**University Group Finder - API Design & Plan**

**Project Overview**

A Spring Boot REST API that enables university students to find group members for their projects, with a Next.js frontend.

**Core Entities**

**1. User (Student)**

"**id**": "int",

"**email**": "String",

"**username**": "String",

"**password**": "String",

"**firstName**": "String",

"**lastName**": "String",

"**specialization**": "String",

"**year**": "Integer (1-4)",

"**Semester**": "Integer (1-2)",

"**bio**": "String (optional)",

"**skills**": "List<String>",

"**gitHub**": " String",

"**linkedin**": " String",

"**profilePicture**": "String (URL)",

"**createdAt**": "LocalDateTime",

"**updatedAt**": "LocalDateTime"

**2. Project**

"**id**": "int",

"**title**": "String",

"**description**": "String",

"**moduleCode**": "String",

"**moduleName**": "String",

"**createdBy**": "int (User ID)",

"**maxMembers**": "Integer",

"**currentMemberCount**": "Integer",

"**requiredSkills**": "List<String>",

"**deadline**": "LocalDate",

"**status**": "Enum (OPEN, IN\_PROGRESS, COMPLETED, CANCELLED)",

"**createdAt**": "LocalDateTime",

"**updatedAt**": "LocalDateTime"

**3. ProjectMember**

"**id**": "int",

"**projectId**": "int",

"**userId**": "int",

"**role**": "Enum (Leader, MEMBER)",

"**status**": "Enum (ACTIVE, LEFT)",

"**joinedAt**": "LocalDateTime"

**4. JoinRequest**

"**id**": "int",

"**projectId**": "int",

"**userId**": "int",

"**message**": "String (optional)",

"**status**": "Enum (PENDING, ACCEPTED, REJECTED)",

"**createdAt**": "LocalDateTime",

"**respondedAt**": "LocalDateTime (optional)"

**API Endpoints**

**Authentication & User Management**

**POST /api/auth/register DONE**

* **Purpose**: Register new student
* **Body**: { email, password, firstName, lastName, studentId, major, year }
* **Response**: { token, user }
* **Validation**: Email must be university domain

**POST /api/auth/login DONE**

* **Purpose**: Student login
* **Body**: { email, password }
* **Response**: { token, user }

**PUT /api/users/profile DONE**

* **Purpose**: Update user profile
* **Headers**: Authorization: Bearer <token>
* **Body**: { firstName, lastName, bio, skills, major, year}
* **Response**: Updated user object

**Project Management**

**GET /api/projects DONE**

* **Purpose**: Get all open projects with filtering
* **Query Params**:
  + courseCode (optional)
  + skills (optional, comma-separated)
  + major (optional)
  + page (default: 0)
  + size (default: 10)
* **Response**: Paginated list of projects with member info

**POST /api/projects/create DONE**

* **Purpose**: Create new project
* **Headers**: Authorization: Bearer <token>
* **Body**: { title, description, courseCode, courseName, maxMembers, requiredSkills, deadline }
* **Response**: Created project object

**GET /api/projects/{id} DONE**

* **Purpose**: Get project details with members
* **Response**: Project with member list and join requests (if owner)

**PUT /api/projects/{id}**

* **Purpose**: Update project (owner only)
* **Headers**: Authorization: Bearer <token>
* **Body**: Project update fields
* **Response**: Updated project

**DELETE /api/projects/{id}/user/{uid} DONE**

* **Purpose**: Delete project (owner only)
* **Headers**: Authorization: Bearer <token>
* **Response**: Success message

**GET /api/projects/my**

* **Purpose**: Get user's projects (created + joined)
* **Headers**: Authorization: Bearer <token>
* **Response**: List of user's projects with role info

**Group Membership**

**POST /api/projects/{id}/join**

* **Purpose**: Request to join project
* **Headers**: Authorization: Bearer <token>
* **Body**: { message } (optional)
* **Response**: Join request object

**GET /api/projects/{id}/requests**

* **Purpose**: Get join requests for project (owner only)
* **Headers**: Authorization: Bearer <token>
* **Response**: List of pending join requests with user info

**PUT /api/projects/{id}/requests/{requestId}**

* **Purpose**: Accept/reject join request (owner only)
* **Headers**: Authorization: Bearer <token>
* **Body**: { status: "ACCEPTED" | "REJECTED" }
* **Response**: Updated request status

**DELETE /api/projects/{id}/leave**

* **Purpose**: Leave project
* **Headers**: Authorization: Bearer <token>
* **Response**: Success message

**DELETE /api/projects/{id}/members/{userId}**

* **Purpose**: Remove member from project (owner only)
* **Headers**: Authorization: Bearer <token>
* **Response**: Success message

**Search & Discovery**

**GET /api/search/students**

* **Purpose**: Search students by skills/major
* **Query Params**:
  + skills (comma-separated)
  + major
  + year
  + q (general search)
* **Response**: List of matching students (public info only)