Software Requirements Specification

for

Ascend Pro

Prepared by,

Adil Abdullah 2180103

Humdan Khattak 2080109

Daeyan Siddiqui 2180121

Supervised by,

Ma’am Ayesha Butt

SZABIST, Hyd

December 12, 2024

Table of Contents

Table of Contents ii

Revision History ii

1. Introduction 1

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Intended Audience and Reading Suggestions 1

1.4 Product Scope 1

1.5 References 1

2. Overall Description 2

2.1 Product Perspective 2

2.2 Product Functions 2

2.3 User Classes and Characteristics 2

2.4 Operating Environment 2

2.5 Design and Implementation Constraints 2

2.6 User Documentation 2

2.7 Assumptions and Dependencies 3

3. External Interface Requirements 3

3.1 User Interfaces 3

3.2 Hardware Interfaces 3

3.3 Software Interfaces 3

3.4 Communications Interfaces 3

4. System Features 4

4.1 System Feature 1 4

4.2 System Feature 2 (and so on) 4

5. Other Nonfunctional Requirements 4

5.1 Performance Requirements 4

5.2 Safety Requirements 5

5.3 Security Requirements 5

5.4 Software Quality Attributes 5

5.5 Business Rules 5

6. Other Requirements 5

Appendix A: Glossary 5

Appendix B: Analysis Models 5

Appendix C: To Be Determined List 6

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Initial Draft | 17-12-24 | Initial Version | 1.0 |
|  |  |  |  |

# Introduction

## Purpose

The purpose of this Software Requirements Specification (SRS) document is to define the functional and nonfunctional requirements for AscendPro, an AI powered employee development and meeting management platform. This document outlines the system's objectives, scope, and detailed specifications to guide the development team in designing and implementing the platform effectively.

## Document Conventions

* Bold text: Represents section headings and important terms.
* Italic text: Used for emphasis and to highlight key concepts.
* Numbered lists: Indicate steps in a sequence or prioritized items.
* Bullet points: Used for listing items without implying order.

## Intended Audience and Reading Suggestions

This document is intended for:

* Developers and Engineers: To understand the system requirements and guide the development process.
* Project Managers: To oversee project scope and ensure alignment with objectives.
* Stakeholders and Clients: To comprehend the features and benefits of the platform.
* Quality Assurance Team: To design test cases and validate system functionality.

**Reading Suggestions:**

* Begin with the Overall Description to get an understanding of the system context.
* Refer to the System Features section for detailed functional requirements.
* Review the Nonfunctional Requirements for performance and security considerations.

## Product Scope

AscendPro aims to assist employees in managing meetings, planning professional development, and achieving their career goals efficiently. By leveraging AI driven tools and integrating external APIs, the platform provides features such as:

* Meeting Management: Real time recording, audio analysis, and summarization of meetings.
* Employee Personas: Creation of AI generated personas of colleagues or supervisors to simulate interactions and enhance relationship building strategies.
* Skill Development Roadmaps: Personalized recommendations and learning paths based on individual skill levels.
* Content Recommendations: Tailored suggestions for resources and materials to support skill advancement.
* User Friendly Interface: A responsive design with both dark and light theme options to enhance the user experience.

The platform is designed to streamline workplace tasks, improve productivity, and foster personal growth within organizations.

## References

* React.js Documentation: https://reactjs.org
* Next.js Documentation: https://nextjs.org
* MongoDB Documentation: https://www.mongodb.com
* OpenAI API Documentation: https://openai.com/api
* Gemini API Documentation: https://ai.google.dev

# Overall Description

## Product Perspective

AscendPro is an AI powered employee development and meeting management platform designed to streamline workplace tasks and foster professional growth. Positioned as a standalone web application, it integrates various external AI based APIs to provide advanced functionalities such as real time meeting analysis, personalized skill development roadmaps, and AI generated colleague personas for relationship management practice. The platform addresses the inefficiencies of manual meeting management and the challenges employees face in skill advancement and interpersonal relationships within organizations.

## Product Functions

The key functions of AscendPro include:

* Meeting Management:
  + Real time meeting recording and audio analysis.
  + Automated meeting summaries and insights.
  + NLP based meeting scheduler integrated with employee data.
* Employee Development Tools:
  + Personalized skill development roadmaps based on individual skill levels.
  + Content recommendations tailored to support skill advancement.
* AI Generated Personas:
  + Creation of AI driven personas of colleagues or supervisors.
  + Simulation of interactions to practice relationship building strategies.
* User Interface Features:
  + Responsive design with both dark and light theme options.
  + Intuitive navigation and user friendly experience.

## User Classes and Characteristics

* Employees:
  + Middle aged professionals seeking to enhance productivity.
  + Interested in managing meetings efficiently and advancing their skills.
  + Looking to improve relationships with colleagues and superiors.
* Managers and Supervisors:
  + Need tools for effective team management and communication.
  + Desire insights into employee development and meeting outcomes.
* HR Professionals:
  + Focused on employee growth and organizational development.
  + Utilize analytics for training needs and performance assessments.

## Operating Environment

* Client Side:
  + Web browsers: Latest versions of Chrome, Firefox, Edge, and Safari.
  + Platforms: Windows, macOS, and Linux operating systems.
* Server Side:
  + Backend: Python based services handling API requests and data management.
  + Frontend: React.js / Next.js framework for building the user interface.
  + Database: MongoDB for storing user data, meeting archives, and AI generated content.
* External Services:
  + Integration with AI based APIs for functionalities like audio analysis, NLP processing, and persona generation.

## Design and Implementation Constraints

* API Dependencies:
  + Reliance on external AI APIs may be subject to availability, rate limits, and changes in third party services.
* Security and Privacy:
  + Must ensure secure handling of sensitive user data and compliance with data protection regulations.
* Performance Requirements:
  + Real time processing capabilities for meeting analysis and persona simulations.
* Compatibility:
  + Need to support multiple browsers and operating systems seamlessly.
* User Interface Considerations:
  + Implementation of dynamic theming requires careful design to maintain usability across themes.

## User Documentation

* Help Guides:
  + Comprehensive user manuals detailing platform features and usage instructions.
* Tutorials:
  + Interactive tutorials to onboard new users and demonstrate key functionalities.
* FAQs:
  + A repository of frequently asked questions to assist users in troubleshooting common issues.
* Support:
  + Contact options for technical support and feedback mechanisms for continuous improvement.

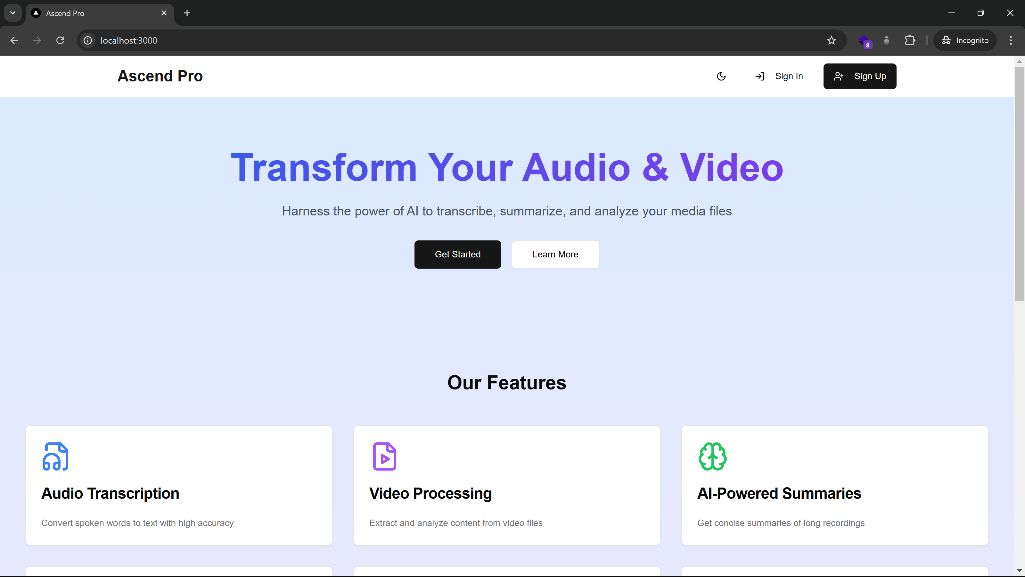
## Assumptions and Dependencies

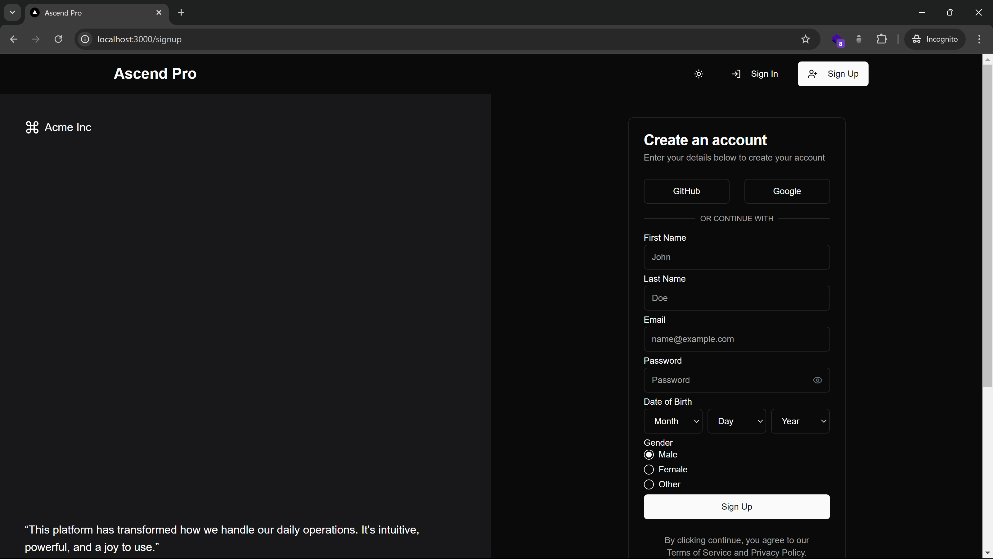
* Assumptions:
  + Users have access to a stable internet connection.
  + Organizations permit the use of AI powered tools and external APIs.
  + Users possess basic proficiency with web applications.
* Dependencies:
  + Functionality is dependent on third party AI APIs for core features.
  + The performance of the platform is tied to the efficiency of external services and the backend infrastructure.
  + Ongoing availability and support of the technologies used (React.js, Next.js, Python, MongoDB).

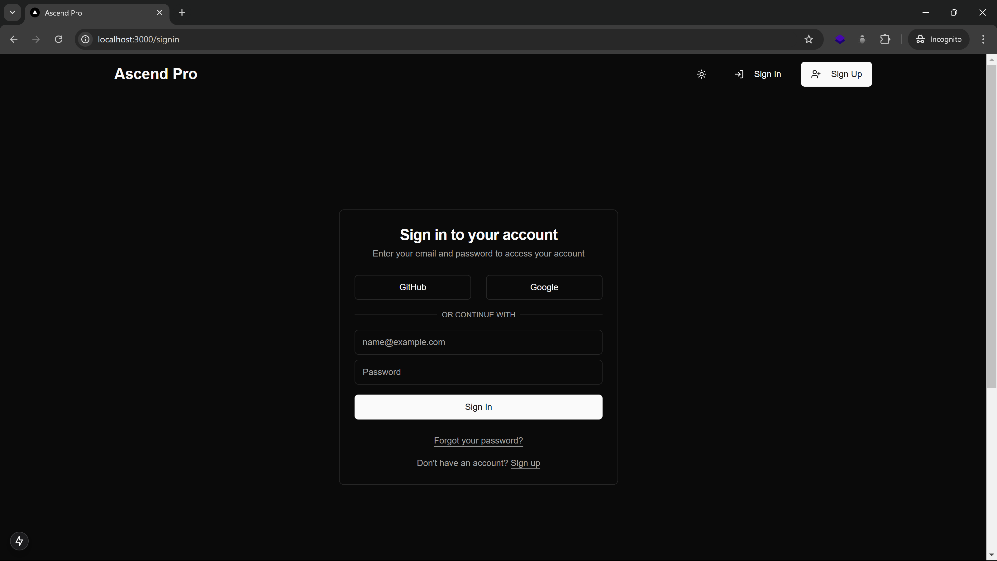
# External Interface Requirements

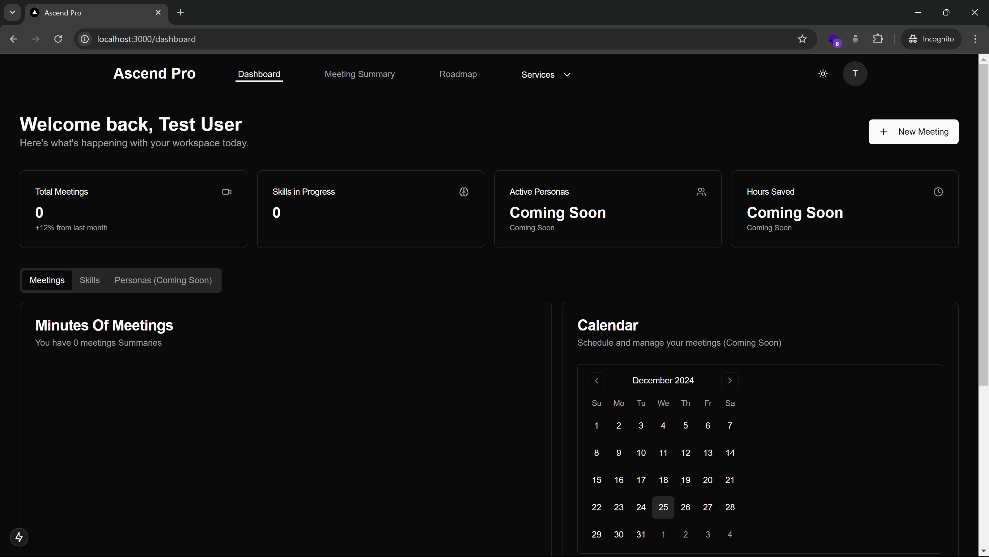
## User Interfaces

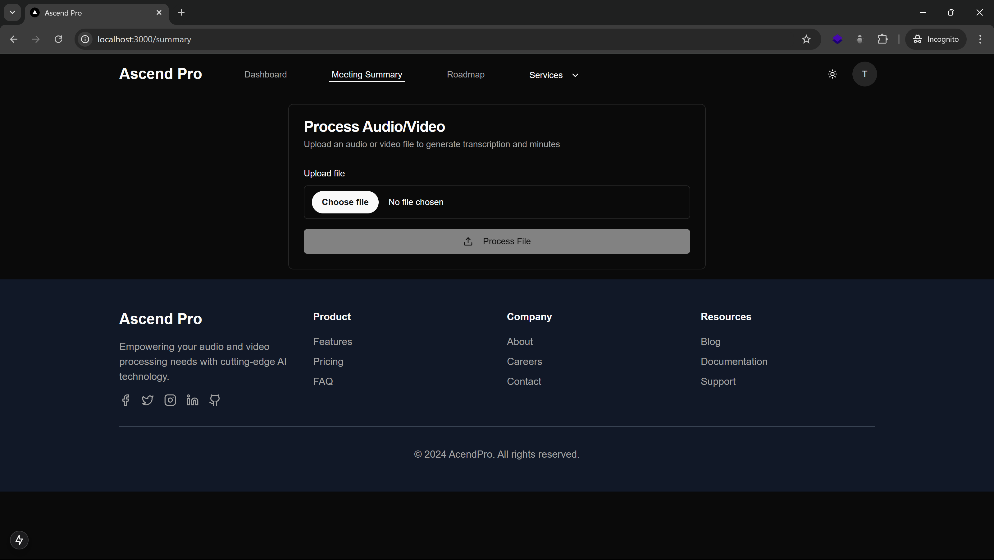
**Landing Page:**

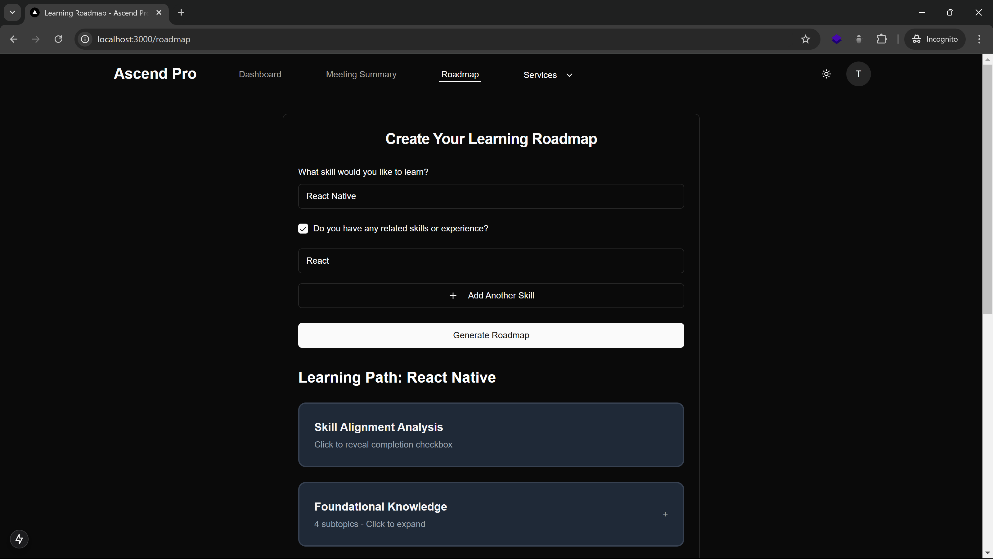
****

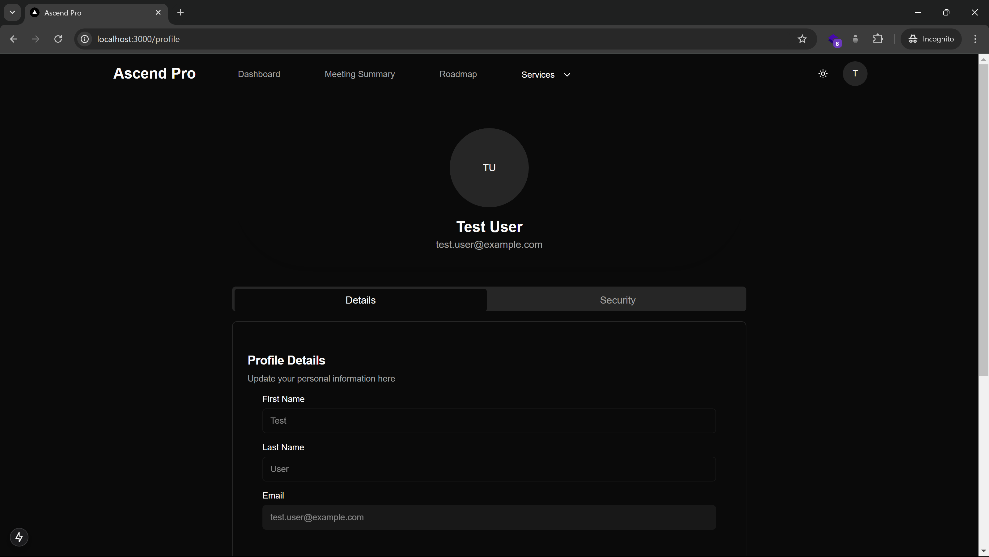
**Signup Page:**

**Signin Page:**

**Dashboard Page:**

**Summary Page:**

**Roadmap Page:**

**Profile Page:**

**Web Application Interface:**

* Responsive Design: Ensures usability across desktop and mobile devices.
* Theme Toggle: Users can switch between dark and light modes.
* Navigation Components:
  + Header: Contains navigation links, theme toggle, and user profile access.
  + Sidebar: Provides quick actions and displays recent activities.
  + Main Content Area: Dynamically updates based on user interactions.

const Layout = {

components: {

Header: {

navigation: ['Dashboard', 'Meetings', 'Skills', 'Personas'],

themeToggle: true,

userProfile: true

},

Sidebar: {

quickActions: ['Schedule Meeting', 'Create Persona', 'Skill Assessment'],

recentActivities: true

},

MainContent: {

dynamicRouting: true,

contentArea: true

}

}

}

**User Interaction Features:**

* Meeting Management: Schedule, record, and summarize meetings.
* Skill Development: Access personalized skill roadmaps and content recommendations.
* Persona Simulation: Interact with AI generated colleague personas for relationship training.

## Hardware Interfaces

**Required Hardware Components:**

* Microphone: For capturing meeting audio.
* Speakers/Headphones: For audio playback during meetings and interactions.
* Webcam (Optional): For video meetings and persona simulations.

**Hardware Configuration Specifications:**

const hardwareConfig = {

audio: {

input: {

sampleRate: 44100,

channels: 1,

format: 'audio/webm'

},

output: {

volume: 1.0,

playbackRate: 1.0

}

},

video: {

input: {

resolution: '720p',

frameRate: 30

}

}

}

**Device Compatibility:**

* Operating Systems: Windows, macOS, Linux.
* Browser Support: Latest versions of Chrome, Firefox, Edge, and Safari to ensure compatibility with hardware APIs.

## Software Interfaces

**External API Integrations:**

* Meeting Analysis:
  + Provider: OpenAI Whisper API
  + Endpoints: transcribe, analyze, summarize
* Persona Generation:
  + Provider: Google Gemini API
  + Endpoints: create\_persona, simulate\_interaction
* Skill Recommendations:
  + Provider: Custom Machine Learning API
  + Endpoints: assess\_skills, generate\_roadmap

API\_INTERFACES = {

'meeting\_analysis': {

'provider': 'OpenAI Whisper API',

'version': '1.0',

'endpoints': ['transcribe', 'analyze', 'summarize']

},

'persona\_generation': {

'provider': 'Google Gemini API',

'version': '1.0',

'endpoints': ['create\_persona', 'simulate\_interaction']

},

'skill\_recommendations': {

'provider': 'Custom ML API',

'version': '1.0',

'endpoints': ['assess\_skills', 'generate\_roadmap']

}

}

**Authentication and Authorization:**

* OAuth 2.0: Securely authenticate with external APIs.
* API Keys: Manage and rotate API keys to maintain security compliance.

**Data Formats:**

* Request and Response: JSON
* Audio Formats: WAV, MP3, WEBM
* Video Formats: WEBM, MP4

## Communications Interfaces

**Network Protocols:**

* HTTP/2: For efficient request handling and reduced latency.
* WebSockets: Enables real time communication for meeting updates and persona interactions.

NETWORK\_CONFIG = {

'protocols': {

'http': {

'version': 'HTTP/2',

'security': 'TLS 1.3'

},

'websocket': {

'enabled': True,

'ping\_interval': 30

}

},

'api\_rate\_limits': {

'requests\_per\_minute': 60,

'concurrent\_connections': 100

},

'data\_formats': {

'request': 'JSON',

'response': 'JSON',

'audio': ['wav', 'mp3', 'webm'],

'video': ['webm', 'mp4']

}

}

**Real time Communication Configuration:**

interface WebSocketConfig {

reconnection: {

attempts: 3,

delay: 1000,

backoff: 1.5

},

events: {

MEETING\_START: 'meeting:start',

MEETING\_END: 'meeting:end',

TRANSCRIPTION\_UPDATE: 'transcription:update',

PERSONA\_RESPONSE: 'persona:response'

}

}

**Security Measures:**

* TLS Encryption: Ensures secure data transmission.
* Firewall Configurations: Protects against unauthorized access.
* Rate Limiting: Prevents abuse of API endpoints.

**Reliability and Redundancy:**

* Load Balancing: Distributes traffic evenly across servers.
* Failover Mechanisms: Maintains service availability in case of server failures.

# System Features

## Meeting Management

**Description:**

The Meeting Management feature enables employees to efficiently schedule, record, and review meetings. It leverages AI driven tools to provide real time audio analysis, automated summaries, and actionable insights.

**Functionalities:**

* Real time Meeting Recording: Capture audio during meetings for accurate record keeping.
* Audio Analysis: Utilize AI to transcribe and analyze meeting discussions.
* Automated Summarization: Generate concise summaries of meetings to highlight key points and decisions.
* NLP based Scheduler: Intelligent scheduling tool that integrates with employees' calendars to arrange meetings optimally.

def start\_meeting(meeting\_id, participants):

# Initialize recording

recorder = AudioRecorder()

recorder.start()

# Notify participants

for participant in participants:

send\_notification(participant, "Meeting Started")

return "Recording started."

def summarize\_meeting(meeting\_id):

audio = retrieve\_audio(meeting\_id)

transcription = transcribe\_audio(audio)

summary = generate\_summary(transcription)

store\_summary(meeting\_id, summary)

return summary

## Employee Personas

**Description:**

This feature allows users to create and interact with AI generated personas of their colleagues or supervisors. It helps in practicing and improving interpersonal relationships within the workplace.

**Functionalities:**

* Persona Creation: Generate AI driven personas based on real or hypothetical colleagues.
* Interaction Simulation: Simulate conversations and scenarios to practice relationship management.
* Personality Analysis: Analyze and display personality traits of the personas to guide interactions.

## Skill Development Tools

**Description:**

Provides personalized skill development roadmaps and content recommendations to help employees advance their professional skills based on their current proficiency and career goals.

**Functionalities:**

* Skill Assessment: Evaluate employees' current skill levels using AI driven assessments.
* Personalized Roadmaps: Generate tailored development plans to achieve desired skill levels.
* Content Recommendations: Suggest relevant courses, articles, and resources to support skill growth.

def assess\_skills(user\_id):

user\_data = get\_user\_data(user\_id)

assessment = AIModel.evaluate\_skills(user\_data)

update\_user\_skills(user\_id, assessment)

return assessment

def generate\_roadmap(user\_id):

skills = get\_user\_skills(user\_id)

roadmap = AIModel.create\_roadmap(skills)

save\_roadmap(user\_id, roadmap)

return roadmap

## User Interface

**Description:**

The User Interface (UI) is designed to be intuitive and user friendly, offering a responsive design that adapts to various devices and supports both dark and light themes to enhance user experience.

**Functionalities:**

* Responsive Design: Ensures optimal display and functionality across desktops, tablets, and mobile devices.
* Theme Toggle: Allows users to switch between dark and light modes based on their preference.
* Intuitive Navigation: Easy access to all major features through a well organized layout.

import React, { useState } from 'react';

const ThemeToggle = () => {

const [theme, setTheme] = useState('light');

const toggleTheme = () => {

const newTheme = theme === 'light' ? 'dark' : 'light';

setTheme(newTheme);

document.body.className = newTheme;

};

return (

<button onClick={toggleTheme}>

Switch to {theme === 'light' ? 'Dark' : 'Light'} Mode

</button>

);

};

export default ThemeToggle;

## Content Recommendations

**Description:**

Delivers personalized content suggestions to employees, aiding in their continuous learning and professional growth. The recommendations are based on individual skill assessments and career objectives.

**Functionalities:**

* Personalized Suggestions: Tailors content to match the user’s current skills and desired career path.
* Resource Aggregation: Collects and organizes relevant courses, articles, and training materials from various sources.
* Progress Tracking: Monitors the user’s engagement with recommended content and adjusts suggestions accordingly.

export const getRecommendedContent = (userId) => {

const userSkills = getUserSkills(userId);

const recommendations = AIModel.fetchContentRecommendations(userSkills);

return recommendations;

};

## Real time Notifications

**Description:**

Provides real time updates and alerts to keep users informed about meeting schedules, task deadlines, and new content recommendations.

**Functionalities:**

* Meeting Alerts: Notify users of upcoming meetings and any changes to schedules.
* Task Reminders: Remind users of pending tasks and deadlines.
* Content Updates: Inform users about new content that matches their development roadmap.

## Data Analytics and Reporting

**Description:**

* Provides comprehensive analytics and reporting tools to track user engagement, meeting effectiveness, and skill development progress.

**Functionalities:**

* User Engagement Metrics: Monitor how users interact with the platform’s features.
* Meeting Effectiveness Reports: Analyze meeting data to assess productivity and identify improvement areas.
* Skill Development Progress: Track users’ advancement in their personalized skill roadmaps.

# Other Nonfunctional Requirements

## Performance Requirements

* Real time Processing: The platform must handle real time audio analysis and meeting summarization with minimal latency to ensure seamless user experience.
* Scalability: Capable of supporting a growing number of users and concurrent meetings without degradation in performance.
* Response Time: All user interactions, including loading pages and executing commands, should have a response time of less than 2 seconds.
* Throughput: The system should process up to 100 simultaneous API requests without performance bottlenecks

PERFORMANCE\_SETTINGS = {

'real\_time\_processing': True,

'scalability': {

'horizontal\_scaling': True,

'max\_users': 1000

},

'response\_time': {

'max\_seconds': 2

},

'throughput': {

'max\_concurrent\_requests': 100

}

}

## Safety Requirements

* Data Integrity: Ensure that all data transactions are reliable and that no data loss occurs during processing.
* Error Handling: Implement robust error handling to manage and log unexpected issues without crashing the system.
* Backup and Recovery: Regular backups must be taken, and a recovery plan should be in place to restore data in case of failures.

{

"data\_integrity": true,

"error\_handling": {

"log\_errors": true,

"notify\_admin": true

},

"backup\_recovery": {

"backup\_frequency": "daily",

"recovery\_plan": "standard\_procedure"

}

}

## Security Requirements

* Data Protection: All user data must be encrypted both in transit and at rest using industry standard encryption protocols.
* Authentication and Authorization: Implement OAuth 2.0 for secure user authentication and role based access control to restrict unauthorized access.
* Vulnerability Management: Regular security assessments and vulnerability scans must be conducted to identify and mitigate potential threats.
* Compliance: Ensure compliance with data protection regulations such as GDPR and CCPA.

## Software Quality Attributes

* Usability: The platform should offer an intuitive and user friendly interface, ensuring that users can navigate and utilize features with ease.
* Reliability: Maintain high availability with an uptime of 99.9%, ensuring that the platform is consistently accessible to users.
* Maintainability: Codebase should follow best practices and be well documented to facilitate easy maintenance and future updates.
* Portability: Ensure that the application can run seamlessly across different operating systems and browsers without compatibility issues.
* Accessibility: Comply with accessibility standards (e.g., WCAG 2.1) to make the platform usable for individuals with disabilities.

## Business Rules

* User Registration: Only authorized employees within an organization can register and access the platform.
* Data Usage: Collected data can only be used for the purposes outlined in the platform’s privacy policy and must not be shared with third parties without explicit consent.
* Subscription Plans: Different tiers of service (e.g., Basic, Professional, Enterprise) with varying levels of access and features.
* Service Level Agreements (SLAs): Define the expected performance and support levels, including response times for support requests and issue resolution.

# Other Requirements

## Feasibility Study

**Technical Feasibility**

* Technologies Used: The platform utilizes React.js and Next.js for frontend development, Python for backend services, and MongoDB for database management. These technologies are well supported with extensive documentation and community resources.
* API Integration: Integration with external AI APIs such as OpenAI Whisper and Google Gemini is achievable due to their robust APIs and comprehensive developer guides.
* Scalability: The chosen technologies support horizontal scaling, ensuring the platform can handle an increasing number of users and data without significant performance degradation.

**Operational Feasibility**

* User Friendly Interface: Designed with a responsive and intuitive UI, the platform caters to middle aged employees with varying levels of technical proficiency.
* Training and Support: Comprehensive user documentation, tutorials, and support channels will be provided to facilitate smooth adoption and usage.
* Maintenance: The modular architecture allows for easy maintenance and updates, minimizing downtime and disruptions.

**Economic Feasibility**

* Cost Efficiency: Utilizing open source technologies and existing AI APIs reduces initial development costs. The subscription based revenue model ensures ongoing financial sustainability.
* Resource Allocation: Adequate allocation of resources for development, testing, and deployment phases ensures the project remains within budgetary constraints.

**Schedule Feasibility**

* Development Timeline: A structured timeline outlines key milestones, including frontend and backend development, API integrations, testing, and deployment, ensuring timely project completion.
* Risk Management: Potential risks, such as API rate limits or technical challenges, are identified with mitigation strategies in place to adhere to the project schedule.

## Constraints

* Time Constraints: The project must be completed within the final year timeline, necessitating efficient time management and prioritization of essential features.
* Resource Limitations: Limited access to advanced AI development resources may impact the complexity and depth of AI driven features.
* API Dependencies: Reliance on third party AI APIs introduces dependencies related to availability, rate limits, and potential changes in API terms of service.
* Data Privacy Regulations: Compliance with data protection laws such as GDPR and CCPA is mandatory, affecting data handling and storage practices.

## Future Enhancements

* Mobile Application Development: Creating native mobile applications for iOS and Android to complement the web platform, enhancing accessibility and user engagement.
* Advanced Analytics: Incorporating more sophisticated analytics and machine learning models to provide deeper insights into employee development and meeting effectiveness.
* Integration with Enterprise Tools: Expanding integrations with popular enterprise software like Microsoft Teams, Slack, and Zoom to streamline workflows and increase platform utility.
* Multilingual Support: Adding support for multiple languages to cater to a diverse global user base, enhancing the platform's accessibility and usability.
* Enhanced Security Features: Implementing additional security measures such as multi factor authentication (MFA) and advanced encryption standards to further protect user data.

### Assumptions and Dependencies

* Stable Internet Connection: It is assumed that users will have access to a reliable internet connection to utilize the platform's features effectively.
* Organizational Support: The platform's success depends on the organization's willingness to adopt AI powered tools and integrate them into existing workflows.
* User Proficiency: Users are expected to have a basic level of proficiency with web applications to navigate and utilize the platform efficiently.
* Third Party API Availability: Continued availability and support of external AI APIs are critical for the platform's core functionalities.

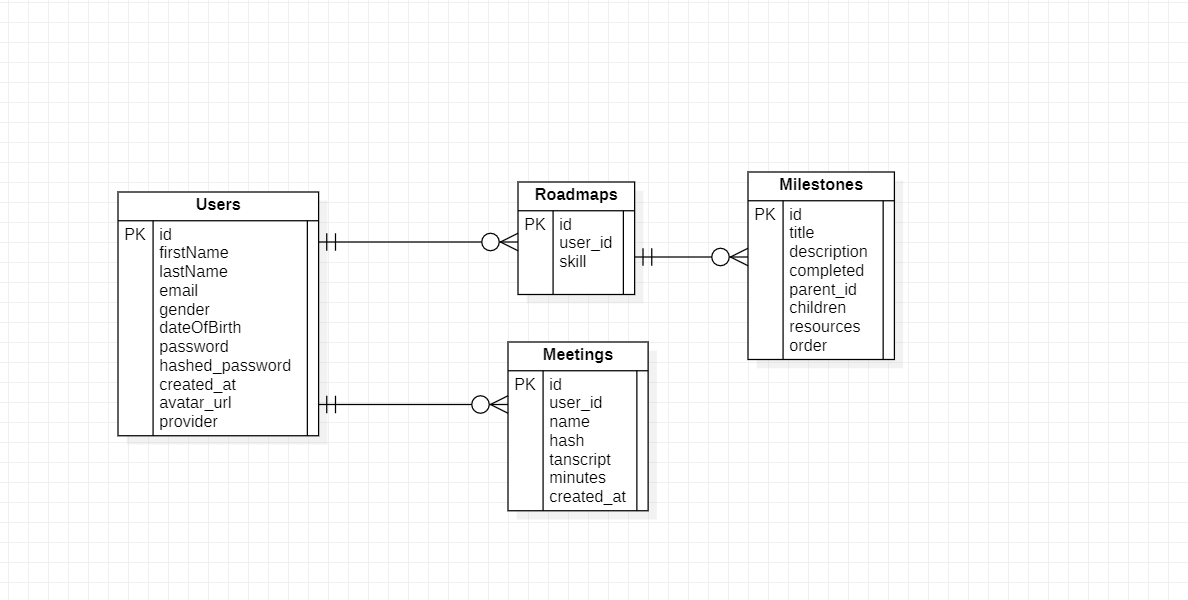
**Appendix A: Glossary**

|  |  |
| --- | --- |
| Term | Definition |
| AI (Artificial Intelligence) | The simulation of human intelligence processes by machines, especially computer systems, including learning, reasoning, and self correction. |
| API (Application Programming Interface) | A set of rules and protocols for building and interacting with software applications, allowing different systems to communicate with each other. |
| NLP (Natural Language Processing) | A branch of AI that focuses on the interaction between computers and humans through natural language, enabling machines to understand and respond to text or voice data. |
| SRS (Software Requirements Specification) | A comprehensive description of the intended purpose and environment for software under development, outlining functional and non functional requirements. |
| UI (User Interface) | The space where interactions between humans and machines occur, encompassing the design and layout of applications to enhance user experience. |
| UX (User Experience) | The overall experience and satisfaction a user has when interacting with a product, especially in terms of ease of use and efficiency. |
| MongoDB | A NoSQL database program that uses JSON like documents with optional schemas, known for its scalability and flexibility in handling large amounts of data. |
| React.js | A JavaScript library for building user interfaces, particularly single page applications, developed by Facebook. |
| Next.js | A React framework that enables functionality such as server side rendering and generating static websites for React based web applications. |
| OAuth 2.0 | An open standard for access delegation, commonly used to grant websites or applications access to user information without exposing passwords. |
| TLS (Transport Layer Security) | A cryptographic protocol designed to provide secure communication over a computer network, ensuring data privacy and integrity between two communicating applications. |
| GDPR (General Data Protection Regulation) | A regulation in EU law on data protection and privacy for individuals within the European Union and the European Economic Area. |
| CCPA (California Consumer Privacy Act) | A state statute intended to enhance privacy rights and consumer protection for residents of California, USA. |
| WebSockets | A protocol providing full duplex communication channels over a single TCP connection, enabling real time data transfer between client and server. |
| HTTP/2 | A major revision of the HTTP network protocol, focusing on performance improvements like multiplexing and header compression. |
| Load Balancing | A method to distribute network or application traffic across multiple servers to ensure reliability and performance. |
| Failover Mechanisms | Processes that automatically switch to a standby system or network upon the failure of the currently active system, ensuring continuity. |
| Unit Tests | Automated tests written and run by developers to ensure that a section of an application (known as the "unit") meets its design and behaves as intended. |
| Persona Simulation | The process of creating AI driven representations of real or hypothetical individuals to simulate interactions and practice relationship management strategies. |
| Skill Roadmaps | Personalized plans outlining the skills an individual needs to develop to achieve their career goals, often including recommended resources and milestones. |
| Real time Processing | The capability of a system to process data and provide immediate responses without noticeable delays, essential for functionalities like live audio analysis. |
| Scalability | The ability of a system to handle a growing amount of work by adding resources, ensuring consistent performance as demand increases. |
| Responsive Design | An approach to web design that makes web pages render well on a variety of devices and window or screen sizes, enhancing usability and accessibility. |
| Dark/Light Theme | Visual design options that allow users to switch between darker and lighter color schemes based on their preference or environmental lighting conditions. |

Appendix B: Analysis Models

B.1 Data Models

The AscendPro platform utilizes several data models to manage and organize information related to users, meetings, and skill development roadmaps. Below are the detailed descriptions of each primary data model, including their attributes and relationships.



B.1.1 User Model

The User model represents the individuals who interact with the AscendPro platform. It captures essential personal information, authentication details, and user preferences.

Attributes:

* firstName: The user's first name.
* lastName: The user's last name.
* email: The user's email address, used for authentication.
* gender: The user's gender.
* dateOfBirth: The user's date of birth (optional).
* password: The user's password (only in UserCreate).
* hashed\_password: The hashed version of the user's password (stored in UserInDB).
* created\_at: Timestamp of account creation.
* avatar\_url: URL to the user's avatar image (optional).

Relationships:

* Each User can have multiple Meetings and Roadmaps associated with their user\_id.

B.1.2 Meeting Model

The Meeting model manages information related to user meetings, including recordings, transcripts, and summaries.

Attributes:

* name: The title of the meeting.
* hash: SHA256 hash of the meeting's audio file for verification and integrity.
* transcript: Textual transcription of the meeting audio.
* minutes: Summarized minutes generated from the transcript.
* user\_id: Reference to the User who created the meeting.
* created\_at: Timestamp when the meeting was created.

Relationships:

Each Meeting is associated with one User via user\_id.

B.1.3 Roadmap Model

The Roadmap model outlines the skill development paths for users, including milestones and resources necessary for achieving specific career goals.

Attributes:

* id: Unique identifier for the roadmap.
* user\_id: Reference to the User associated with the roadmap.
* skill: The specific skill targeted for development.
* milestones: A list of Milestone objects outlining specific goals and steps.

Milestone Attributes:

* id: Unique identifier for the milestone.
* title: Title of the milestone.
* description: Detailed description of the milestone.
* completed: Status indicating whether the milestone is achieved.
* parent\_id: Reference to a parent milestone, if any.
* children: List of child milestones associated with this milestone.
* resources: Additional resources or materials related to the milestone.
* order: The sequence order of the milestone within the roadmap.

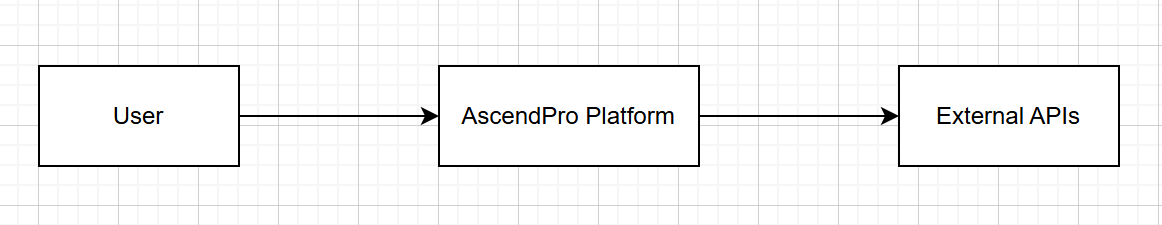
Relationships:

* Each Roadmap is linked to one User via user\_id.
* Milestones can have hierarchical relationships through parent\_id and children.

B.2 Data Flow Diagrams

B.2.1 Level 0: Context Diagram

Description: The Context Diagram provides a high level overview of the AscendPro system, illustrating its interactions with external entities.

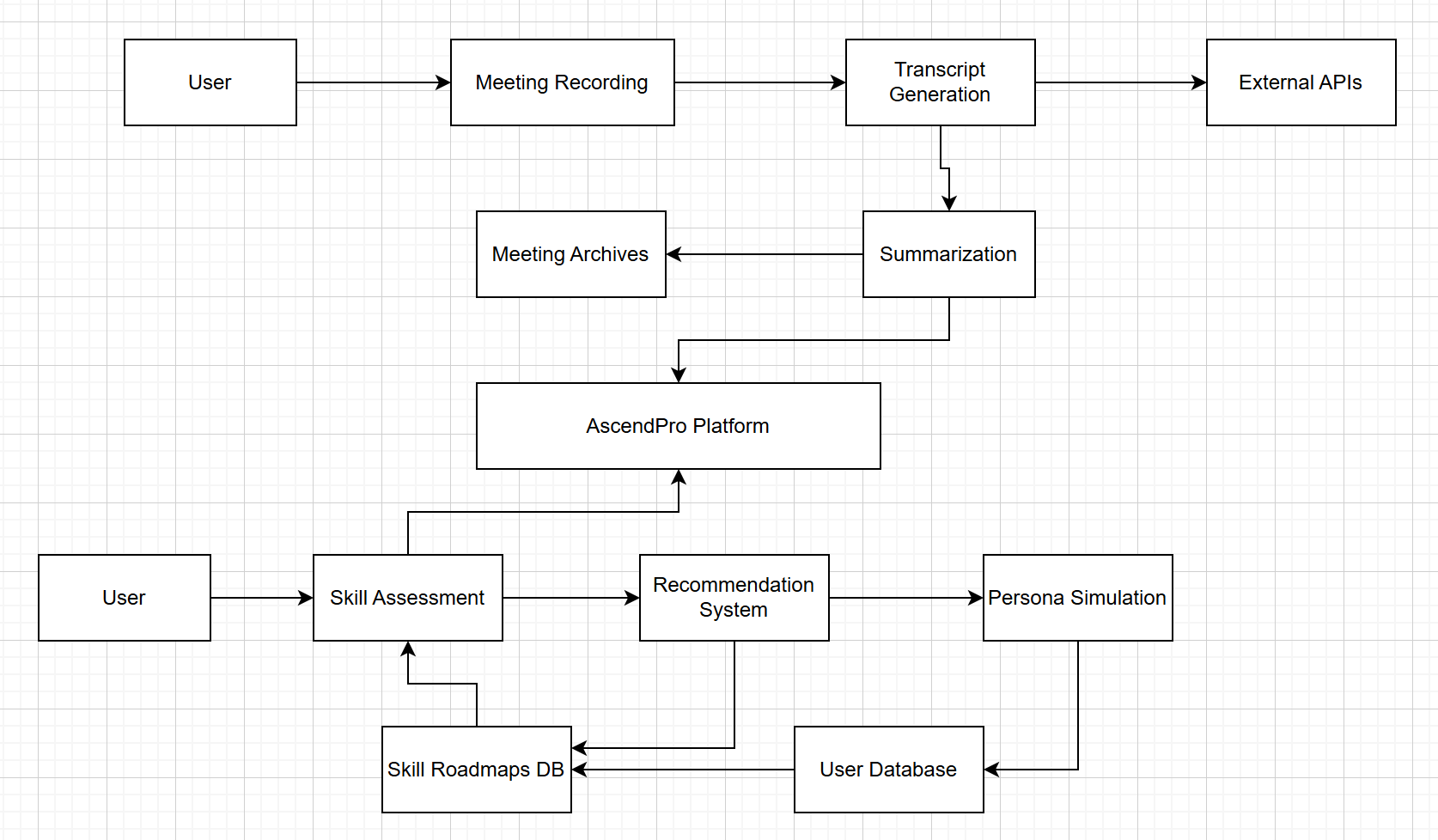


Components:

* User: Interacts with the AscendPro platform to manage meetings and develop skills.
* External APIs: Provide AI powered services such as audio analysis, persona generation, and skill recommendations.
* AscendPro Platform: Central system handling meeting management and employee development functionalities.

B.2.2 Level 1: Data Flow Diagram

Description: The Level 1 Data Flow Diagram breaks down the main processes within the AscendPro platform, detailing how data flows between different modules and external entities.



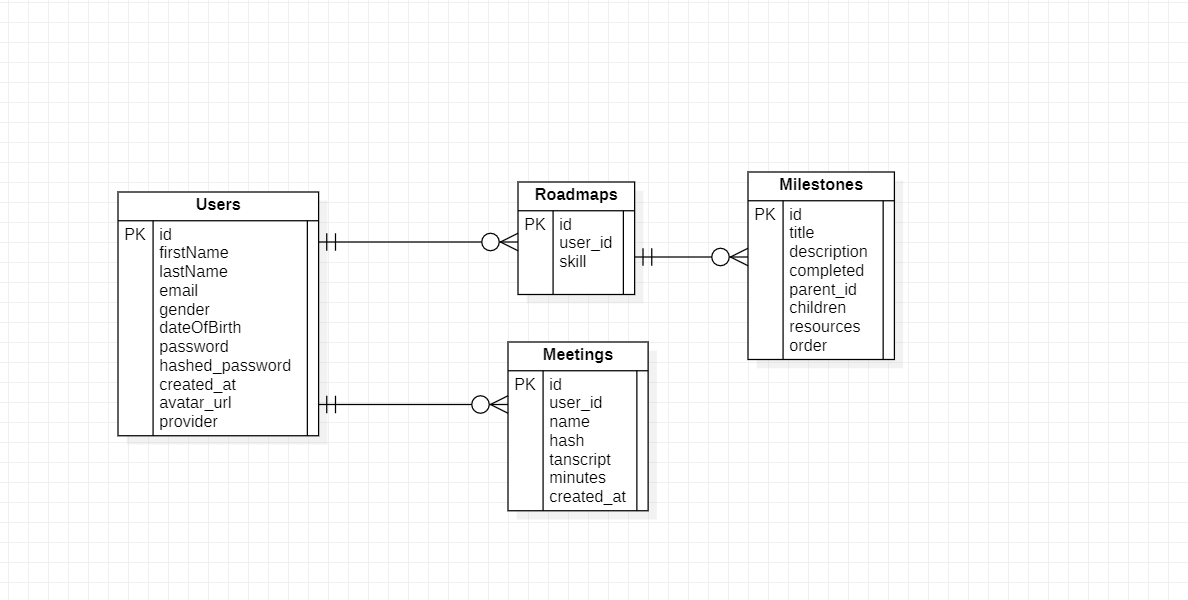
Processes:

* Meeting Management:
  + Meeting Recording: Captures audio from meetings.
  + Transcript Generation: Sends audio to External APIs for transcription.
  + Summarization: Generates meeting minutes from transcripts.
* Employee Development Tools:
  + Skill Assessment: Evaluates user skills using AI driven assessments.
  + Recommendation System: Provides personalized skill roadmaps and content suggestions.
  + Persona Simulation: Creates and interacts with AI generated personas for relationship training.

Data Stores:

* User Database: Stores user profiles, authentication details, and preferences.
* Meeting Archives: Stores meeting recordings, transcripts, and summaries.
* Skill Roadmaps Database: Stores users' skill development plans and milestones.

B.3 Entity Relationship Diagrams (ERD)



B.3.1 User and Meeting Relationship

Relationship:

* One to Many: A User can create multiple Meetings, but each Meeting is associated with only one User.

B.3.2 User and Roadmap Relationship

Relationship:

* One to Many: A User can have multiple Roadmaps for different skills, but each Roadmap is linked to one User.

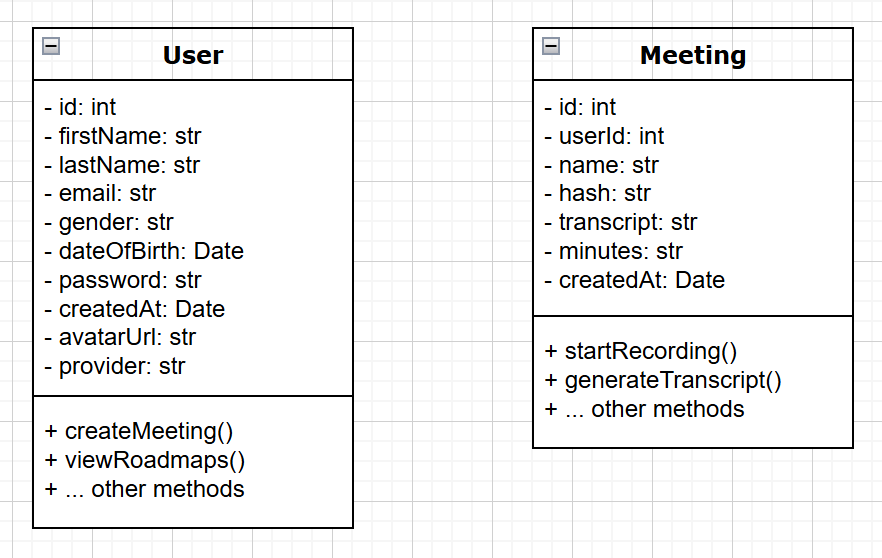
B.3.3 Roadmap and Milestone Relationship

Relationship:

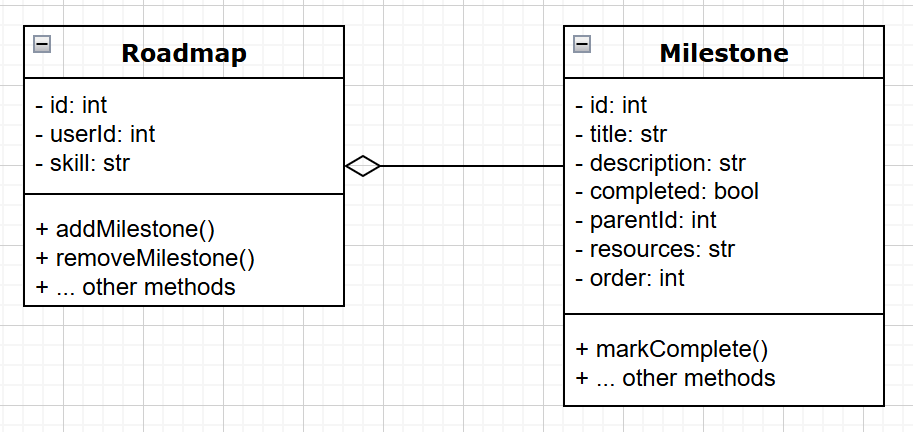
* One to Many (Hierarchical): A Roadmap consists of multiple Milestones. Milestones can have parent child relationships to represent dependencies or sequential steps.

B.4 Class Diagrams

B.4.1 User Class and Meeting Class



B.4.2 Roadmap and Milestone Classes

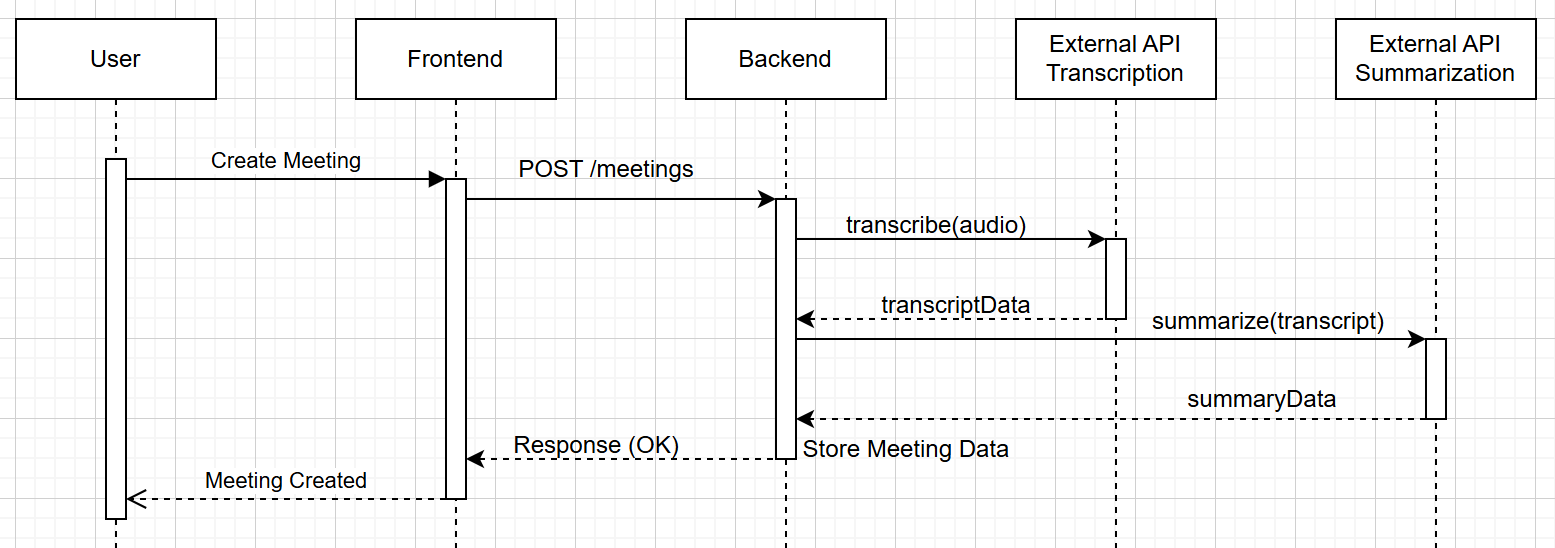


Notes:

* The Roadmap class aggregates multiple Milestone instances.
* The <>------ notation indicates aggregation, where a Roadmap is composed of multiple Milestones.

B.5 Sequence Diagrams

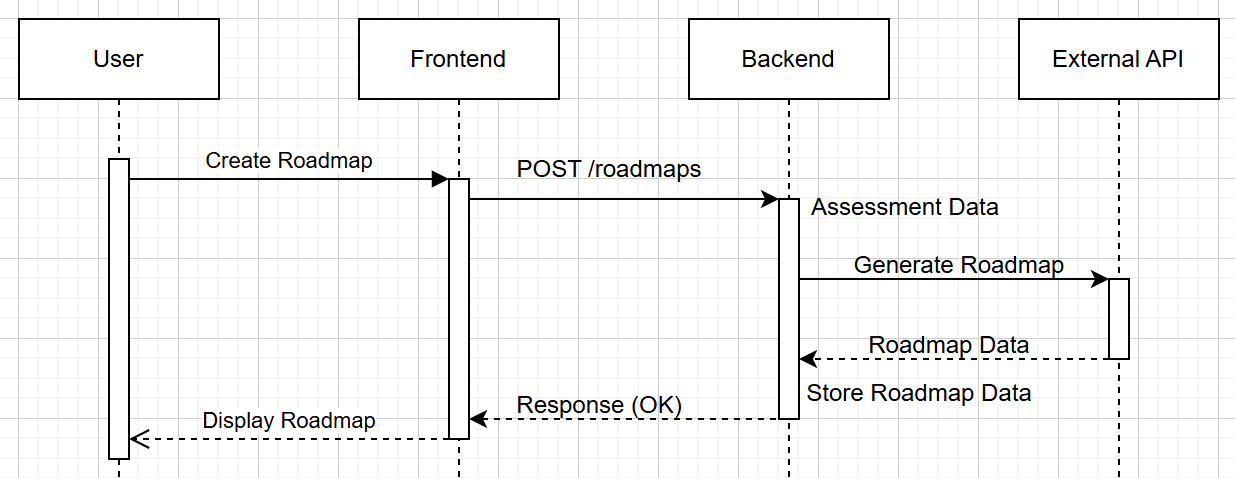
B.5.1 Creating a Meeting



Steps:

* User initiates the creation of a meeting via the Frontend.
* Frontend sends a POST request to the Backend to create the meeting.
* Backend calls the External API to transcribe the meeting audio.
* External API returns the transcript data to the Backend.
* Backend summarizes the meeting using another External API.
* External API returns the summary data.
* Backend stores the meeting details and responds with a confirmation to the Frontend.
* Frontend notifies the User of the successful creation of the meeting.

B.5.2 Generating a Skill Roadmap



Steps:

* User requests the generation of a skill roadmap via the Frontend.
* Frontend sends a POST request to the Backend to create the roadmap.
* Backend assesses the user's current skills.
* Backend generates a personalized roadmap using the assessment data via the External API.
* External API returns the roadmap data.
* Backend stores the roadmap and responds with a confirmation to the Frontend.
* Frontend displays the newly created roadmap to the User.

B.6 State Diagrams

B.6.1 Meeting Lifecycle

States:

* Scheduled: Meeting is scheduled but not yet started.
* In Progress: Meeting is currently ongoing.
* Recorded: Meeting audio has been recorded.
* Transcribed: Audio has been transcribed into text.
* Summarized: Meeting minutes have been generated.
* Completed: Meeting has ended.
* Archived: Meeting data is stored for future reference.

B.6.2 Roadmap Progression

States:

* Roadmap Created: A new skill roadmap is established.
* Milestone Pending: Milestones are created but not yet started.
* Milestone In Progress: A milestone is actively being worked on.
* Milestone Completed: A milestone has been achieved.
* Roadmap In Progress: The roadmap is actively being followed with milestones completed.
* Roadmap Completed: All milestones in the roadmap have been achieved.

B.7 Use Case Diagrams

B.7.1 User Interactions with Meeting Management

Actors:

* User: Initiates and manages meetings.
* Transcription API: Provides audio transcription services.
* Summarization API: Generates meeting summaries.

B.7.2 User Interactions with Skill Development

Actors:

* User: Requests skill assessments and roadmaps.
* AI Recommendation API: Processes skill data and provides recommendations.

B.8 Class Responsibility Collaborator (CRC) Cards

B.8.1 User Class

B.8.2 Meeting Class

B.8.3 Roadmap Class

B.9 Sequence of Operations

B.9.1 User Registration and Authentication

* User accesses the registration page via the Frontend.
* User submits registration details.
* Frontend sends a POST request to the Backend with user information.
* Backend hashes the user's password and creates a new User entry in the Database.
* Backend responds with a success message to the Frontend.
* Frontend notifies the User of successful registration.
* User logs in using their credentials.
* Frontend sends login credentials to the Backend.
* Backend verifies credentials and issues an authentication token.
* Frontend stores the token and grants User access to the platform.

B.9.2 Updating User Profile

* User navigates to the profile settings page via the Frontend.
* User updates profile information (e.g., name, avatar).
* Frontend sends the updated data to the Backend.
* Backend validates and updates the User record in the Database.
* Backend responds with the updated profile data.
* Frontend displays the updated profile to the User.

Appendix C: To Be Determined List

The following items are identified as needing further clarification, development, or decision making to complete the AscendPro platform's design and implementation. Addressing these items will ensure a comprehensive and robust system that meets all user and business requirements.

C.1 Functional Requirements

Mobile Application Development

* Determine the scope and features for native mobile applications on iOS and Android platforms.
* Decide whether to develop separate apps or use a cross platform framework.

Advanced Reporting Features

* Define the specific metrics and analytics to be included in user and administrative reports.
* Establish customizable reporting options for different user roles.

Integration with Additional Enterprise Tools

* Identify other enterprise software (e.g., Microsoft Teams, Slack, Zoom) for potential integration.
* Define the integration workflows and data synchronization methods.

Multilingual Support

* Determine the languages to be supported initially and plan for future additions.
* Establish a localization strategy for UI elements, content recommendations, and documentation.

C.2 Technical Requirements

API Endpoint Specifications

* Finalize the detailed specifications for all backend API endpoints.
* Define request and response formats, authentication mechanisms, and error handling procedures.

Scalability Strategies

* Decide on specific scalability solutions, such as containerization, microservices architecture, or serverless deployments.
* Plan for horizontal and vertical scaling based on projected user growth.

Performance Benchmarks

* Establish detailed performance benchmarks for various system components, including response times, throughput, and resource utilization.

Define acceptable performance thresholds and monitoring strategies.

Data Backup and Recovery Procedures

* Develop comprehensive data backup schedules and recovery plans.
* Determine the tools and technologies to be used for automated backups and disaster recovery.

C.3 Security Requirements

Multi Factor Authentication (MFA) Implementation

* Decide on the types of MFA to be supported (e.g., SMS, authenticator apps, hardware tokens).
* Define the user experience and fallback options for MFA processes.

Advanced Encryption Standards

* Determine additional encryption methods for sensitive data beyond TLS and AES 256.
* Plan for encryption key management and rotation policies.

Compliance with Additional Regulations

* Identify other relevant data protection regulations based on target markets (e.g., HIPAA, CCPA).
* Develop strategies to ensure ongoing compliance with evolving legal requirements.

C.4 User Interface and Experience

Final UI/UX Design Elements

* Complete the design of all user interface components, ensuring consistency and accessibility.
* Conduct user testing to gather feedback and make necessary adjustments.

Accessibility Enhancements

* Implement advanced accessibility features to support users with disabilities.
* Ensure compliance with all relevant accessibility standards (e.g., WCAG 2.1).

User Personalization Options

* Define the extent of user customization available within the platform (e.g., dashboard widgets, notification preferences).
* Develop personalization algorithms to enhance user experience based on behavior and preferences.

C.5 Content and Resource Management

Content Recommendation Sources

* Identify and curate the sources for courses, articles, and training materials used in content recommendations.
* Establish partnerships or licensing agreements with content providers as needed.

Dynamic Content Updating

* Develop mechanisms for regularly updating and refreshing recommended content to ensure relevance.
* Implement user feedback loops to refine content suggestions continuously.

C.6 Testing and Quality Assurance

Comprehensive Testing Plans

* Develop detailed testing strategies, including unit tests, integration tests, system tests, and user acceptance tests.
* Define testing tools and frameworks to be used throughout the development lifecycle.

Automated Testing Pipelines

* Set up continuous integration and continuous deployment (CI/CD) pipelines with automated testing stages.
* Ensure that all code changes are automatically tested before deployment.

C.7 Deployment and Maintenance

Deployment Strategies

* Decide on the cloud infrastructure and services (e.g., AWS, Azure, Google Cloud) for hosting the platform.
* Plan for automated deployment processes to ensure smooth rollouts and updates.

Maintenance Schedules

* Establish regular maintenance windows for system updates, patches, and performance optimizations.
* Develop monitoring and alerting systems to proactively address issues.

C.8 User Training and Support

Comprehensive Training Programs

* Design training modules and materials to help users effectively utilize all platform features.
* Plan for onboarding sessions, webinars, and workshops as needed.

Support Infrastructure

* Define the support channels available to users (e.g., live chat, email support, knowledge base).
* Establish response time targets and support quality standards.

C.9 Future Enhancements

AI Model Enhancements

* Plan for the continuous improvement of AI models used for audio analysis, persona generation, and skill recommendations.
* Explore the integration of advanced machine learning techniques to enhance feature capabilities.

Feature Expansion

* Identify potential new features based on user feedback and market trends.
* Prioritize feature development to align with strategic business goals.