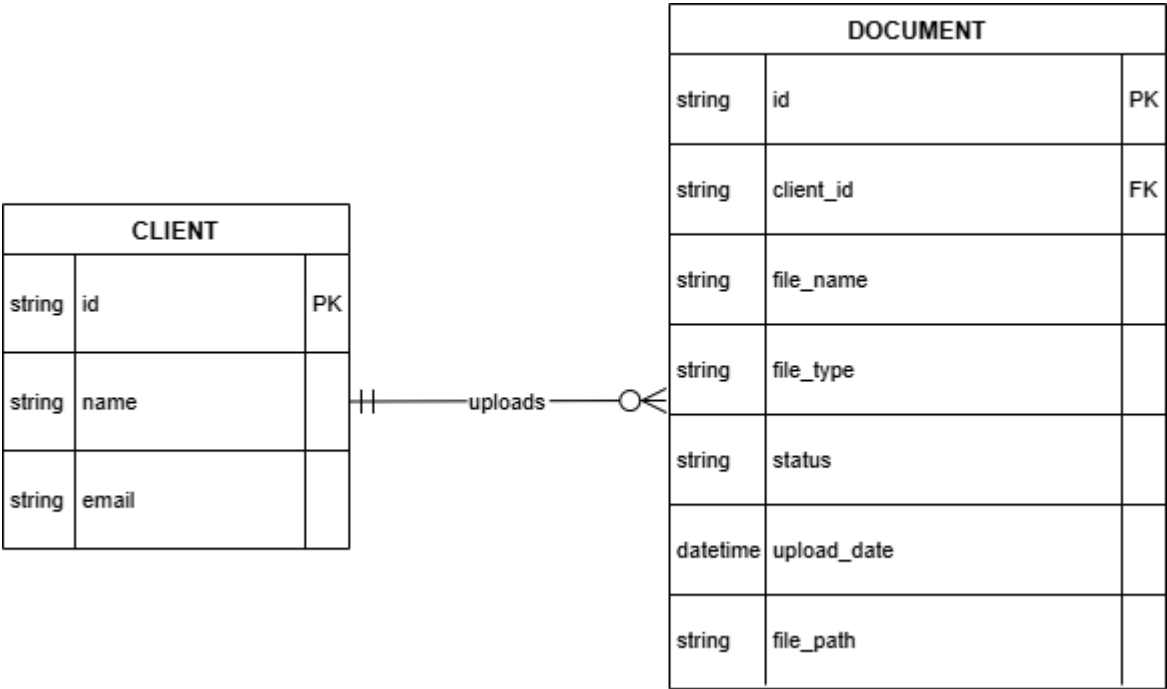


- Transaction 4.2: View Document List



- Data Design

- ERD or schema



Module 5: Appointment Scheduling

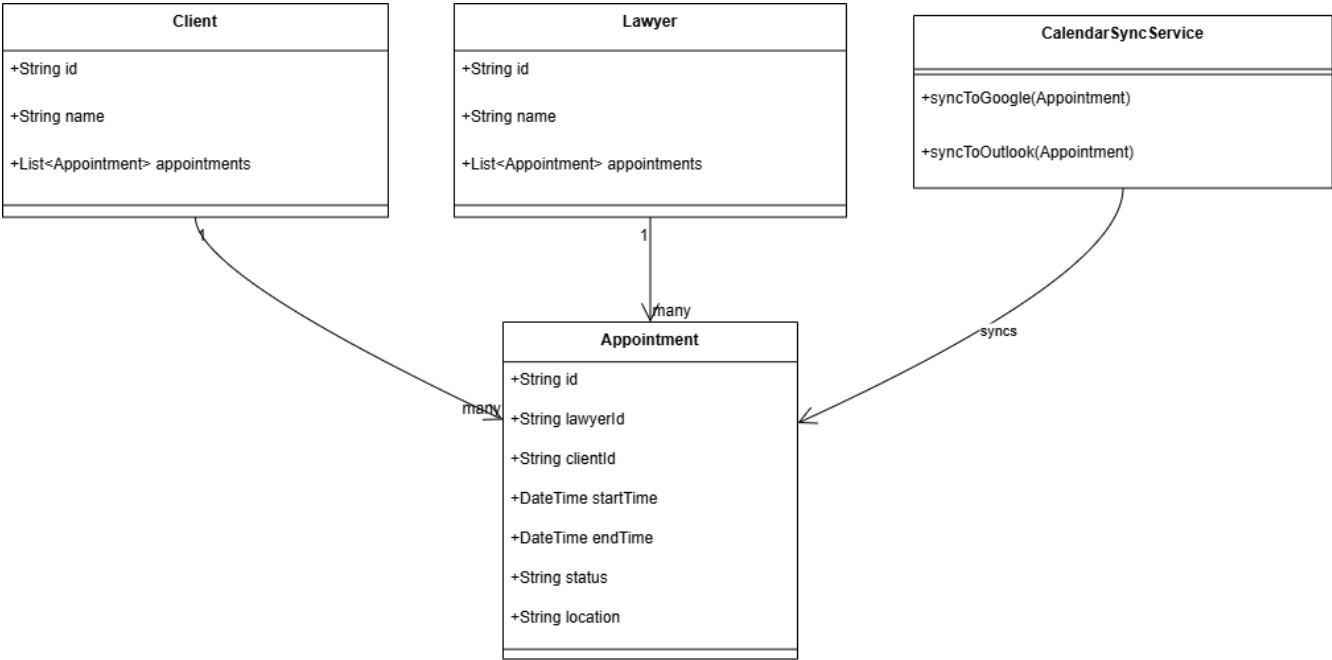
- User Interface Design
- Front-end component(s)

Component Name	Description and Purpose	Component Type / Format
Appointment Booking Interface	Enables clients to browse lawyer availability and book time slots.	Calendar Picker / Form UI
Appointment Dashboard	Shows upcoming, past, or canceled appointments for clients/lawyers.	Calendar View / Table
Edit/Cancel Form	Allow users to reschedule or cancel appointments.	Modal Inline Form
Sync Calendar Toggle	Lets users opt to sync appointments to Google	Toggle + OAuth Integration UI

- Back-end component(s)

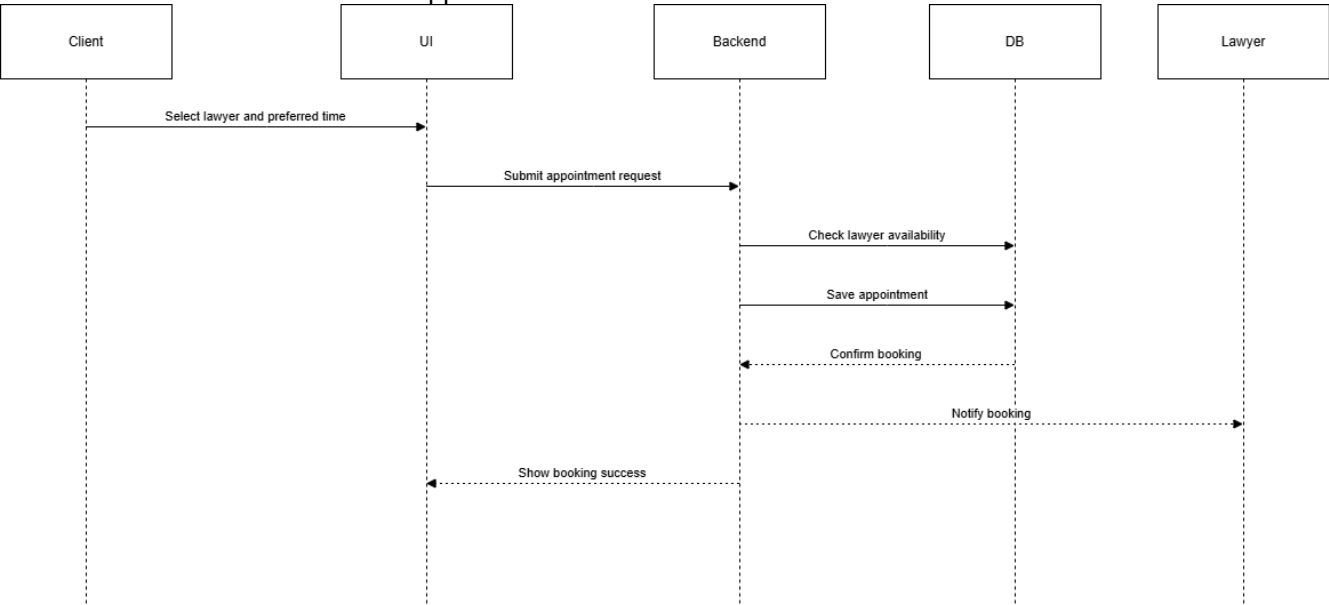
Component Name	Description and Purpose	Component Type / Format
Appointment Scheduling Service	Handles booking, validation, conflict checking, and calendar updates.	API Endpoint / Backend Service
Availability Checker	Ensures that requested slot does not overlap or conflict	Service class
Appointment Manager	Handles CRUD for appointment records	JPA Repository
Calendar Sync Service	Connects with external APIs (Google) to sync confirmed appointments	OAuth2 Integration, Google API

- Object-Oriented Components
 - Class Diagram

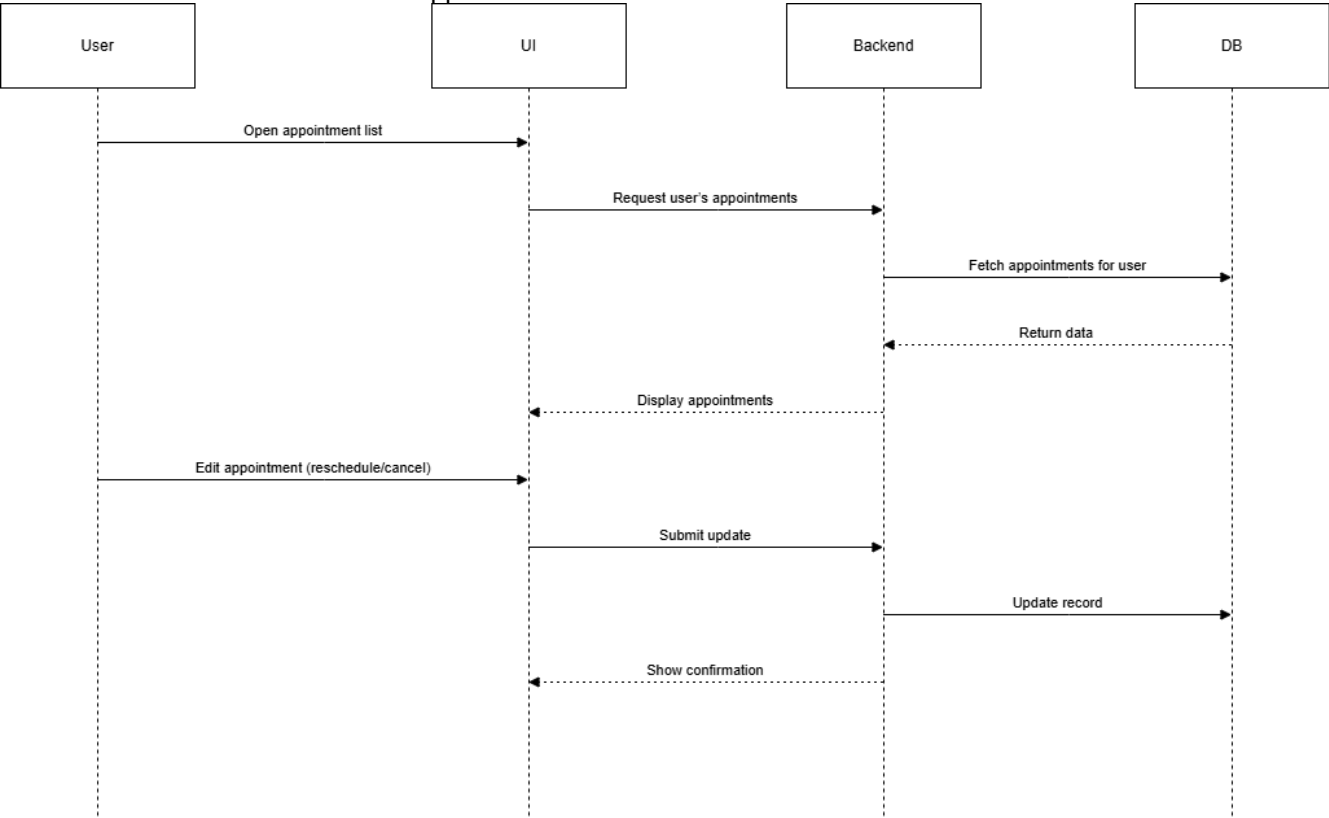


- Sequence Diagram

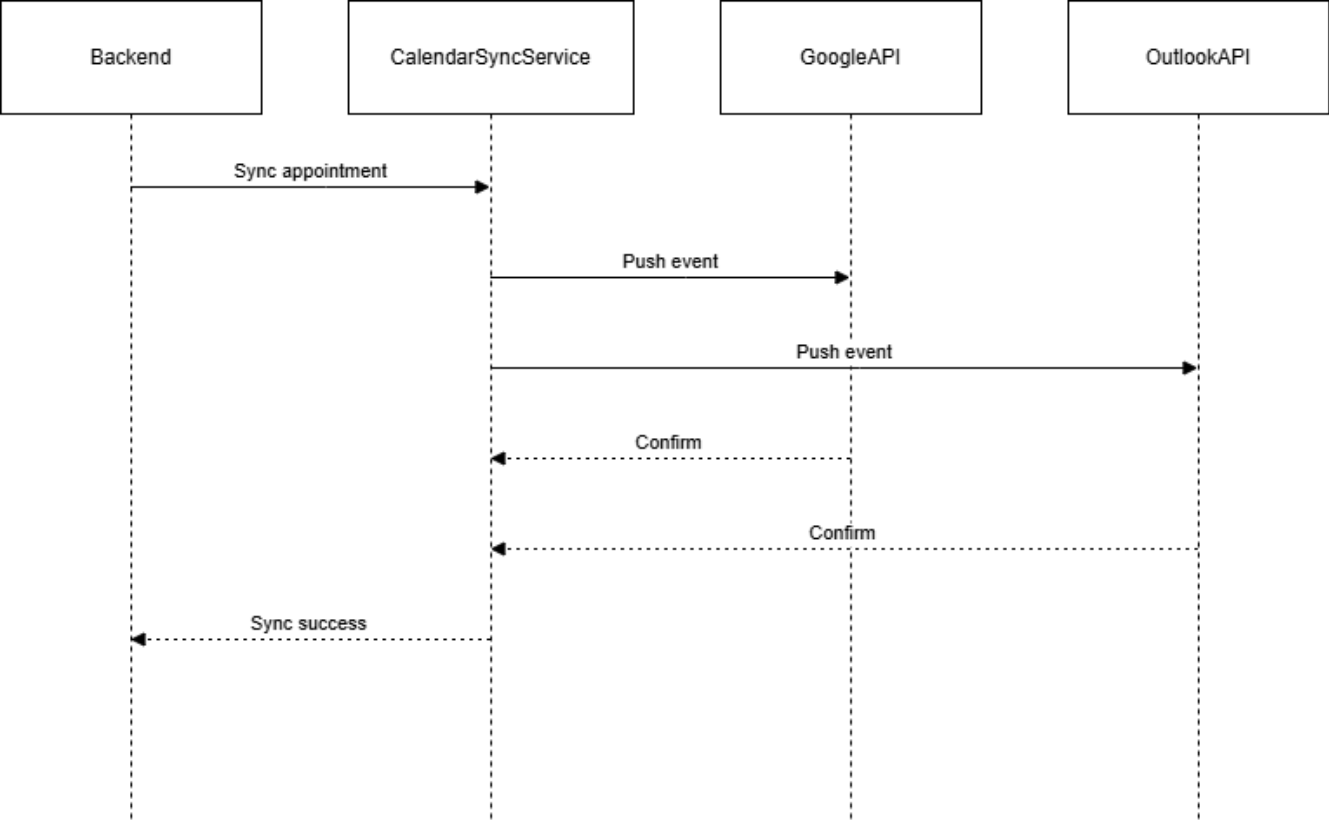
• Transaction 5.1: Book Appointment



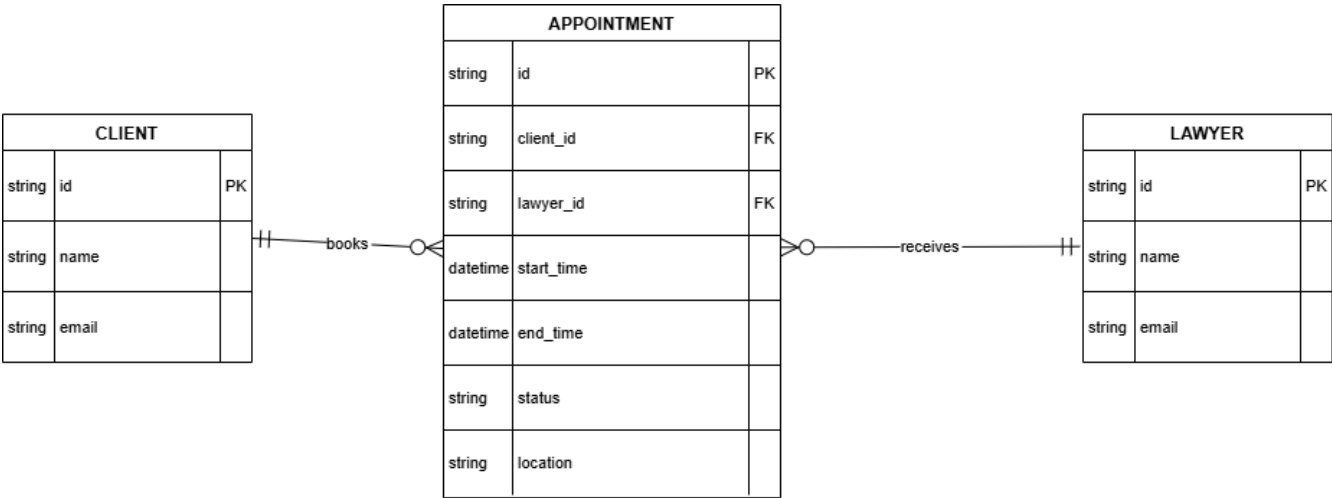
• Transaction 5.2: View/Edit Appointment



Transaction 5.3: Calendar Sync



- Data Design
 - ERD or schema



Module 6: Messaging System

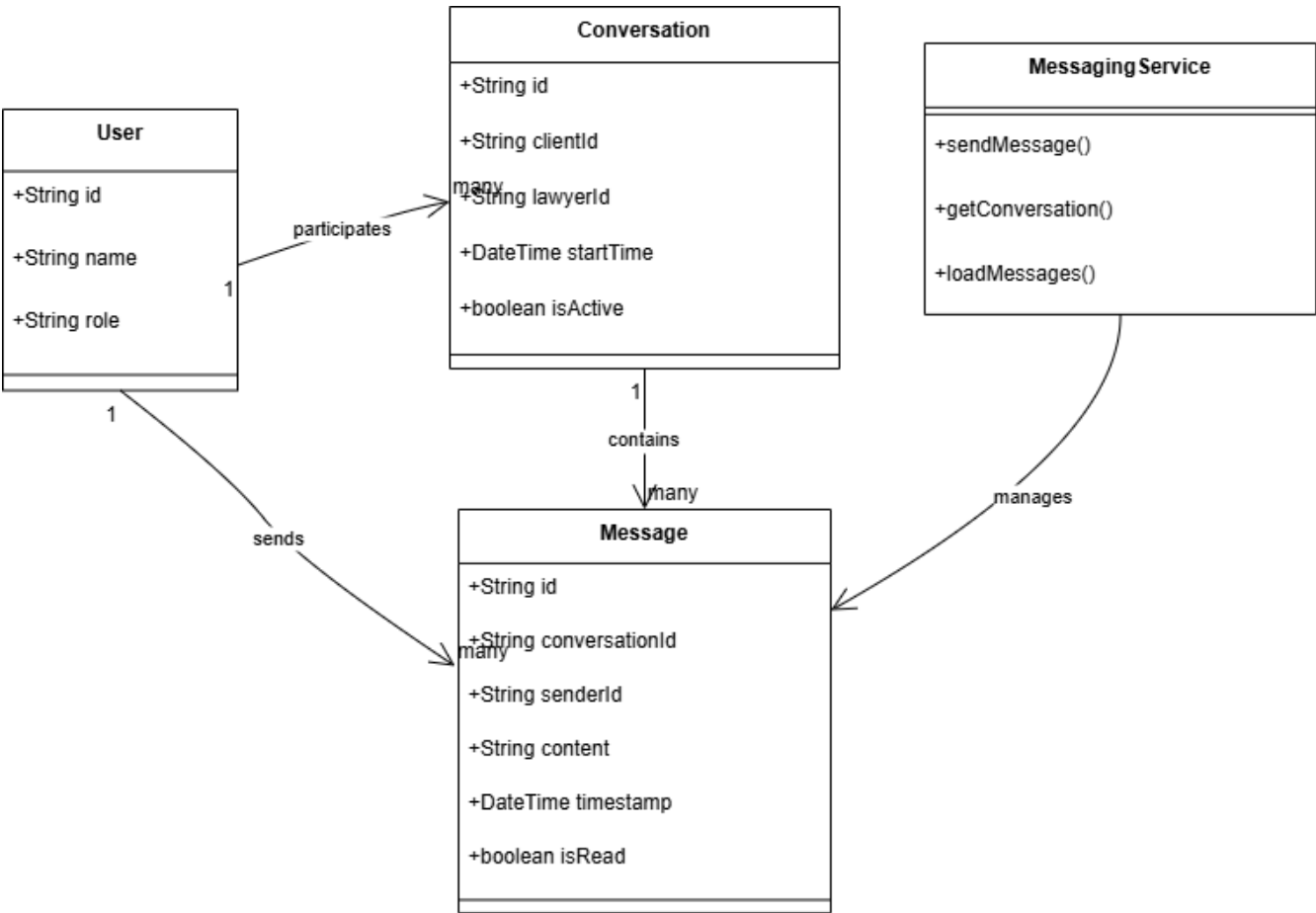
- User Interface Design
- Front-end component(s)

Component Name	Description and Purpose	Component Type / Format
Messaging Interface	Provides real-time or threaded chat experience for secure lawyer-client conversations.	Web Page / Chat Widget (HTML, CSS, JavaScript)
Message Input Form	Text input area with send button	Text area + button
Firestore Listener Setup	Subscribes to real-time updates in conversation collection.	onSnapshot() (Firestore Listener)
Message List Renderer	Dynamically renders messages as they arrive.	DOM rendering / Firebase binding
Road Receipt Updater	Marks messages as read upon opening	Firestore doc update
Firebase Auth Token Pass	Authenticate current user with Firebase identity	Firebase Auth (email)

- Back-end component(s)

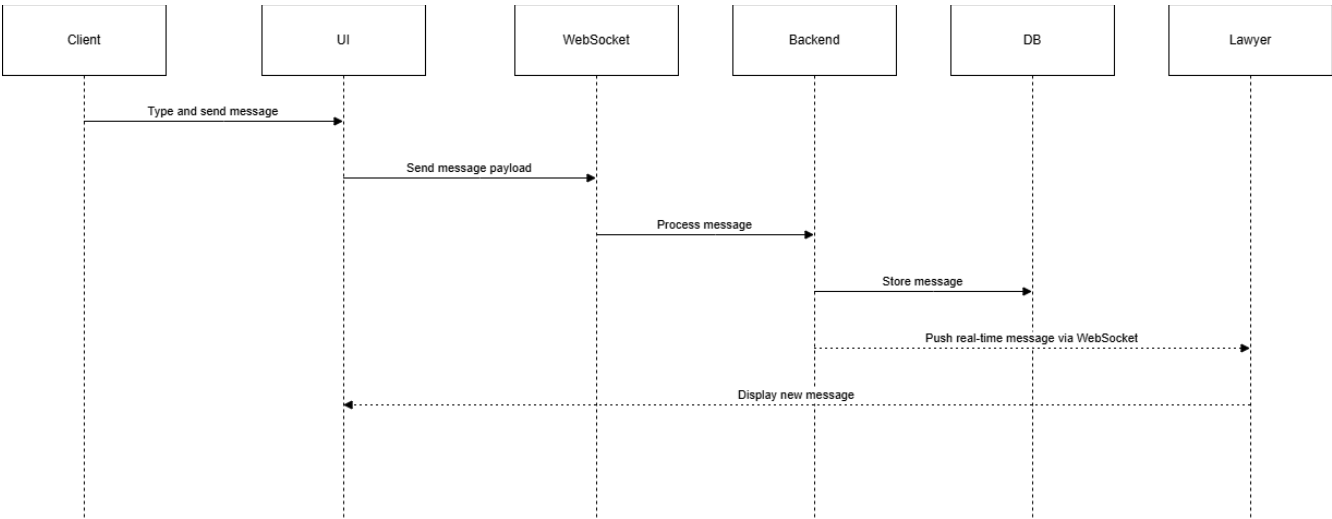
Component Name	Description and Purpose	Component Type / Format
Firebase Firestore	Stores messages, conversations, users collection	NoSQL database with subcollections
Firebase Security Rules	Enforces role-based access e.g., only client/lawyer in a conversation can read/write	Firestore Custom Rules (JSON)
Firebase Cloud Functions	For custom logic archiving, analytics, notifications	Node.js

- Object-Oriented Components
 - Class Diagram

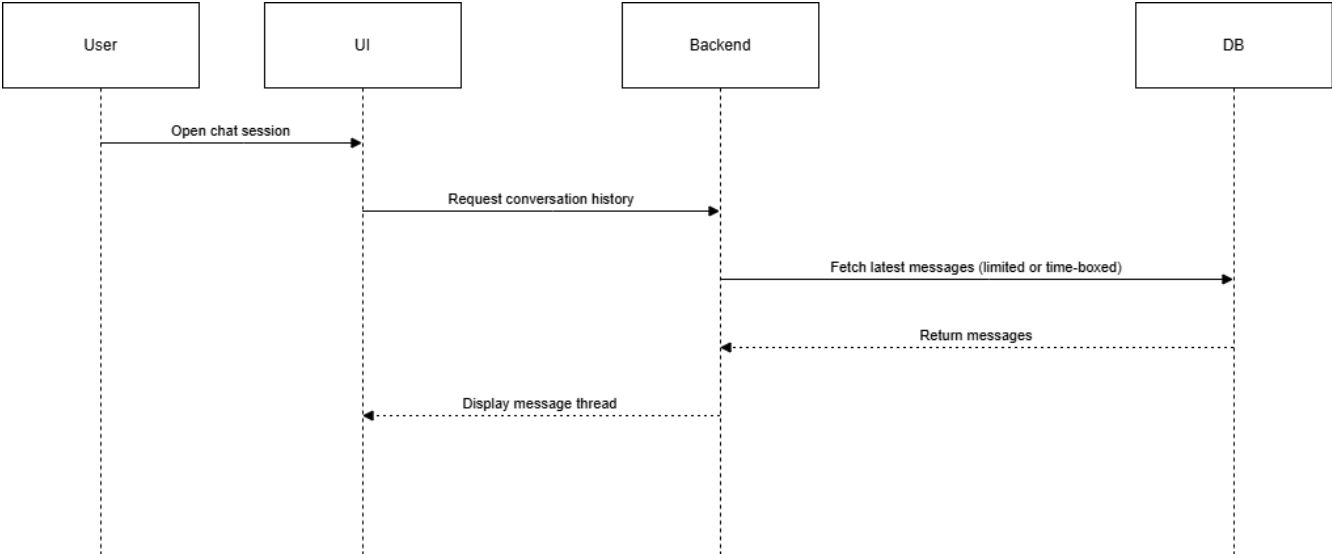


Sequence Diagram

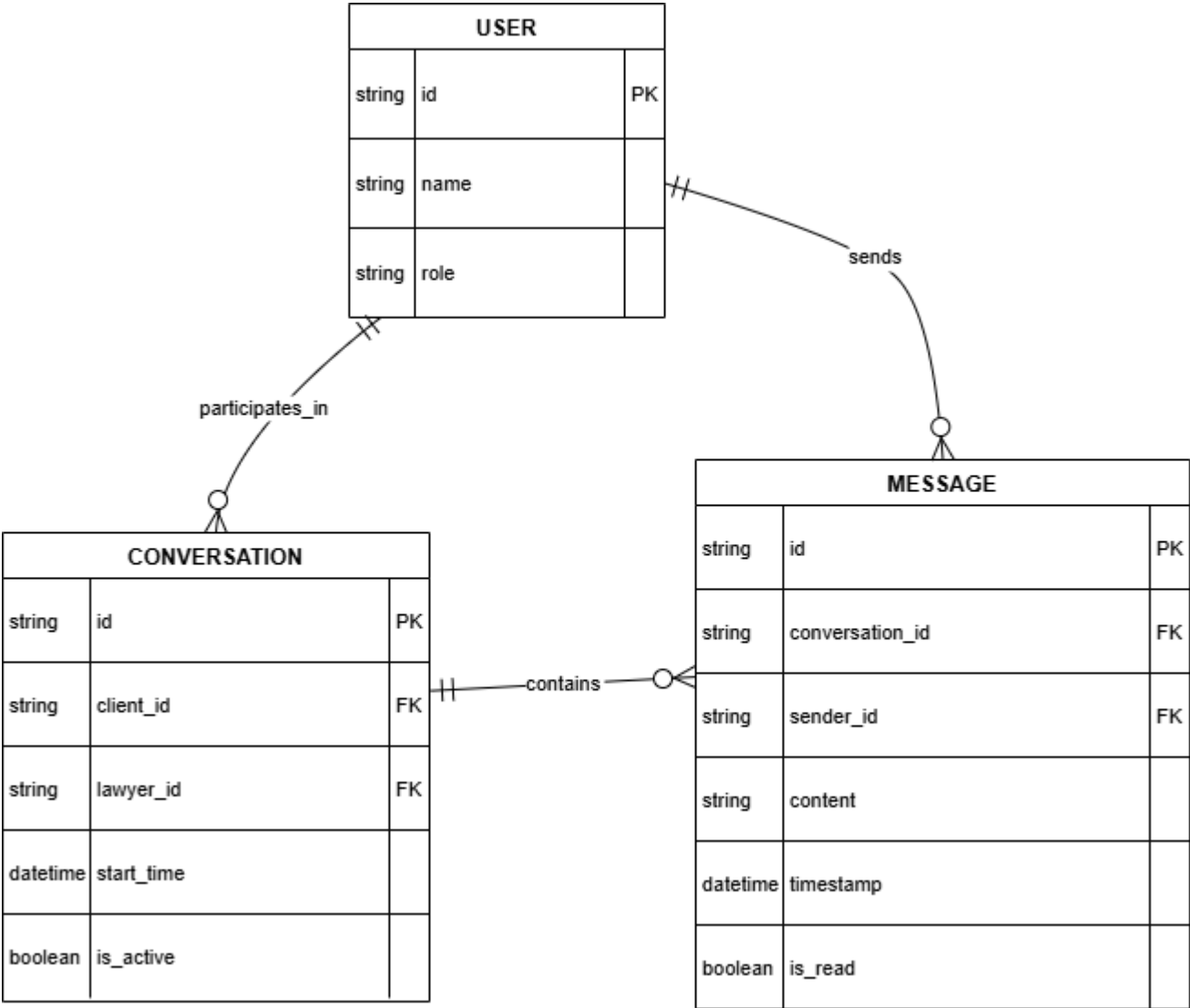
Transaction 6.1: Send and Receive Message



Transaction 6.2: Load History Message



- Data Design
 - ERD or schema



Module 7: Report and Analytics

- User Interface Design
- Front-end component(s)

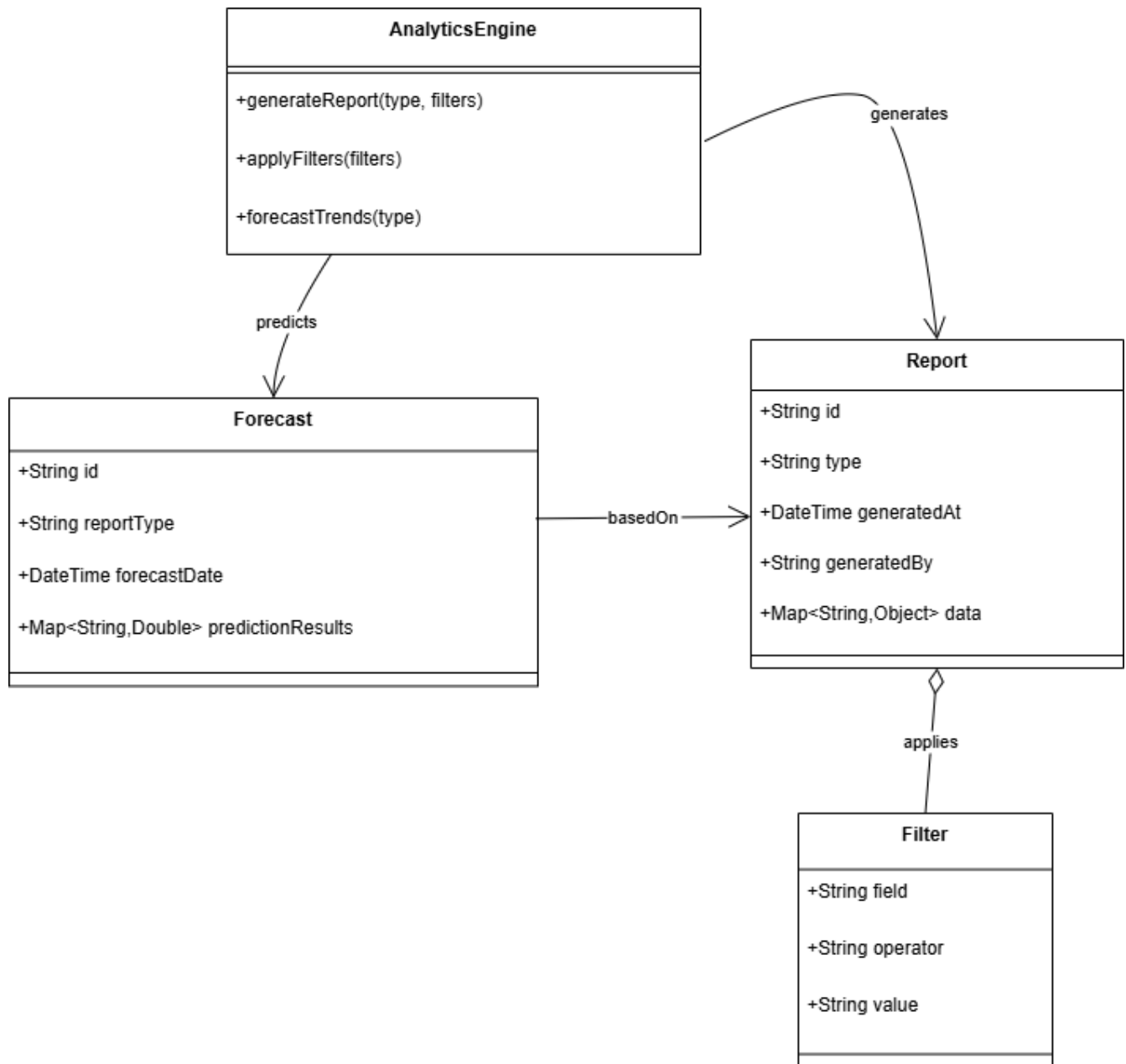
Component Name	Description and Purpose	Component Type / Format
Reports Dashboard	Displays metrics like active users, case trends, lawyer performance, appointment stats, etc.	Web Page / Dashboard (HTML, CSS, JavaScript)
Filter Panel	UI to apply filters such as date, user, role, location, etc.	Form + Filter Logic
Forecast Visualizer	Displays predicted trends (Line charts) based on the system data	Chart Component

- Back-end component(s)

Component Name	Description and Purpose	Component Type / Format
Reporting Service	Aggregates data from multiple modules (appointments, users, messages) and compiles reports.	Backend Service
Filter Processor	Applies filtering logic to existing report queries	Backend Filter
Forecast Engine	Uses historical data to generate trends using regression or ML model	Linear regression
Report Repository	Stores report generation logs and configurations.	DB table

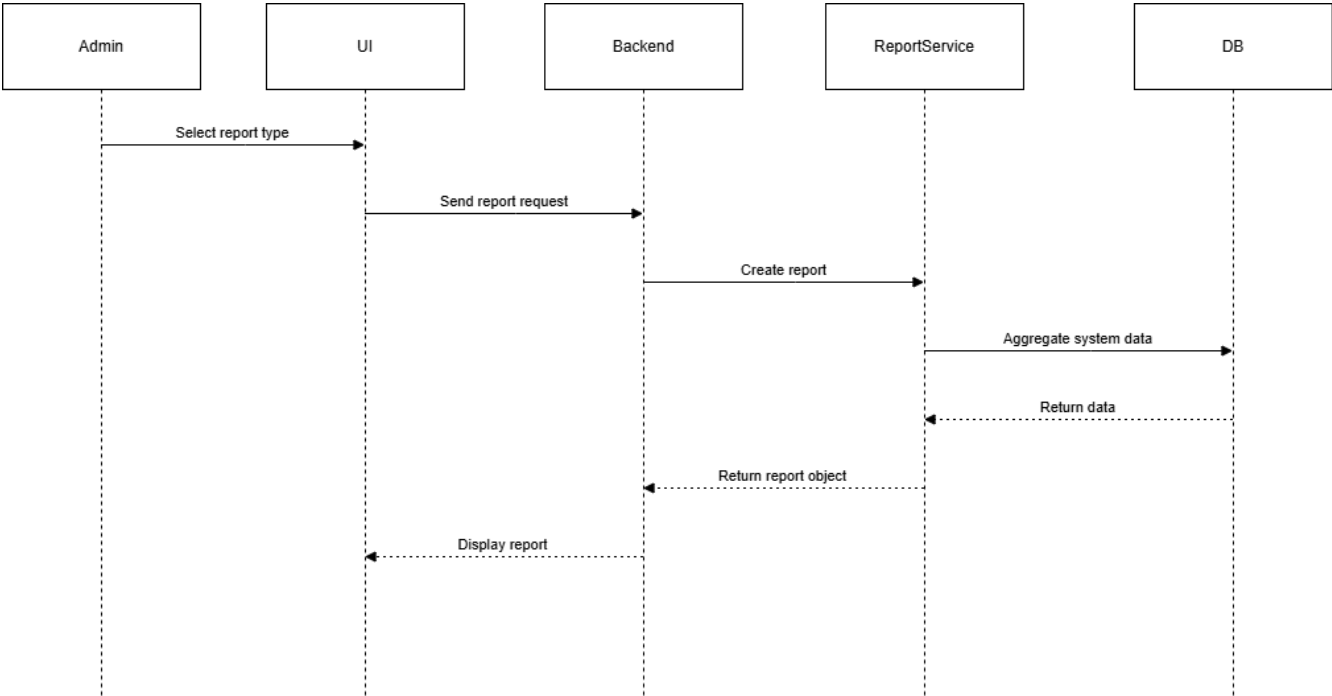
Object-Oriented Components

○ Class Diagram

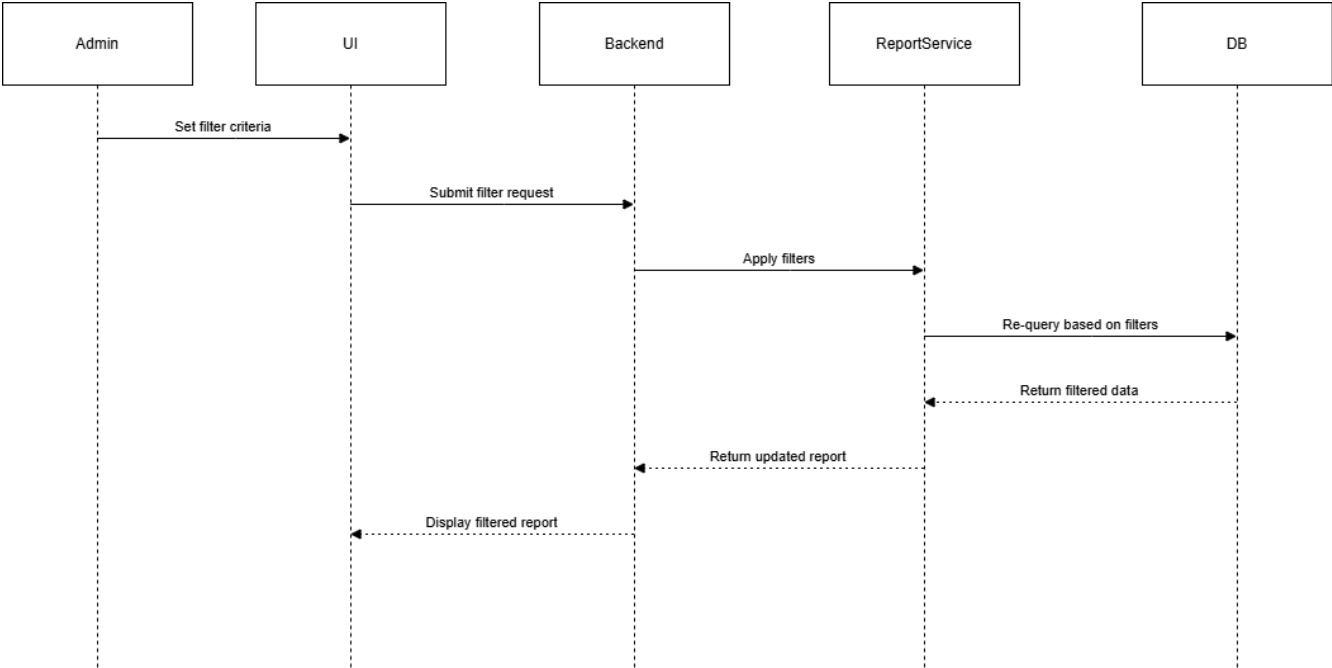


○ Sequence Diagram

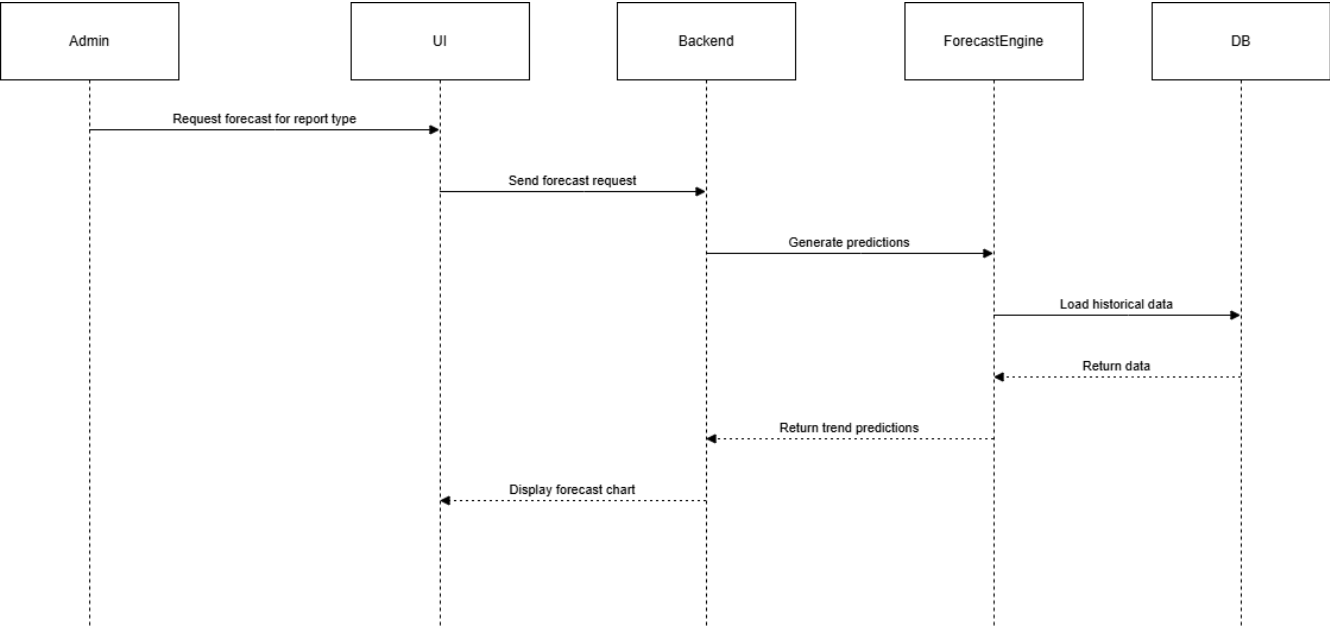
● Transaction 7.1: Generate Report



● Transaction 7.2: Apply Report Filters



Transaction 7.3: Forecast Trends



- Data Design
 - ERD or schema

