SW Engineering CSC648/848 Section 01

EzJobs

Team 01

Team members

Name	Role
Rishita Meharishi	Team Leader / Front end
Luai Almaznai	Frontend Lead
L Chow	Backend Lead
Zaw Win Tun	Scrum Master
Yee Yang	Github Master

Milestone 2

10/07/2024

History Table

Revision #	Date	Description
1	10/07/2024	Initial document for milestone 2

1. Data Definitions V2

Primary Data Name	Definition	Usage	Sub-Data
users	Data to store user information. Represents a person using the job kanban app, including credentials and email.	Holds the core details about users including login credentials and identification. Manages user login, registration, and email addresses that are used to reset passwords.	• `id`: Unique identifier for users. • `username`: User's display name. • `email`: User's email address (unique). • `password`: Hashed password for authentication.
job_status	The current stage or progress of the job application in the recruitment process.	Tracks the different stages a job application goes through.	• `interested`: The user shows interest in the job. • `pending`: Awaiting response or decision. • `Progress`: The application is being processed. • `offer`: The job offer has been extended. • `declined`: The job offer has been declined. • `ghosted`: No further communication from the employer. • `follow_up`: Following up on the application status.
jobs	Information related to a specific job application, including details about the job and its status.	Tracks user job applications, status, and related metadata.	• `id`: unique job application ID • `user_id`: Reference to the user who applied. • `column_id`: Status column in the Kanban board, which tracks the current application phase. • `company`: Name of the hiring company. • `position`: Title of the job applied for. • `salary`: Salary information for the position. • `type`: Nature of the work (remote, hybrid, onsite). • `location`: Physical location of the job. • `link`: Link to the job posting. • `description`: Details about job roles and responsibilities. • `user_note`: Personal notes or reminders. • `date_applied`: Date the application was submitted. • `date_scheduled`: Date of scheduled interviews or meetings.
columns	Kanban board columns that track job application status.	Organizes job statuses for the Kanban board.	• `id`: Unique column ID. • `name`: Status from job_status above. • `color`: The hex color code for the column. • `position`: Column position/order in the kanban board.
contacts	This data stores relevant information about professional contacts such as recruiters,	It helps users keep track of people related to job applications for	• `id`: Unique contact ID. • `user_id`: Reference to the user who added the contact. • `user_notes`: Additional details about the contact. • `name`: Name of the contact person.

Primary Data Name	Definition	Usage	Sub-Data
	hiring managers, or networking connections.	networking or follow-up purposes.	• position: Job title of the contact. • company: Company associated with the contact. • email: Contact's email address. • phone_num: Contact's phone number.
docs	PDF format with a max size of 5MB. This data stores user-uploaded or AI-generated files relevant to job applications, such as resumes, cover letters, and other supporting documents.	It allows users to upload, manage, and retrieve documents they need when applying for jobs or preparing for interviews.	• id: Unique document ID. • user_id: Reference to the document owner. • title: Document title. • file-path: Location of the document.
ai_resume	Inputs for generating a resume based on AI, capturing user details, skills, and experience.	Generates tailored resumes for job applications based on user input.	• id: Unique resume builder ID. • user_id: Reference to the owner. • personal_information: User's name, contact, and profile details. • skills: Relevant skills for the resume. • education: Academic background and qualifications. • experience: Professional work history. • projects: Details of projects completed.
ai_interview_prep	AI-generated interview preparation based on the job position, description, and company research provided by the user.	Helps users prepare for interviews by generating custom questions and example responses.	• id: Unique interview prep ID. • user_id: Reference to the user. • company: Research on the company. • job_position: Position for which interview prep is required. • job_description: Details of job responsibilities. • ai_response: AI-generated interview suggestions.

2. Functional Requirements V2

NOTE:
***Requirement IDs are the numbers before the functional requirements (eg. Sign Up has requirement ID 1)**
***Priority Scale is from 1 to 3 (High, Medium, Low)**
***User stories to be referenced will be from IDs 1-13 to reflect their respective user story.**
***Assuming multiple user stories can be referenced for each requirement.**

Account

Sign-Up

1. The user should be able to sign up.
   Description:
   Just a simple sign-up way for users new to the website/app. Sign-up is mandatory to start using our website/app.

   Priority: 1

   User Story To Be Referenced: All 13 since you need an account to do what is listed.

Login

2. The user should be able to log in using email/username and password **Description: Another simple way for users to log in using their email/username and password.**
   Has a "Forgot Password" option that users can use to recover their forgotten password through the use of their email.
   Possibly with text too in the future if time permits.

   Priority: 1

   User Story To Be Referenced: All 13 since you need an account to do what is listed.

View Profile

3. The user should be able to both view and update their profile with basic information such as name, email, and address.
   Description:
   This allows users to view their profile and lets them edit and save their basic information, as well as being able to add or change their profile picture.

   Priority: 2

   User Story To Be Referenced: No specific user story to reference to, but still would want it.
   —--

Job Functionalities

Add Jobs

4. The user should be able to add job(s) that ask for specific important information such as the job's name, the job's position, etc.
   Description:
   A feature that allows users to add jobs to their job screen overview with specific fields asked beforehand, such as the job's name or the job's position.

   Priority: 1

   User Story To Be Referenced: 1, 9 (Both are more about tracking jobs, but you need to be able to add jobs to the track, no?)

Job Movement

5. The user should be able to drag their job(s) around to several categories that are broken down individually
   as their columns, as follows: Interested, Offer, Pending, In-Progres, Accepted, Declined, Ghosted, and Follow-Up.
   Description:
   A feature that allows users to drag and drop their jobs around the overview screen into different columns that represent different statuses of the jobs.

   Priority: 1

   User Story To Be Referenced: 1, 9 (helps with tracking), 11 (save jobs for later)

Job Sort

6. The user should be able to sort through jobs with either a drop-down menu that offers several choices. **(eg. Position, Location, etc)**
   or through their own specified manual filter. **(eg. User types a job name)**
   Description:
   A feature that allows users to either search through a filter to find specific jobs relevant to them or through a dropdown menu with a selector for several choices like by Company, by Location, etc.

Job Notifications

7. The user should be able to view updates to any job-related functionalities via notifications and have some control over what sort of notifications. **(eg. notifications for when they get job offers, when jobs they applied for get either accepted or rejected and what notifications they would like on and off)**
   Description:
   This allows users to keep track of various notifications regarding their job statuses or possible messages from companies/recruiters.

Priority: 3

User Story To Be Referenced: 1, 3, 6, 13

—---

AI

AI Help

8. The user should be able to utilize AI to help with job search.
   Description:
   A feature that uses AI to help users out with their job search through various recommendations/tips given by AI with job-related functionalities and/or job searches in general.

   Priority: 1

   User Story To Be Referenced: 4, 5, 7, 8

—---

Functional Tabs

Portfolio

9. The site should have a Portfolio tab.
   Description:
   This tab allows users to upload one's resumes/cover letters to be able to use for their job search.

   Priority: 1

   User Story To Be Referenced: 5 (related to resume/cover letter, but 5 is more about guidance on them, so not sure if it fits here)

Connections

10. The site should have a Connections tab.
    Description:
    This tab allows users to keep track of companies that the have applied for and/or are interested in.

    Priority: 1

    User Story To Be Referenced: 1, 9 (helps with tracking)

Task Alerts

11. The site should have a Task Alerts tab.
    Description:
    This tab allows users to add reminders for important dates, times, and their own personal notes.

    Priority: 2

    User Story To Be Referenced: 3, 6

Analytics

12. The site should have an Analytics tab.
    Description:
    This tab allows users to get a summary of their job search progress. **(eg. graphs, charts, etc)**

    Priority: 1

    User Story To Be Referenced: 1, 9 (A summary of job search progress has to do with tracking, ya? Else, unsure what to put here)

Settings

13. The site should have a Settings tab.
    Description:
    This tab allows users to make basic and simple site adjustments based on their preferences.

    Priority: 2

    User Story To Be Referenced: No specific user story about this here, but we would still want one.

—---

User Interaction

User Support

14. The site should have a place to get support and/or ask for help.
    Description:
    This allows users to get information for support/help, such as displaying our names, an email (personal or company email), and possibly a phone number.

    Priority: 3

    User Story To Be Referenced: No specific user story about this here, but we would probably still want one.

FAQ

15. The site should have a FAQ that answers common questions.
    Description:
    This allows users to have a sort of reference page where common questions are answered.

    Priority: 3

    User Story To Be Referenced: No specific user story about this here, but we would probably still want one.

User Feedback

16. The site should have a place where users can give feedback.
    Description:
    This allows users to provide feedback on the website and/or any of the feature(s).

    Priority: 3

    User Story To Be Referenced: No specific user story about this here, but we would probably still want one.

—--

User Experience Optimization

LinkedIn Integration

17. The site should have a LinkedIn integration of sorts for resumes and jobs.
    Description:
    This feature gives users a convenient way to connect to and stay connected to LinkedIn while being able to utilize some of LinkedIns' features, such as jobs on their site.

    Priority: 3

    User Story To Be Referenced: No specific user story about this here, but we would probably still want one.

Form Autofill

18. The site should have an autofill feature for job forms like what Simplify has.
    Description:
    This feature allows users to save time and tediousness by having the autofill fill out the forms for them, which lets them focus their attention elsewhere, such as looking for more jobs to apply.

    Priority: 3

    User Story To Be Referenced: 2, 12

Custom Message

19. The user should get appropriately curated message(s) for when the job(s) are moved from one column to another.
    (eg. When a job is moved from Interested to Offer, a message could pop up saying, “Good job on landing a job offer!” or when a job is moved from Pending to Declined, a message could pop up saying, “Aw, that's too bad. Don't give up and keep trying!”)
    Description:
    A feature that allows users to get their own curated or sort of customized message(s) for when they move job(s) around. Can be simple motivational phrases, general reactions to even tips/encouragements. These can be turned on/off in the Settings or perhaps customized to better meet user needs/wants.

    Priority: 3

    User Story To Be Referenced: No specific user story to reference, but think we want this feature still.

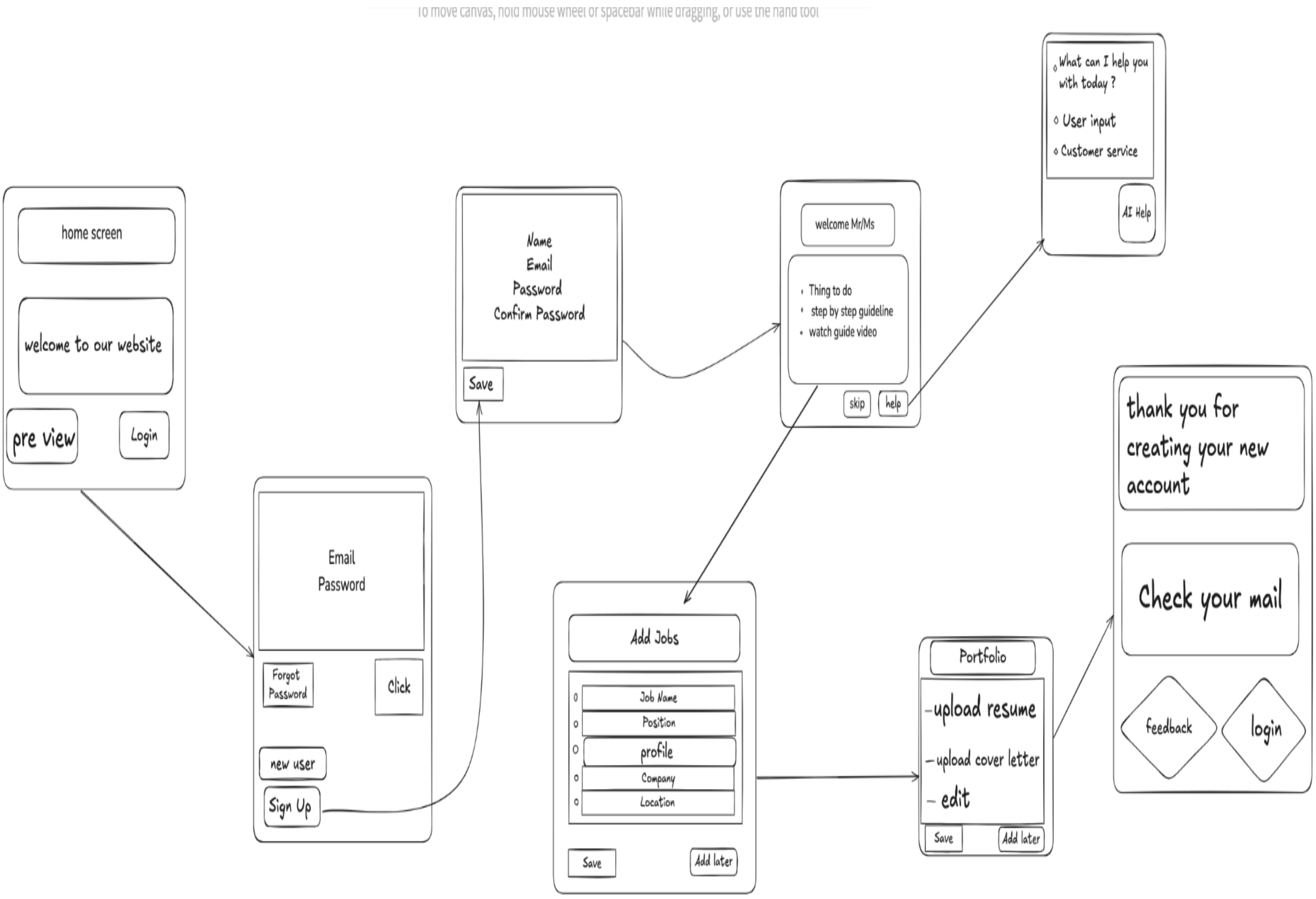
Mode Selector

20. The site should have a simple Dark/Light mode selector.
    Description:
    This allows users to quickly and easily switch between the two modes of Light/Dark to fit their preference.

    Priority: 2

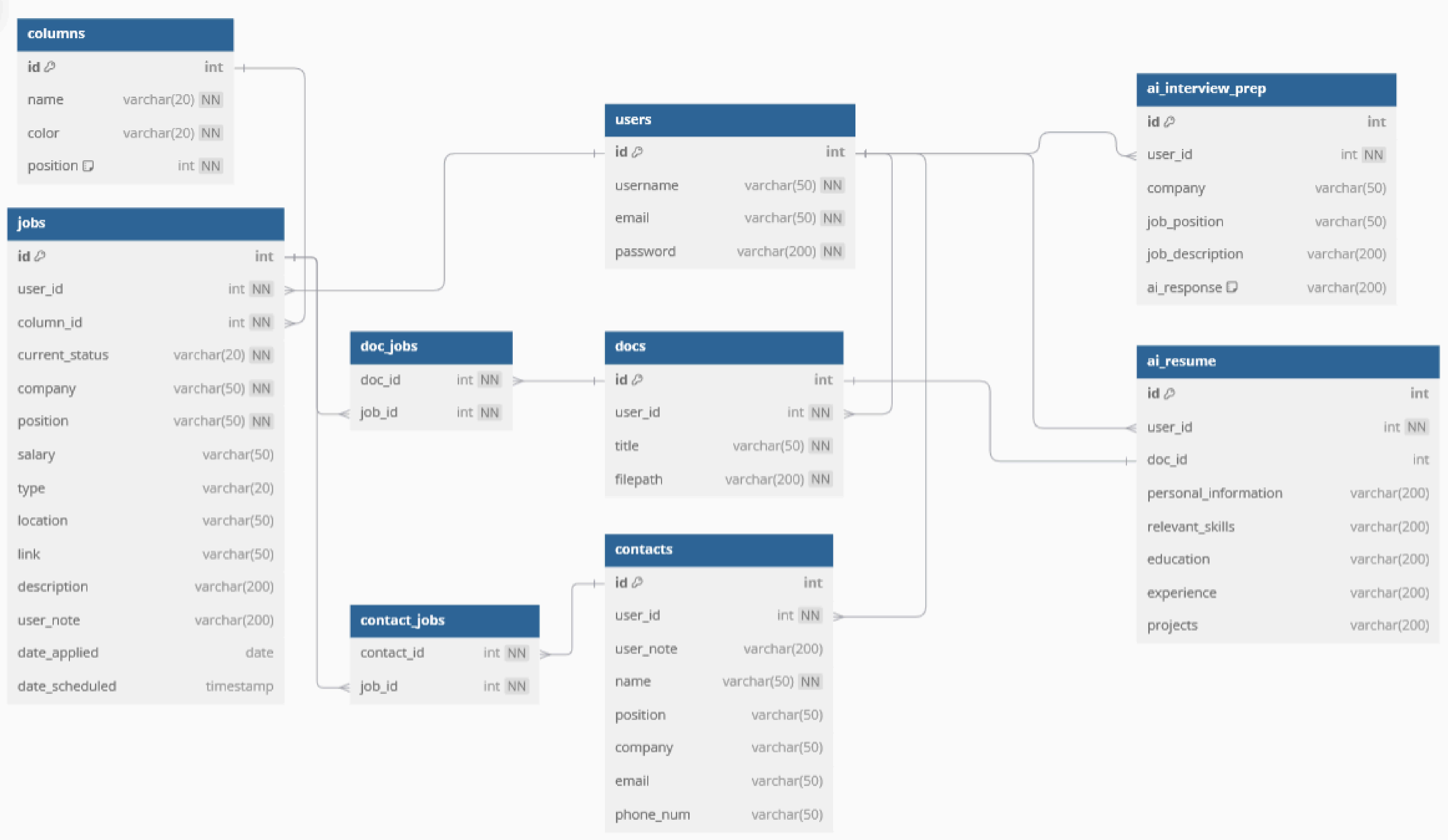
    User Story To Be Referenced: No specific user story about this here, but we would still want one.

3.UI Mockups and UX Flows



4.High-level Architecture, Database Organization

Database Organization



columns		
id	int	
name	varchar(20)	NN
color	varchar(20)	NN
position	int	NN

jobs		
id	int	
user_id	int	NN
column_id	int	NN
current_status	varchar(20)	NN
company	varchar(50)	NN
position	varchar(50)	NN
salary	varchar(50)	
type	varchar(20)	
location	varchar(50)	
link	varchar(50)	
description	varchar(200)	
user_note	varchar(200)	
date_applied	date	
date_scheduled	timestamp	

users		
id	int	
username	varchar(50)	NN
email	varchar(50)	NN
password	varchar(200)	NN

doc_jobs		
doc_id	int	NN
job_id	int	NN

docs		
id	int	
user_id	int	NN
title	varchar(50)	NN
filepath	varchar(200)	NN

contact_jobs		
contact_id	int	NN
job_id	int	NN

contacts		
id	int	
user_id	int	NN
user_note	varchar(200)	
name	varchar(50)	NN
position	varchar(50)	
company	varchar(50)	
email	varchar(50)	
phone_num	varchar(50)	

ai_interview_prep		
id	int	
user_id	int	NN
company	varchar(50)	
job_position	varchar(50)	
job_description	varchar(200)	
ai_response	varchar(200)	

ai_resume		
id	int	
user_id	int	NN
doc_id	int	
personal_information	varchar(200)	
relevant_skills	varchar(200)	
education	varchar(200)	
experience	varchar(200)	
projects	varchar(200)	

Add/Delete/Search Architecture

Table Name	Add	Search	Delete	Displayed
users	☑	☑	□	☑
jobs	☑	☑	☑	☑
columns	□	□	□	☑
docs	☑	☑	☑	☑
doc_jobs	☑	☑	☑	☑
contacts	☑	☑	☑	☑
contact_jobs	☑	☑	☑	☑
ai_resume	☑	☑	☑	☑
ai_interview_prep	☑	☑	☑	☑

Backend endpoint APIs

API Endpoint	HTTP Method	Description	HTTP Request Body Example	HTTP Response Example	Authentication Required?
/api/auth/register	POST	Register a new user	`{ "username": "abc", "email": "abc@gmail.com", "password": "password123" }`	`{ "message": "User registered successfully." }``	No

API Endpoint	HTTP Method	Description	HTTP Request Body Example	HTTP Response Example	Authentication Required?
/api/auth/login	POST	Log in an existing user	`{ "username": "ABC", "password": "password123" }`	{"message": "Login successful." }	No
/api/auth/logout	POST	Log out the current user	N/A	{ "message": "Logout successful." }	Yes
/api/jobs/add	POST	Add a new job application	{ "user_id": 1, "column_id": 2, "company": "XYZ Corp", "position": "Software Engineer", "salary": "90k", "type": "Full-time", "location": "Remote", "link": "https://xyz.com/job", "description": "Frontend Developer", "user_note": "Applied through LinkedIn", "date_applied": "2024-10-01" }	{ "id": 1, "message": "Job added successfully." }	Yes
/api/jobs/{id}	GET	Retrieve details of a specific job by ID	N/A	{ "id": 1, "company": "XYZ Corp", "position": "Software Engineer", "salary": "90k", "location": "Remote", "description": "Frontend Developer", "user_note": "Applied through LinkedIn", "date_applied": "2024-10-01", "column_id": 2 }	Yes
/api/jobs/search	GET	Search for jobs based on criteria (e.g., company, position, status)	?company=XYZ&position=Engineer	[{"id": 1, "company": "XYZ Corp", "position": "Software Engineer", "status": "in_progress"}]	Yes
/api/jobs/{id}	DELETE	Delete a job by ID	N/A	{ "message": "Job deleted successfully." }	Yes
/api/docs	POST	Add a new document (resume, cover letter, etc.)	{ "user_id": 1, "title": "Resume", "filepath": "/docs/resume.pdf" }	{ "id": 1, "message": "Document added successfully." }	Yes
/api/docs/{id}	GET	Retrieve a document by ID	N/A	{ "id": 1, "title": "Resume", "filepath": "/docs/resume.pdf" }	Yes
/api/docs/{id}	DELETE	Delete a document by ID	N/A	{ "message": "Document deleted successfully." }	Yes
/api/contacts	POST	Add a new contact	{ "user_id": 1, "name": "John Doe", "position": "HR Manager", "company": "XYZ Corp", "email": "john.doe@xyz.com", "phone_num": "123-456-7890", "user_note": "Met at job fair" }	{ "id": 1, "message": "Contact added successfully." }	Yes
/api/contacts/search	GET	Search for contacts by name, company, or position	?name=John&company=XYZ	[{"id": 1, "name": "John Doe", "company": "XYZ Corp", "position": "HR Manager"}]	Yes
/api/contacts/{id}	GET	Retrieve details of a specific contact by ID	N/A	{ "id": 1, "name": "John Doe", "company": "XYZ Corp", "position": "HR Manager", "email": "john.doe@xyz.com", "phone_num": "123-456-7890" }	Yes
/api/contacts/{id}	DELETE	Delete a contact by ID	N/A	{ "message": "Contact deleted successfully." }	Yes

API Endpoint	HTTP Method	Description	HTTP Request Body Example	HTTP Response Example	Authentication Required?
/api/ai_resume_builder	POST	Add a new AI-generated resume	`{ "user_id": 1, "doc_id": 1, "personal_information": "Yee Yang", "relevant_skills": "JavaScript, React", "education": "CS Degree", "experience": "2 years in frontend development", "projects": "Kanban board project" }`	`{ "id": 1, "message": "AI Resume generated successfully." }`	Yes
/api/ai_resume_builder/{id}	GET	Display a specific AI-generated resume	N/A	`{ "id": 1, "user_id": 1, "personal_information": "Yee Yang", "relevant_skills": "JavaScript, React", "education": "CS Degree", "experience": "2 years", "projects": "Kanban board project" }`	Yes
/api/ai_resume_builder/{id}	DELETE	Delete an AI-generated resume by ID	N/A	`{ "message": "Resume deleted successfully." }`	Yes
/api/ai_interview_prep	POST	Generate AI interview prep	`{ "user_id": 1, "company": "XYZ Corp", "job_position": "Software Engineer", "job_description": "Frontend Developer" }`	`{ "id": 1, "ai_response": "Expected questions and tips for XYZ Corp Software Engineer position." }`	Yes
/api/ai_interview_prep/{id}	GET	Retrieve AI interview prep details	N/A	`{ "id": 1, "company": "XYZ Corp", "job_position": "Software Engineer", "job_description": "Frontend Developer", "ai_response": "Expected questions and tips..." }`	Yes
/api/ai_interview_prep/{id}	DELETE	Delete AI interview prep by ID	N/A	`{ "message": "AI Interview Prep deleted successfully." }`	Yes

External API - OpenAI API

What is OpenAI API?

The OpenAI API provides developers with access to powerful AI models, including GPT (Generative Pre-trained Transformer), that can perform a variety of natural language processing tasks. These models are capable of understanding and generating human-like text, making them versatile tools for applications like content generation, summarization, language translation, question-answering, and much more.

How Our Project Utilizes OpenAI API

Overview

This project aims to leverage the capabilities of the OpenAI API to assist users in two primary areas:

- **Resume Building Assistance**: The API will summarize user input (e.g., work experience, skills) and format it into a structured resume template.
- **Interview Preparation**: The API will generate example interview questions and provide example responses for users to practice.

Use Case 1: Resume Template Summarization

In this feature, users will provide details about their work experience, education, and skills. The OpenAI API will be used to:

- **Summarize**: Summarize user-provided content into succinct points suitable for inclusion in a resume.
- **Formatting Assistance**: Ensure that the summarized content adheres to standard resume formatting and industry-specific language.

This will help users create clean, concise, and professionally worded resumes quickly and efficiently.

Use Case 2: Interview Question Generation

This feature focuses on preparing users for job interviews by:

- **Generating Custom Interview Questions**: Based on the user's role or industry, the OpenAI API will create relevant and challenging interview questions.
- **Providing Sample Responses**: The API will also generate well-structured example responses, offering insights on how to approach different types of interview questions (e.g., behavioral, technical).

This can greatly enhance the user's interview preparation by offering realistic questions and guiding them on how to respond effectively.

Benefits of Using OpenAI API

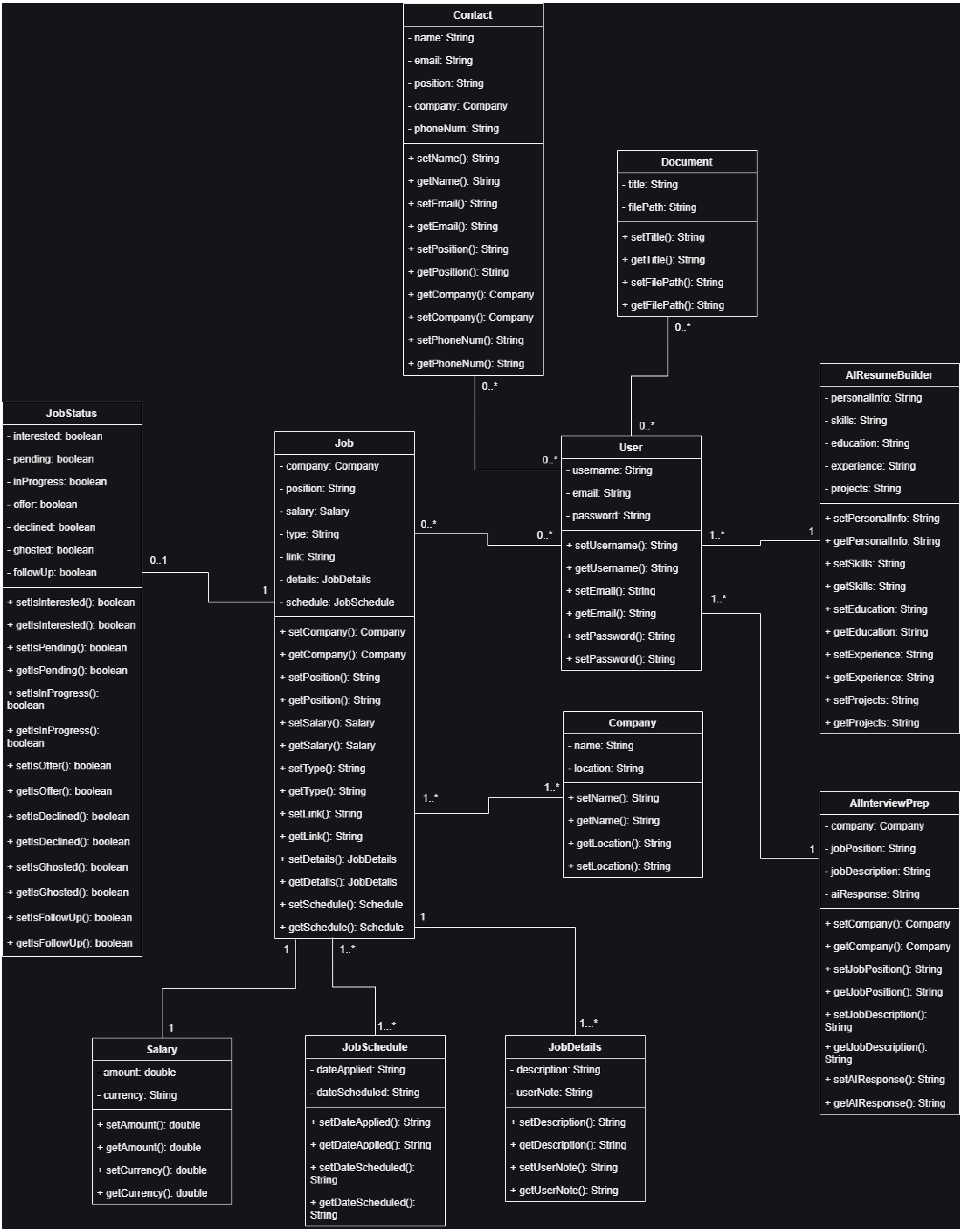
- **Efficiency**: The API can process large amounts of user data quickly, allowing for instant generation of summaries and interview questions.
- **Personalization**: By analyzing the user's input, the API can tailor its responses to specific industries, roles, and experiences.
- **Consistency**: OpenAI's models provide high-quality, professional output consistently, ensuring that the generated content meets user expectations.
- **Flexibility**: The API can be adjusted in terms of creativity, response length, and tone, depending on the user's preferences.

Conclusion

By integrating the OpenAI API, our project will enable users to enhance their resume-building and interview preparation processes. The power of AI-driven summarization and content generation will provide users with valuable tools to streamline these critical job-seeking activities.

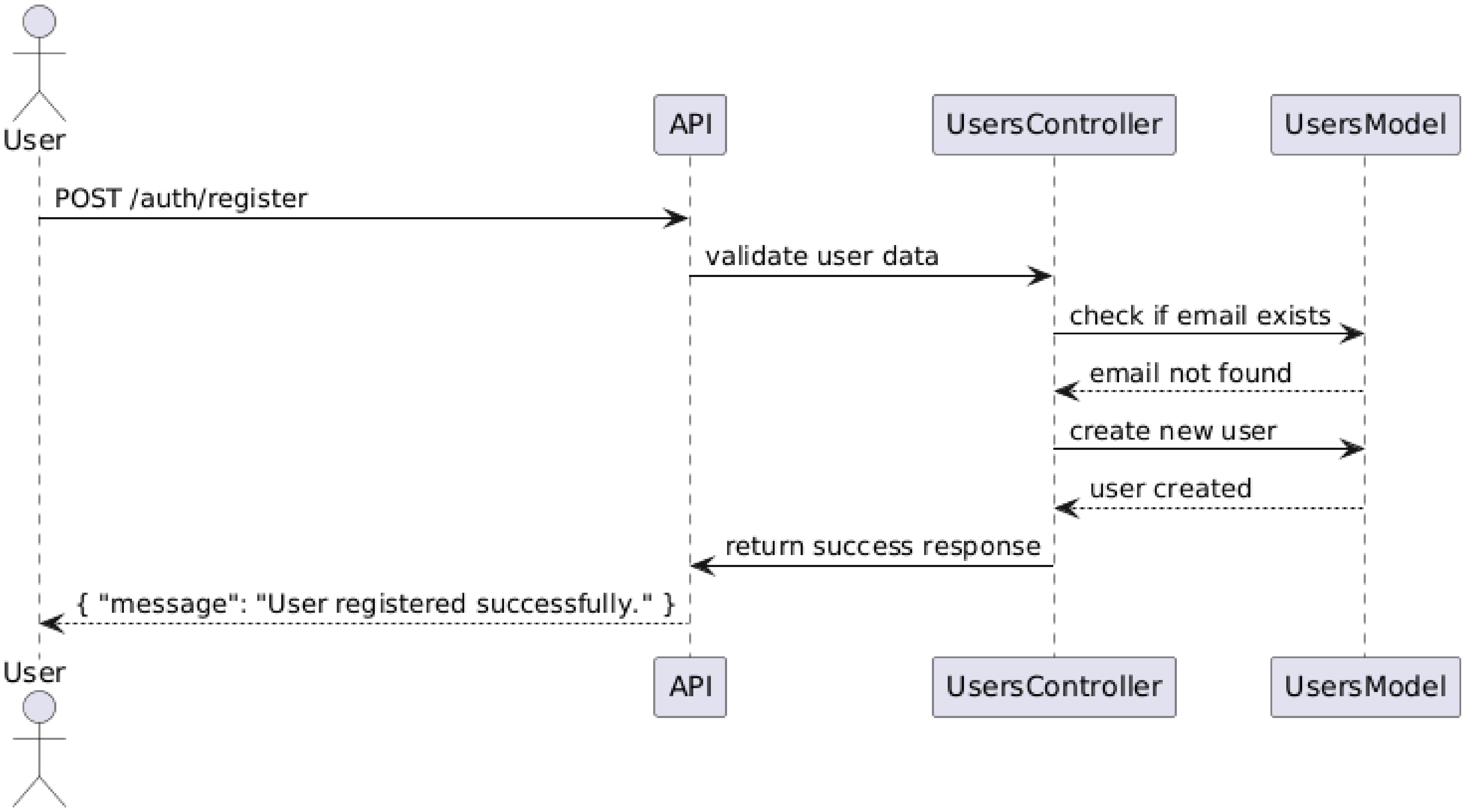
5.High Level UML Diagrams

Class Diagram

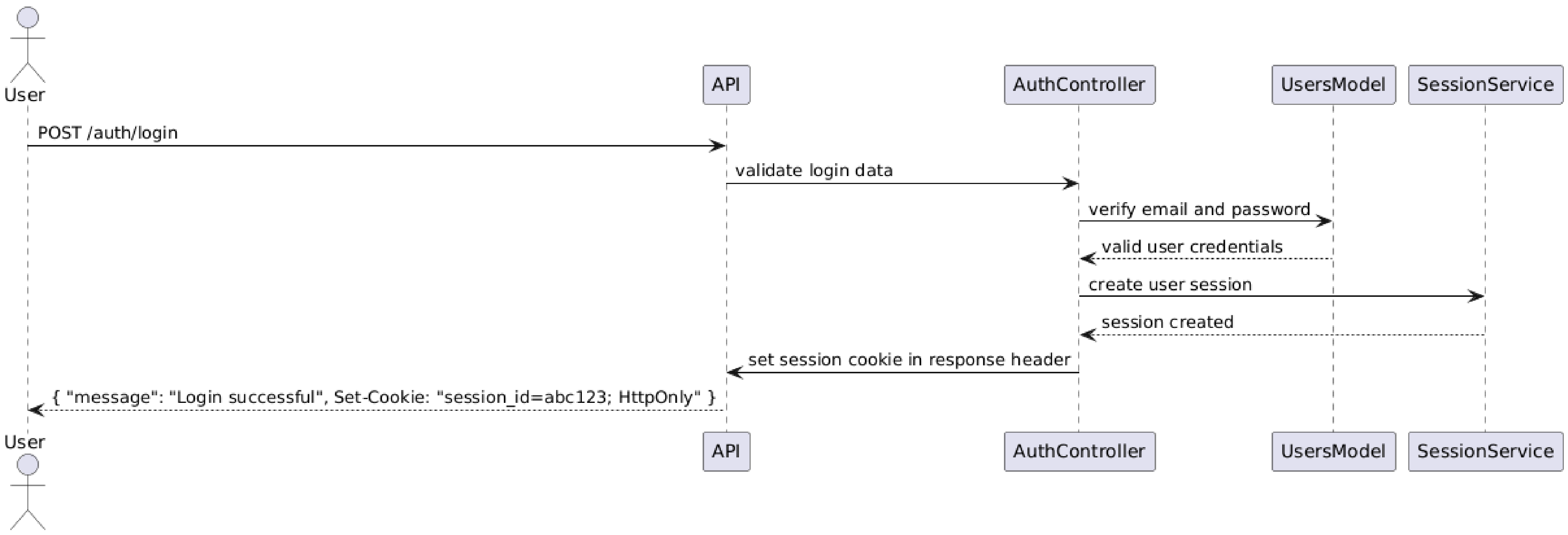


Sequence diagrams

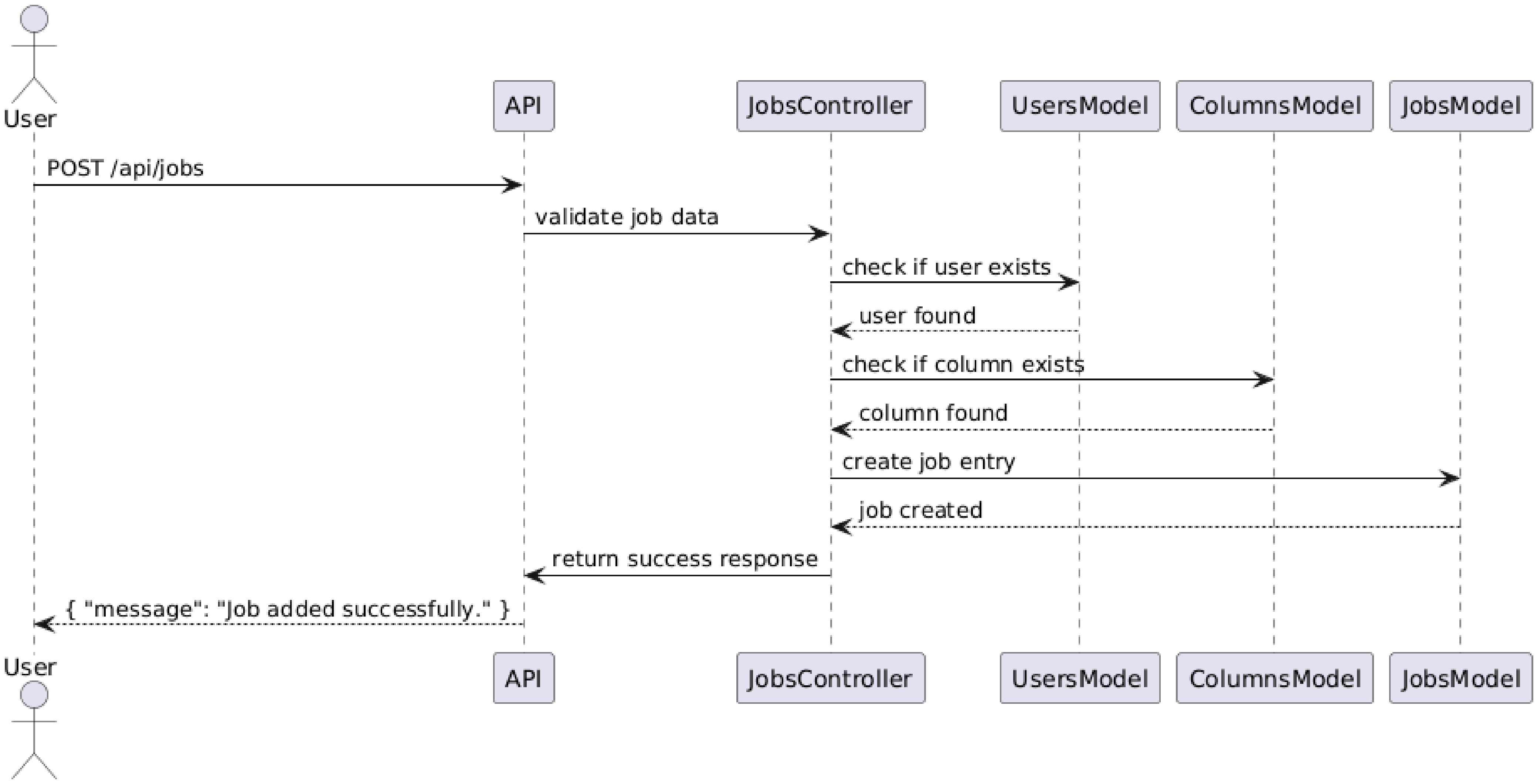
Register user



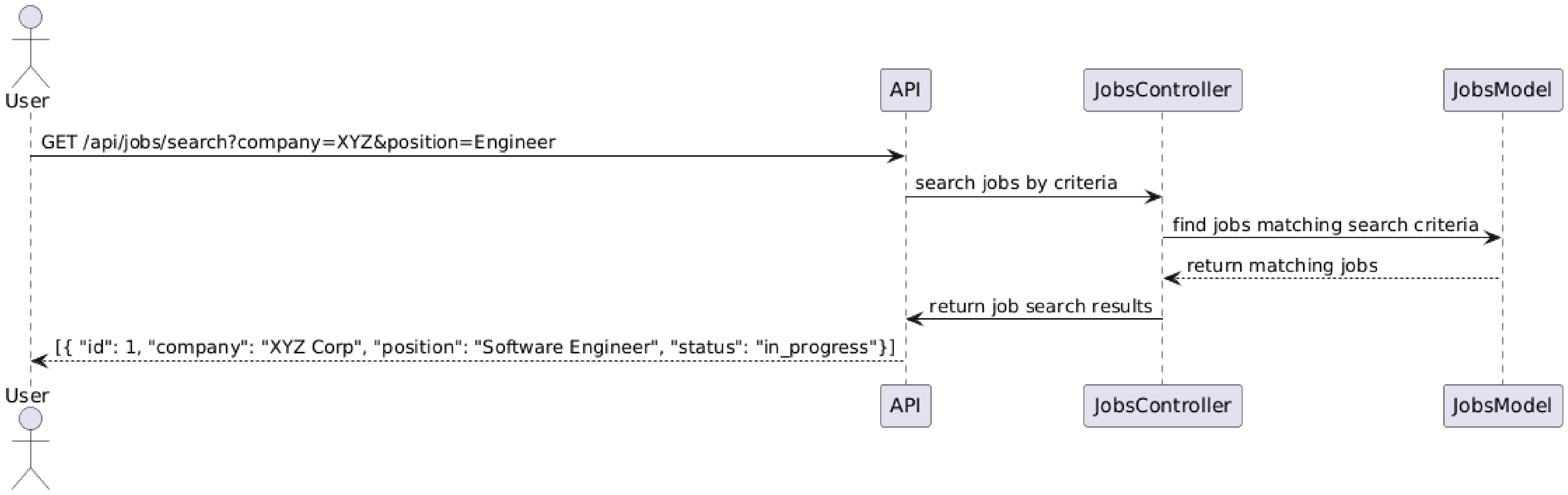
Login user



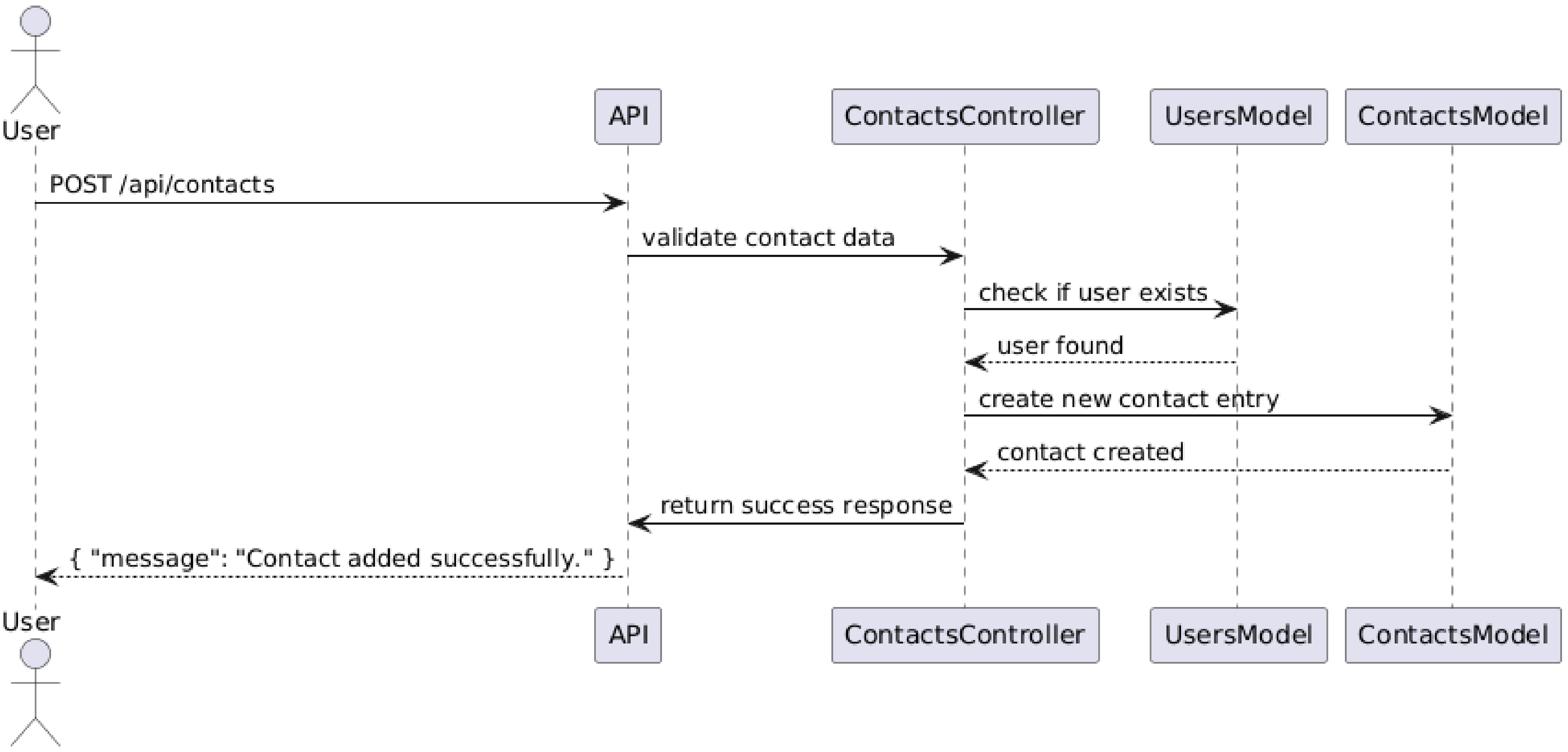
Add job application



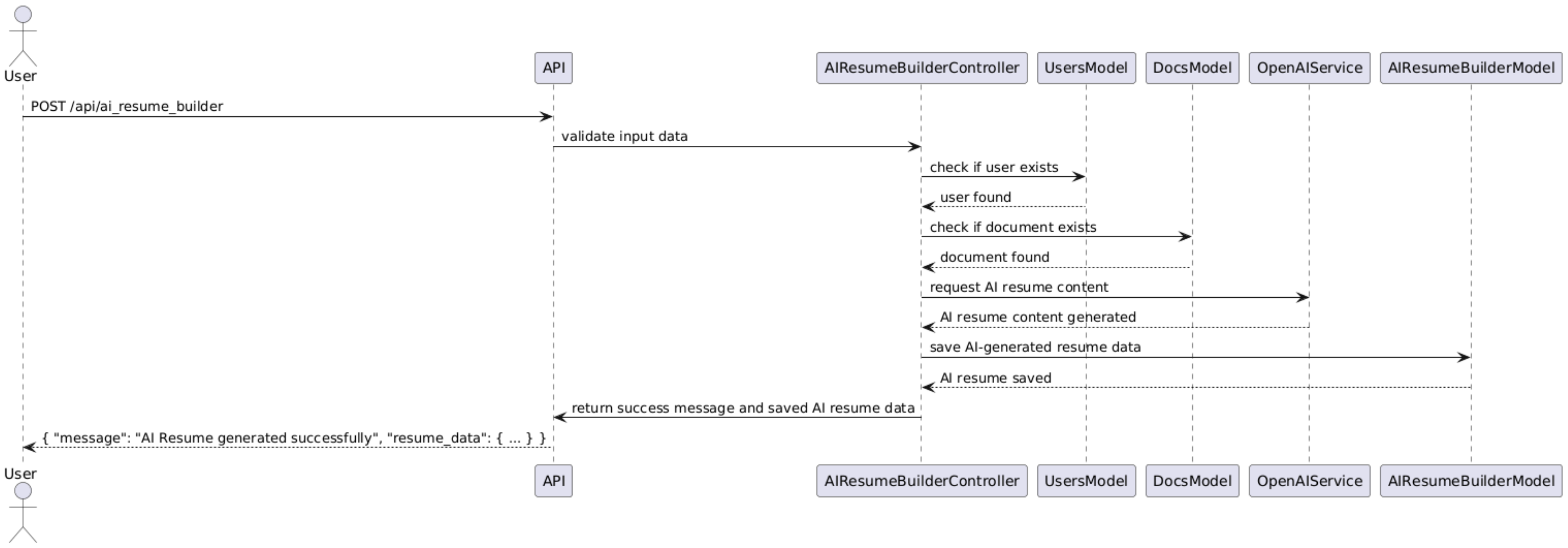
Search job application


User
API
JobsController
JobsModel
GET /api/jobs/search?company=XYZ&position=Engineer
search jobs by criteria
find jobs matching search criteria
return matching jobs
return job search results
[{ "id": 1, "company": "XYZ Corp", "position": "Software Engineer", "status": "in_progress"}]
User
API
JobsController
JobsModel

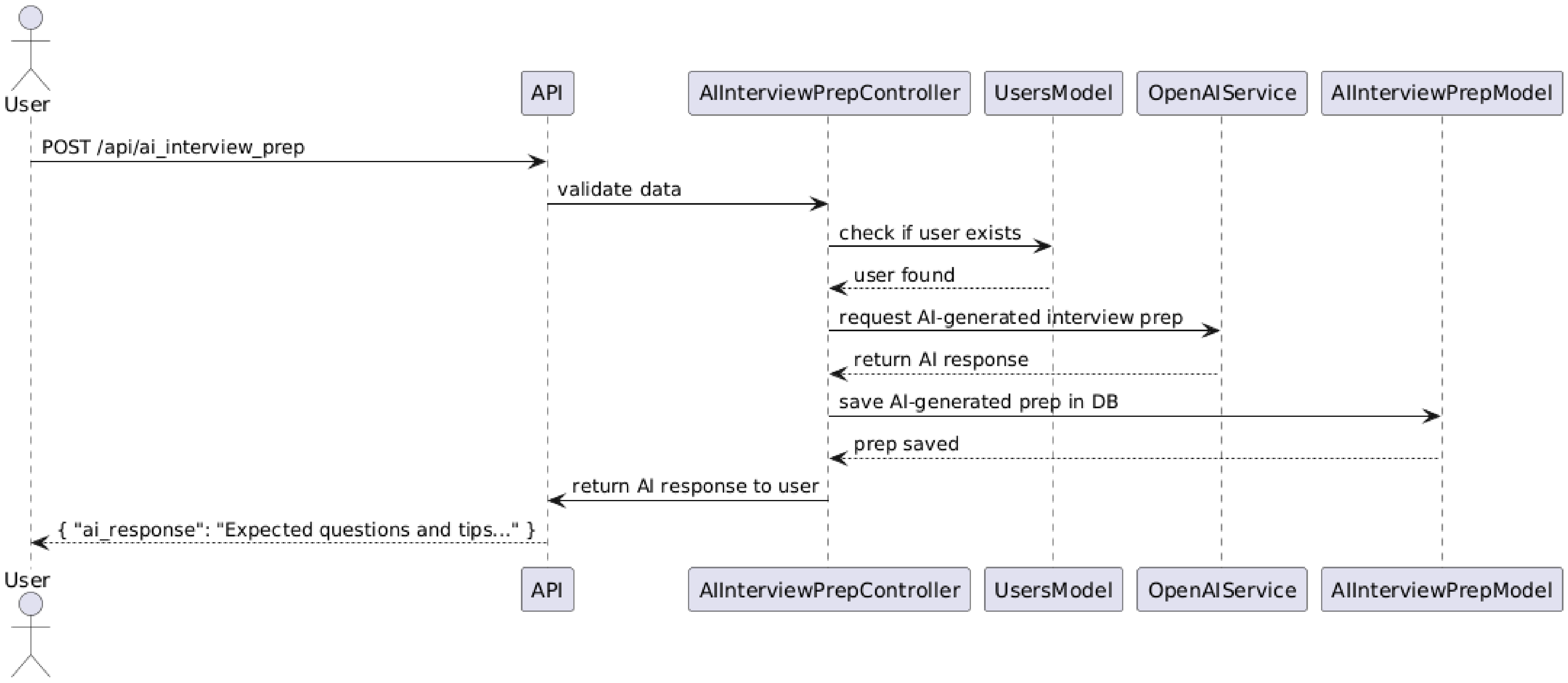
Add Contact



AI resume builder


User
API
AIResumeBuilderController
UsersModel
DocsModel
OpenAIService
AIResumeBuilderModel
POST /api/ai_resume_builder
validate input data
check if user exists
user found
check if document exists
document found
request AI resume content
AI resume content generated
save AI-generated resume data
AI resume saved
return success message and saved AI resume data
{ "message": "AI Resume generated successfully", "resume_data": { ... } }
User
API
AIResumeBuilderController
UsersModel
DocsModel
OpenAIService
AIResumeBuilderModel

AI interview prep


User
API
AIInterviewPrepController
UsersModel
OpenAIService
AIInterviewPrepModel
POST /api/ai_interview_prep
validate data
check if user exists
user found
request AI-generated interview prep
return AI response
save AI-generated prep in DB
prep saved
return AI response to user
{ "ai_response": "Expected questions and tips..." }
User
API
AIInterviewPrepController
UsersModel
OpenAIService
AIInterviewPrepModel

6. Identify actual key risks for your project at this time

Risk 1: Time Management

- **Issue:** Managing time effectively with other classes and work commitments, making it hard to study new technologies.
- **Resolution:** We will create a shared study plan, and split into smaller groups based on availability. For example, weekend study sessions for some and weekday ones for others, ensuring coverage and progress checks in our team meetings.
- **Sharing:** We'll communicate our progress in team meetings, ensuring everyone stays updated and accountable.

Risk 2: Task Tracking & Scheduling

- **Issue:** Lack of a transparent and up-to-date task schedule for each team member.
- **Resolution:** After each milestone is assigned, we will immediately document tasks in a shared Google Doc. Any updates or help requests will be communicated via group chat or during weekly meetings.
- **Sharing:** All members will have access to the Google Doc, ensuring real-time updates and task visibility.

Risk 3: Legal/Content Risks

- **Issue:** Ensuring we legally obtain and use APIs or software, preventing copyright or licensing issues.
- **Resolution:** We are actively researching APIs that align with our requirements and are compliant with copyright laws. We will consult legal guidelines to ensure proper usage.
- **Sharing:** The team will collectively review and approve any API selections in meetings to avoid legal complications.

7. Project management

For managing the tasks in Milestone 2 (M2), our team follows a transparent and collaborative approach. Here's a summary of how we manage and share our progress:

Progress Sharing in Scrum Meetings

- **Weekly Scrum Meetings**: We have scheduled weekly meetings every Wednesday where each team member shares their individual progress. These meetings are structured using a shared **Google Doc** agenda that is updated weekly, ensuring transparency in our discussions.
- **Communication Channels**: Outside of meetings, we maintain active communication via **Discord** and text messages, ensuring any blockers or updates are promptly shared with the team.
- **Scrum Updates**: Each member provides an update on the tasks they have completed, tasks they are currently working on, and any challenges they are facing. This process ensures that all progress is openly communicated.

Task Management Tool

- **GitHub Projects**: We use **GitHub Projects** to manage and track tasks assigned to each team member. Through this tool, we create **issues** for each task, assign them to the respective team members, and track their progress in real time. This ensures that the team stays organized and everyone is aware of their responsibilities.
- **Transparency**: By using GitHub Projects, the progress of each member's tasks is visible to the entire team, enhancing accountability and ensuring that everyone stays aligned on project goals.

This structured approach to task and progress management helps the team stay on track and ensures smooth collaboration.