

IT 314

Group 25

# Gen-scholar:- A Collaborative Research Platform



PROF: DR. SAURABH TIWARI

MENTOR: Shyam Patel

# System design

## Domain Analysis

### 1)Entity Objects

#### **User**

Represents a system user with authentication details

**Attributes:** id: Integer, username: String, email: String, password: String, is\_active: Boolean, date\_joined: DateTime

#### **EmailOTP**

Email verification code handling

**Attributes:** id: Integer, email: String, otp\_code: String, is\_verified: Boolean, expires\_at: DateTime

#### **Workspace**

Central collaborative environment

**Attributes:** id: Integer, name: String, created\_by: User, created\_at: DateTime, index\_path: String, processing\_status: Enum {NONE, PROCESSING, READY, FAILED}

#### **WorkspaceMember**

Connects users to workspaces.

**Attributes:** id: Integer, role: Enum {RESEARCHER, REVIEWER}, joined\_at: DateTime

#### **WorkspaceInvitation**

Tracks pending invitations

**Attributes:** id: Integer, email: String, role: Enum {RESEARCHER, REVIEWER}, status: Enum {PENDING, ACCEPTED, DECLINED}, token: String

#### **Notification**

Alerts for system events

**Attributes:** id: Integer, type: Enum {SYSTEM, MENTION, INVITE}, is\_read: Boolean, created\_at: DateTime

#### **PinnedNote**

Shared workspace note

**Attributes:** id: Integer, content: Text, updated\_at: DateTime

#### **PDFFile**

Represents uploaded research documents

**Attributes:** id: Integer, title: String, file: Binary, uploaded\_at: DateTime, is\_indexed: Boolean, summary: Text, abstract: Text

### **Annotation**

Notes on a PDF page

**Attributes:** id: Integer, page\_number: Integer, coordinates: JSON {quads, type, color}, comment: Text, created\_at: DateTime

### **Thread**

Anchored discussion created from selected text

**Attributes:** id: Integer, page\_number: Integer, anchor\_rect: JSON, anchor\_side: String, selection\_text: Text, last\_activity\_at: DateTime

### **ThreadMessage**

Message inside a thread

**Attributes:** id: Integer, content: Text, created\_at: DateTime

### **ChatMessage**

Workspace-wide chat message

**Attributes:** id: Integer, content: Text, timestamp: DateTime

### **AIChatMessage**

Stores AI chat exchanges

**Attributes:** id: Integer, message: Text, is\_from\_bot: Boolean, timestamp: DateTime

## **2)Boundary Objects (Interfaces)**

**AuthenticationAPI**-Login, signup, OTP verification, session validation

**WorkspaceAPI**-Workspace Create, delete, membership, pinned notes, invitations

**PDFAPI**-Upload, download, view metadata, trigger background processing

**AnnotationAPI**-Create, list, delete annotations

**ThreadAPI**-Create anchored threads, post thread messages, list threads

**ChatAPI**-Workspace chat via HTTP fallback

**NotificationAPI**-List, mark-read, unread count

**ChatbotAPI**-AI question-answer interface

**InvitationAPI**-Send, accept, decline invitations

**WebSocketChatGateway**-Real-time workspace chat

**WebSocketThreadGateway**-Real-time thread events

**WebSocketNotificationGateway**-Real-time notification delivery

**ReactPDFViewer**-Frontend PDF rendering, selection capture, annotation display

**EmailService**-Send OTPs, invitations, password reset emails

### 3)Control Objects (Coordinators)

**AuthenticationController**-Handles login, signup, OTP, session flows

**SessionManager**-Validates and manages user sessions

**WorkspaceManager**-Role-based workspace validation and operations

**InvitationController**-Handles invitation lifecycle

**PDFProcessingCoordinator**-Manages PDF extraction, chunking, embedding, indexing

**AnnotationCoordinator**-Validates coordinates and annotation lifecycle

**ThreadCoordinator**-Creates threads, manages messages, triggers notifications

**ChatCoordinator**-Handles workspace chat logic and real-time broadcasts

**NotificationCoordinator**-Creates and routes notifications

**WebSocket Gateways**-Manage socket auth, group subscription, event broadcasting

**ChatbotQueryCoordinator**-RAG pipeline and LLM orchestration

**MentionDetector**-Finds @mentions and triggers notifications

## Subsystem Design

**Subsystem 1: Authentication & User Management**-Handles registration, login, OTP verification, sessions, and password resets.

- **Constituent Objects:** User, EmailOTP, SessionManager, AuthenticationController
- **Key Services:** signup, verifyOTP, login, logout, password reset, getCurrentUser
- **Interfaces:** requestEmailVerification, verifyOTP, signup, login, logout, requestPasswordReset, confirmPasswordReset, getCurrentUser

**Subsystem 2: Workspace Management**-Manages workspaces, membership, invitations, pinned notes, and notification generation logic

- **Constituent Objects:** Workspace, WorkspaceManager, InvitationController, PinnedNote, WorkspaceInvitation, NotificationCoordinator
- **Key Services:** createWorkspace, deleteWorkspace, inviteUser, acceptInvitation, listWorkspaces, updatePinnedNote, createNotification
- **Interfaces:** createWorkspace, deleteWorkspace, listWorkspaces, getWorkspaceMembers, inviteUser, acceptInvitation, declineInvitation, updateMemberRole, getPinnedNote, updatePinnedNote

**Subsystem 3: PDF Document Management**-handles PDF upload, storage, retrieval, metadata, and triggers background processing.

- **Constituent Objects:** PDFFile, PDFProcessingCoordinator
- **Key Services:** uploadPDF, listPDFs, deletePDF, getPDFFile, checkProcessingStatus
- **Interfaces:** uploadPDF, listPDFs, getPDFFile, getPDFMetadata, deletePDF, getPDFProcessingStatus

**Subsystem 4: PDF Annotation System**-Manages annotations and normalised coordinates.

- **Constituent Objects:** Annotation, AnnotationCoordinator

- **Key Services:** createAnnotation, deleteAnnotation, listAnnotations
- **Interfaces:** createAnnotation, listAnnotations, deleteAnnotation, and getAnnotationsByPage

**Subsystem 5: Threaded Discussions**-Anchored discussions tied to PDF selections, with real-time updates.

- **Constituent Objects:** Thread, ThreadMessage, ThreadCoordinator
- **Key Services:** createThread, postThreadMessage, getThreads, getThreadMessages, broadcastThreadEvent
- **Interfaces:** createThread, postThreadMessage, getThreads, getThreadMessages, broadcastThreadEvent

**Subsystem 6: Workspace Chat**-Real-time workspace chat with mention detection.

- **Constituent Objects:** ChatMessage, ChatCoordinator
- **Key Services:** postChatMessage, fetchHistory, broadcast
- **Interfaces:** postChatMessage, getChatMessages, broadcastChatMessage, detectMentions

**Subsystem 7: AI Chatbot**-Workspace-specific AI assistance with RAG + LLM.

- **Constituent Objects:** AIChatMessage, ChatbotQueryController
- **Key Services:** askQuestion, getChatHistory, indexWorkspacePDFs
- **Interfaces:** askQuestion, getChatHistory, indexWorkspacePDFs, getIndexStatus

**Subsystem 8: Real-Time Communication**-WebSocket management for chat, threads, and notifications.

- **Constituent Objects:** WebSocketGateway (Chat), WebSocketThreadGateway, WebSocketNotificationGateway
- **Key Services:** connectChatWebSocket, connectThreadWebSocket, connectNotificationWebSocket, broadcastToGroup, disconnectUser
- **Interfaces:** connectChatWebSocket, connectThreadWebSocket, connectNotificationWebSocket, broadcastToGroup, disconnectUser

# Interface Design

- **User authentication - WebSocket join:** Must join WebSocket since only a User object can authenticate (request validation on which workspaces can be accessed).
- **Workspace access - PDF upload:** System knows to which workspace it belongs due to the workspace ID. It will prompt me to send a PDF because it receives it through the background.
- **PDF upload - Annotation creation:** Annotations are created from the PDF and pages. Normalisation occurs through AnnotationCoordinator.
- **Annotation creation - Threaded discussion:** Threads/annotations will exist in a normalised space based upon coordinates.
- **Threaded discussion - WebSocket broadcasting:** Posts made on the thread will be sent through the WebSocketThreadGateway.
- **Workspace chat - WebSocket broadcasting:** Messages in the chat are sent for every member of this workspace.
- **Chat and threaded discussion notifications - Workspace management:** Notifications from @mentions will be received by NotificationCoordinator.
- **Workspace management - Notifications creation:** Notifs must be seen by admins for creations, invitations and other actions.
- **Workspace management - AI Assistant inquiry:** AI can only be asked about if a role is authenticated/permission strings validated.
- **Authentication - WebSocket join:** Clients must all authenticate to SessionManager to validate they're allowed to join the WebSocket on this end.

# Package Diagram :-

