**PeakNote**

**Introduction**

Modern enterprises and teams hold frequent meetings, and manually recording meeting minutes is not only time-consuming and inefficient but also prone to missing key information. This project aims to develop an intelligent meeting assistant that integrates AI technology into the PeakNote platform. By leveraging speech recognition and natural language processing (NLP), the system will enable automatic transcription and summarization of meeting content. The project objectives include reducing the burden of manual note-taking, improving meeting efficiency, and ensuring that participants can easily access clear and accurate meeting records and key points afterward.

With the rise of remote and hybrid work models, more organizations require AI-powered tools to support virtual meeting collaboration. Market research indicates that the AI meeting assistant sector is experiencing rapid growth, with a market size of approximately $2.5 billion in 2023, projected to reach $12.7 billion by 2031. Therefore, this project holds significant practical relevance and promising application potential.

**Market Research**

Based on the three core feature requirements—real-time speech-to-text transcription, AI-driven meeting summarization, and summarization of individual opinions—current meeting assistant software can be categorized into the following three types:​

**I. Direct Integration Solutions: Built-in Features in Real-Time Video Conferencing Platforms**

These platforms integrate the core features directly into their video conferencing services, allowing users to utilize them without installing additional plugins or third-party tools.​

**1. Zoom**

*•Real-time Speech-to-Text Transcription*: Provides live transcription during meetings, converting spoken words into text in real-time.​

*• AI-Driven Meeting Summarization*: Automatically generates meeting summaries post-session and emails them to the host.​

*• Summarization of Individual Opinions*: Identifies key discussion points, important conclusions, and action items through its smart recording feature.​

**2. Google Meet**

*• Real-time Speech-to-Text Transcription*: Offers live captioning using AI technology to provide real-time subtitles during meetings.​

*• AI-Driven Meeting Summarization*: Features like "Take notes for me" automatically capture meeting highlights and organize them into Google Docs.​

*• Summarization of Individual Opinions*: Provides a "Summary so far" feature to help latecomers quickly catch up on discussed content.​

**3. Microsoft Teams**

*• Real-time Speech-to-Text Transcription*: Offers live transcription, converting spoken content into text during meetings.​

*• AI-Driven Meeting Summarization*: Through its intelligent recap feature, it automatically generates meeting summaries, identifies key topics, and creates chapters.​

*• Summarization of Individual Opinions*: AI-generated notes capture discussion points and action items.​

**4. Tencent Meeting**

*• Real-time Speech-to-Text Transcription*: Supports real-time speech-to-text conversion, providing live subtitles.​

*• AI-Driven Meeting Summarization*: Automatically generates meeting minutes post-session, extracting key content.​

*• Summarization of Individual Opinions*: Intelligently analyzes speakers' viewpoints and opinions.​

**5. Feishu Meeting**

*• Real-time Speech-to-Text Transcription*: Provides live transcription, recording meeting content in real-time.​

*• AI-Driven Meeting Summarization*: Automatically generates meeting minutes, extracting key topics.​

*• Summarization of Individual Opinions*: Capable of distinguishing different speakers and organizing their respective viewpoints.​

**II. Indirect Integration Solutions: Implemented Through Plugins or Third-Party Tools**

These tools integrate with mainstream video conferencing platforms, acting as plugins or third-party services to provide the core features.​

**1. Otter.ai**

*• Real-time Speech-to-Text Transcription*: Can automatically join Zoom, Google Meet, or Microsoft Teams meetings to transcribe conversations in real-time.​

*• AI-Driven Meeting Summarization*: Automatically generates meeting summaries, capturing key points.​

*• Summarization of Individual Opinions*: Highlights key points and allows tagging teammates to assign action items.​

**2. Fireflies.ai**

*• Real-time Speech-to-Text Transcription*: Automatically records and transcribes meetings from various video conferencing applications.​

*• AI-Driven Meeting Summarization*: Generates meeting summaries, capturing key discussion points.​

*• Summarization of Individual Opinions*: Supports collaboration features, allowing users to add comments, tags, and emojis to specific parts of the conversation.​

**3. Fathom**

*• Real-time Speech-to-Text Transcription*: Records and transcribes meetings on Zoom, Google Meet, and Microsoft Teams.​

*• AI-Driven Meeting Summarization*: Automatically generates meeting summaries, extracting key content.​

*• Summarization of Individual Opinions*: Identifies and highlights key points and action items during meetings.​

**4. Tactiq**

*• Real-time Speech-to-Text Transcription*: Provides live transcription for Google Meet, Zoom, and Microsoft Teams.​

*• AI-Driven Meeting Summarization*: Generates summaries with one click and creates follow-up emails containing action items.​

*• Summarization of Individual Opinions*: Utilizes AI to extract insights, capturing and tagging each participant's contributions.​

**III. Post-Processing Solutions: Analysis Through Uploaded Audio or Video Files**

These tools do not participate directly in real-time meetings but transcribe and analyze content through user-uploaded audio or video files.​

**1. Alibaba Tongyi Tingwu**

*•Real-time Speech-to-Text Transcription*: Supports multi-language transcription and translation, providing real-time subtitles and intertranslation in languages like Chinese and English.​

*• AI-Driven Meeting Summarization*: Automatically generates meeting summaries, extracting key content.​

*• Summarization of Individual Opinions*: Capable of distinguishing different speakers and organizing their respective viewpoints.​

**2. Trint**

*• Real-time Speech-to-Text Transcription*: Transcribes video, audio, or live speech into over 40 languages.​

*• AI-Driven Meeting Summarization*: Automatically generates meeting summaries, capturing key points.​

*• Summarization of Individual Opinions*: Supports editing, collaboration, and integration with real-time workflows.​

Choosing the appropriate meeting assistant software should be based on your specific needs and existing workflows. If you aim to implement these features directly within real-time video conferences, opting for direct integration solutions may be more convenient. Conversely, if you seek in-depth analysis of meeting content post-session, post-processing solutions might be more suitable.

**Market Demand and Industry Trends**

Regardless of the solution, the demand for automated meeting transcription and intelligent summarization is growing rapidly. On one hand, enterprises seek to improve meeting efficiency by freeing employees from the burden of manual note-taking, allowing them to focus on discussions and decision-making. On the other hand, with the rise of remote and hybrid work models, meeting minutes have become a crucial component of both synchronous and asynchronous collaboration. Automated tools ensure that absent participants can stay informed about key discussions in a timely manner.

Research indicates that AI meeting assistants are evolving beyond simple transcription to provide deeper insights, including key point summarization, decision and action item extraction, and even task tracking based on meeting content. This trend is reflected in the functionalities of industry-leading products. For example, Otter.ai supports real-time transcription, multilingual recognition, and automatic summarization. Tencent Meeting's AI Assistant Pro generates real-time summaries during meetings, highlights participants' key concerns, and compiles action items post-meeting.

Overall, the industry is moving towards the deep integration of AI meeting assistant functionalities within major collaboration platforms to meet the increasing market demand.

**Implementation Goals**

Based on the aforementioned research, this project aims to develop a highly efficient and practical intelligent meeting transcription and summarization system with the following core functionalities:

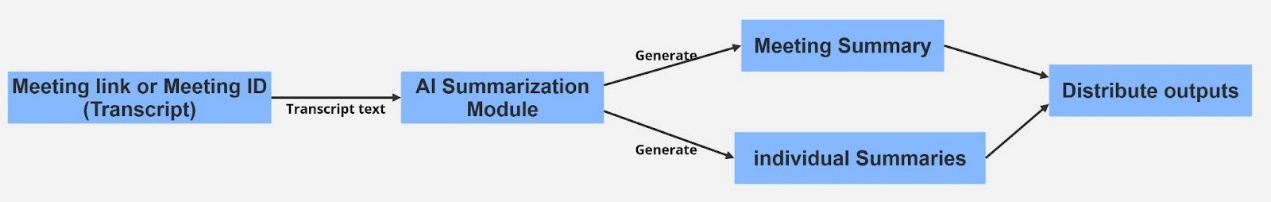
**1. AI-Powered Meeting Summarization**At the end of each meeting, the system will automatically generate a structured meeting summary based on the full transcription. The summary should extract key topics, decisions, conclusions, and important action items discussed during the meeting. By leveraging advanced natural language processing (NLP) and generative AI models, the summarization should be concise yet accurately reflect the discussion. For longer meetings, the system may offer sectional summaries or agenda-based summaries to ensure clarity and readability.

**2. Summarization of Individual Opinions**The system should generate an overview of each participant's contributions to the meeting. This feature will extract key opinions, concerns, or assigned tasks for each speaker and present them as brief summaries. This functionality is particularly valuable for decision-making meetings, as it clearly presents the viewpoints of different stakeholders. To achieve this, the system will classify transcription data by speaker and apply NLP techniques to distill each participant’s key statements.

**3. Action Item and Task Tracking**To enhance the execution of meeting outcomes, the system can automatically detect action items discussed in the meeting (such as task assignments and deadlines) and generate a to-do list. Each task should specify the responsible person and due date, making it easier to export into team task management tools. While this feature is of lower priority in the initial implementation, it will be considered if time allows, providing a more comprehensive meeting solution.

These core features will work together to create an end-to-end AI-powered meeting assistant, ensuring that participants receive AI support throughout the entire meeting lifecycle—from real-time transcription to post-meeting summaries and action tracking.

**Workflow Diagram**



As shown in the diagram, the workflow of this system consists of the following stages:

**1. Input Source (Meeting Link or Meeting ID):**

The system starts by retrieving the meeting transcript via the meeting link or ID, which serves as the input source for generating the transcribed text.

**2. Data Storage and Processing:**The transcribed text is temporarily stored in the system’s backend for further processing and summarization. To prevent data loss in case of network issues, the system supports local caching, ensuring that all transcriptions are securely stored until they can be fully processed.

**3. AI-Powered Summarization:**Once the transcript is obtained, it is processed by the AI Summarization Module. This module leverages Natural Language Processing (NLP) techniques to analyze the meeting content and extract key discussion points, decisions, and action items.

The AI module generates both a comprehensive meeting summary and individual summaries for each participant. These summaries focus on the main points of discussion, decisions made, and each participant’s contributions.

**4. Output and Distribution:**Once the summaries are generated, the system organizes and delivers the results. The outputs include:

•Displaying key meeting points and speaker overviews on the Web/App interface.

•Providing full-text transcriptions and summaries for download (e.g., generating PDF/Markdown meeting minutes).

•One-click sharing, such as sending the meeting summary via email to all participants or synchronizing it with collaboration platforms (e.g., Slack, DingTalk).

At this stage, the workflow is complete, achieving fully automated meeting transcription and summarization from speech input to structured meeting notes.

**Task Time Estimation**

To ensure an orderly development process, the overall project has been broken down into several key tasks, with estimated time requirements as follows:

**1. Requirement Analysis & Technology Selection (1 week)**This phase involves finalizing the functional requirements, selecting the appropriate speech recognition and AI summarization models, and designing the system architecture. The key tasks include:

• Finalizing the ASR engine selection (e.g., Whisper, Google Speech API) and NLP models for summarization (e.g., extractive vs. generative, fine-tuning needs).

• Determining the overall system requirements and setting clear goals for each phase.

**2. Core System Development (6 weeks)**

**• Backend & Database (2 week):** Develop a scalable backend architecture and storage model to handle real-time transcripts and meeting data.

**• Frontend UI Development (2 week):** Create an interactive interface to display real-time transcriptions, speaker identification, and keyword highlights.

**•AI Summarization & Opinion Extraction (2 weeks):** Implement automated meeting summarization using pre-trained NLP models (BERT, GPT, or TextRank), ensuring clear extraction of key discussions, decisions, and action items.

**3. System Integration & Optimization (2 weeks)**

**•** Ensure seamless interaction between real-time transcription, backend processing, and AI-generated summaries.

• Conduct end-to-end testing for transcription accuracy, summarization quality, and system performance.

• Optimize algorithms based on feedback from test results.

**4. Final Testing, Deployment & Documentation (1.5 weeks)**

**•** Comprehensive system testing to validate functionality, accuracy, and robustness under different scenarios.

• Deployment to a production or demo environment.

• Documentation for users and developers, including a system guide, API references, and troubleshooting steps.

**Team Roles & Contributions**

This project encompasses product management, cloud architecture, AI development, frontend and backend development, UI/UX design, and quality assurance. Successful implementation relies on a multifunctional team, with each member contributing in their respective roles:

* Ximeng Liu (Team Leader & QA Engineer): Oversees team management, task coordination, quality assurance, and testing, ensuring product stability and feature completeness.
* Rongze Gao (Product Owner): Manages product requirements, defines feature priorities, and collaborates with the team to align the product with market needs.
* Jamie Zhang (Cloud Engineer): Designs and deploys cloud infrastructure, optimizes system performance and scalability, and ensures project stability.
* Tianfa Zhu (UI/UX Designer): Designs the user interface and interaction experience, enhancing the usability and user-friendliness of the meeting assistant.
* Tianxiang Zhang (Software Developer): Handles frontend and backend development, implementing core functionalities and ensuring seamless business logic execution.
* Yuheng Li (AI Engineer): Develops speech-to-text and AI summarization algorithms, optimizing meeting content analysis and improving system intelligence.
* Zining He (Software Architect): Designs the system architecture, establishes technical standards, and ensures software stability and scalability.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Week 3 | | Week 4 | |
| Team Member | Have Done | Will Do | Have Done | Will Do |
| Ximeng Liu (Team Leader & QA Engineer) | - Established initial contact.  - Defined roles and responsibilities for collaboration. | - Create Project Repository on GitHub  - Assign Team Member Roles | - Contacted Stakeholders for the Initial Meeting  - Created Project Wiki Page | - Send Meeting Minutes and Next Meeting Agenda to Stakeholder |
| Rongze Gao (Product Owner) | - Assessed project feasibility and clarified the problem space  - Identified key challenges. | - Finalize key stakeholder questions to guide project refinement. | - Created a task and progress board to track weekly work updates.  - Defined Sprint 1 objectives based on stakeholder feedback. | - Maintain clear documentation of project progress, including completed tasks, key decisions, and challenges. |
| Jamie Zhang (Cloud Engineer) | - Built up team  connections, and set up the development environment. | - Investigate the current apps and existing project description. | - Investigation about current products and prepared questions for the first client meeting. | - Conduct market research on popular meeting apps and ai assistant tools. |
| Tianfa Zhu (UI/UX Designer) | - Gain a detailed understanding of the project structure and analyze its feasibility. | - Write user stories and scenarios for the project | - Write clear user stories and scenarios for the project, outlining user needs, goals, and expected outcomes. | - Create a landing page for the project |
| Tianxiang Zhang (Software Developer) | - Research project introduction, set up the development environment. | - Study the existing meeting assistant app | - Research the use of fathom on teams, Google meeting, zoom | - Study the technical feasibility of meeting assistants |
| Yuheng Li (AI Engineer) | - Help Simon to design the board of Scrum | - Wrote the user stories and scenarios and added it to the scrum board. | - Understanding the basic concept of Tensorflow to know how to apply the model to the meeting assistant | - Got some info about the Tensor, and tried the Tensorflow framework to apply the transfer between voice and text |
| Zining He (Software Architect) | - Set up the development environment and participate in creating a Git repository. | - Investigate and find the technologies and SDKs that might be needed. | - Investigate the SDK and tech stacks may need for target app. | - Research relates  online meeting platform, choose one and try to read its development documentation. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Week 5 | | Week 6 | |
| Team Member | Have Done | Will Do | Have Done | Will Do |
| Ximeng Liu (Team Leader & QA Engineer) | - Send follow-up team Meeting Minutes and Agenda to Stakeholder | - Update GitHub Project scrum board  - Maintain Sprint 1 backlog items | - Learned JavaScript and TypeScript basics to start with app design | - Plan for next sprint iteration items and discuss and assign them to team members |
| Rongze Gao (Product Owner) | - Conducted market research on related products.  - Established the project progress document. | - Design and create the paper prototype for the Meeting Assistant interface | - Created and updated the paper prototype.  - Updated the project documentation. | - Build a simple interface to access meeting records via shared links.  - Design the layout of the meeting record page |
| Jamie Zhang (Cloud Engineer) | - Market research on Microsoft Teams and Otter.ai, compared their features, and summarized desired functions for our product. | - Participate in the UI design process and refine the project documentation | - Analysis and documentation of risk management section, identifying key project risks and outlining corresponding mitigation strategies. | Continue improving the UI layout and study for the technology required in developing process. |
| Tianfa Zhu (UI/UX Designer) | - Create a landing page for the project. | - Modify the landing page based on client’s feedback. | - Refined the landing page for the project according to client’s feedback. | - Conduct a preliminary investigation into the UI design of similar plugins. |
| Tianxiang Zhang (Software Developer) | - Call the team api to get the transcript of the meeting | - Research ai tools that can analyze meeting content | - Study the ability of ai platforms such as gpt, deepseek, gemini and others to analyze meeting notes | - Investigate the feasibility of invoking the deepseek api to analyze meeting notes |
| Yuheng Li (AI Engineer) | - Deploy the template of the apps.  - Try to build the basic frame of app in Teams. | - Modify the user stories and scenarios to fit the features.  - Syncornize the features with the clients | - Update the user stories and scenarios to fit the features. | - Get the info about the API of mainstream AI open api. |
| Zining He (Software Architect) | - Investigate Microsoft Teams Developer Documentation and test some API provided by Microsoft. | - Continue researching the Microsoft Teams API and attempt to create a prototype. | - Participated in the design and improvement of the model. | - Continue studying the Teams development documentation and related tech stack. |

The team follows an Agile (Scrum) development process, progressing in iterative phases with regular progress assessments to ensure efficient collaboration and continuous improvement, ultimately delivering a high-quality product.

**Sprint Plan & Progress Tracking**

This project follows an Agile development approach, divided into multiple Sprint iterations, each lasting three weeks. Through short development cycles, we continuously deliver usable features, identify issues early, and adjust plans accordingly. Below is the planned Sprint breakdown and goals for each phase:

**Sprint 1: Planning & Foundation Setup**

**Timeline:** Weeks 3–6  
**Goal:** Finalize requirements, design system architecture, and set up the project framework.

**Tasks:**

* Confirm detailed requirements (e.g., transcription accuracy goals, summary format).
* Select speech recognition service (e.g., Google Speech API, Whisper) and preliminary NLP model.
* Set up the backend framework & database schema, implement basic API health checks.
* Build the frontend structure, preparing the UI for real-time transcription display.

**Acceptance Criteria:**

* A project plan & architecture document is delivered.
* Development environment is ready, with a basic backend-frontend integration demo running.

**Sprint 2: Initial Summarization Implementation**

**Timeline:** Weeks **7–9**

**Goal:** Develop a basic AI-powered meeting summarization module.

**Tasks:**

* Implement an extractive summarization model (e.g., TextRank) to generate concise meeting notes.
* Develop a frontend page for displaying post-meeting summaries.
* Implement individual speaker summaries, extracting key points from each participant’s contributions.
* Conduct small-scale testing to evaluate summary quality and coverage, gathering feedback for improvements.

**Acceptance Criteria:**

* After a meeting, the system generates an automatic summary, including key decisions and participant contributions.

**Sprint 3: UI and Database Systems Setup**

**Timeline:** Weeks **10-12**

**Goal:**  Build the frontend UI for displaying meeting summaries and set up data storage for meeting information.

**Tasks:**

* **Frontend Development**: Build a UI to display meeting summaries and speaker contributions.
* **Backend Development**: Set up the backend to store transcription data, summaries, and action items.
* **Database Setup**: Implement a database to store transcriptions, summaries, and speaker data.
* **Testing**: Test data flow from the database to the UI and ensure correct summary display.

**Acceptance Criteria:**

* Functional UI displaying meeting summaries and speaker contributions.
* Correct storage and retrieval of meeting data from the database.

**Sprint 4: Feature Enhancement & Optimization**

**Timeline:** Weeks **1–3** (Second Semester)  
**Goal:** Improve system intelligence, usability, and completeness.

**Tasks:**

* Optimize speech recognition parameters to improve transcription accuracy and reduce filler words.
* Upgrade summarization models (e.g., integrate generative AI) to refine meeting notes, extract key decisions & action items.
* Enhance speaker summarization logic, ensuring accurate representation of individual viewpoints.
* Improve frontend usability, adding features like summary editing, export to Markdown/PDF, and real-time meeting annotations.
* Perform full system testing, including stress testing and edge case handling.

**Acceptance Criteria:**

* Core features function smoothly, transcription accuracy exceeds 90%, and summaries accurately capture meeting content.
* User experience is refined based on iterative improvements.

**Sprint 5: Performance & Security Optimization**

**Timeline:** Weeks **4–6** (Second Semester)  
**Goal:** Enhance system performance, security, and scalability.

**Tasks:**

* Further optimize ASR models to improve accuracy in noisy environments.
* Strengthen data security & encryption, ensuring compliance with privacy regulations.
* Improve system performance, reducing latency and increasing concurrent meeting support.
* Conduct large-scale user testing, gather feedback, and fix critical issues.

**Acceptance Criteria:**

* The system performs well under noisy conditions and supports multiple simultaneous meetings.
* All meeting data is securely encrypted, ensuring compliance with security standards.

**Sprint 6: Final Deployment & Documentation**

**Timeline:** Weeks **7–9** (Second Semester)  
**Goal:** Finalize the project for deployment and release.

**Tasks:**

* Fix remaining bugs and fine-tune system performance.
* Deploy the demo version for product validation, ensuring correct configuration and monitoring.
* Prepare user training materials and technical documentation (e.g., Quick Start Guide, System Architecture Overview).
* Conduct a retrospective to document key lessons learned for future iterations.

**Acceptance Criteria:**

* The project passes final validation, documentation is complete, and the team aligns on next-phase plans (if applicable).

By following this Agile framework, the team ensures transparency, adaptability, and continuous progress, allowing the project to evolve toward its final goal of an AI-powered meeting assistant.

**Project Risks**

**1.** **Poor AI Summary Model Performance**Meeting content often includes informal expressions and industry-specific terminology, which may result in inaccurate or unclear summaries generated by the model.

**2. Third-Party API Dependency**Speech recognition and natural language processing rely on third-party APIs (such as OpenAI, Google, AWS). Changes in pricing, service restrictions, or instability may affect the system's core functionality.

**3. Data Privacy and Security Risks**Meeting content may contain sensitive information. Data storage and processing must comply with regulations such as GDPR and CCPA to ensure security and privacy protection.

**4. User Dissatisfaction with AI-Generated Summaries**AI may not fully understand the meeting context, leading to summaries that lack key points or do not meet user expectations.

**5. High Market Competition and Investment Risks**

The market already has many similar products, making it challenging to differentiate and gain user adoption. Entering the market requires strong product competitiveness and strategic investment.

**Risk Management**

To ensure the success and sustainability of the project, a proactive risk management strategy has been designed. The following table outlines the major identified risks, their potential impact, likelihood, and the corresponding mitigation strategies:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk | Description | Impact Level | Likelihood | Mitigation Strategy |
| Poor AI Summary Model Performance | Informal language, ambiguous expressions, or industry-specific terminology may degrade the quality of generated meeting summaries. | High | Medium | Fine-tune NLP models on domain-specific datasets; allow users to edit AI-generated summaries; incorporate user feedback loops to iteratively improve summarization accuracy. |
| Third-Party API Dependency | Dependence on external APIs (e.g., OpenAI) may lead to service instability or cost volatility. | High | Medium | Implement abstraction layers for API calls, allowing quick switching between providers; monitor API pricing and usage limits; maintain a minimal viable offline fallback solution. |
| Data Privacy and Security Risks | Sensitive meeting data may pose legal and ethical challenges if mishandled or exposed. | High | High | Encrypt data in transit and at rest; provide user consent and data deletion features. |
| User Dissatisfaction with AI Summaries | Users may find AI-generated content inadequate or lacking specific highlights. | High | Medium | Include manual edit functionality; offer different summary formats (e.g., bullet points, paragraph view); allow summary feedback or ratings to guide future model refinement. |
| High Market Competition and Investment Risks | The market is saturated with similar products, increasing the risk of low user adoption. | Medium | High | Focus on seamless Teams integration to target a specific niche; conduct early user testing and iterate on product-market fit. |

**Monitoring & Response Plan**

- Weekly Review: The team will assess risk indicators during sprint retrospectives and planning sessions.  
- Risk Dashboard: A real-time dashboard will be used to track high-priority risks and their status, ensuring transparency and timely action.  
- Contingency Planning: For critical risks (e.g., API failure, data breach), contingency protocols will be documented and rehearsed to ensure system continuity.