Meeting Assistant Application: System Architecture and UI Design

Table of Contents

- System Architecture
  - Component Diagram
  - Data Flow
  - Technical Stack Recommendations
  - Standalone App vs. Browser Extension
  - Security Considerations
- User Interface Design
  - Wireframes
  - User Flow
  - Key Features and Controls
  - Accessibility Considerations
- Implementation Plan
  - Development Phases and Milestones
  - Testing Strategy
  - Deployment Considerations

System Architecture

Component Diagram

The following component diagram illustrates the major parts of the Meeting Assistant application and their relationships:

```
graph TD
    subgraph "macOS Device"
        A[Audio Capture Module] --> |Raw Audio| B[Audio Processing]
        B --> |Processed Audio| C[Transcription Engine]
        C --> |Text Transcript| D[OpenAI Integration]
        D --> |Summaries/Insights| E[User Interface]

        F[System Audio] --> A
        G[Microphone Input] --> A

        H[Local Storage] <--> E
        H <--> C
        H <--> D

        I[Meeting Platform Detection] --> E
        I --> A
    end

    subgraph "Cloud Services"
        J[OpenAI API] <--> D
        K[Microsoft SSO] <--> E
        L[Cloud Storage] <--> H
    end

    style A fill:#f9d5e5,stroke:#333,stroke-width:2px
    style B fill:#eeeeee,stroke:#333,stroke-width:2px
    style C fill:#d5e8f9,stroke:#333,stroke-width:2px
    style D fill:#e5f9d5,stroke:#333,stroke-width:2px
    style E fill:#f9e5d5,stroke:#333,stroke-width:2px
```

Data Flow

The data flows through the system as follows:

```
sequenceDiagram
    participant SA as System Audio
    participant MI as Microphone Input
    participant ACM as Audio Capture Module
    participant AP as Audio Processing
    participant TE as Transcription Engine
    participant OAI as OpenAI Integration
    participant UI as User Interface
    participant LS as Local Storage
    participant CS as Cloud Services

    SA->>ACM: Raw system audio
    MI->>ACM: Raw microphone input
    ACM->>AP: Combined raw audio
    AP->>TE: Processed audio stream
    TE->>LS: Store raw transcript
    TE->>OAI: Send transcript chunks
    OAI->>CS: API requests
    CS->>OAI: AI-generated summaries/insights
    OAI->>UI: Display summaries/insights
    OAI->>LS: Store summaries/insights
    UI->>LS: User preferences/settings
    LS->>CS: Backup (optional)
```

Technical Stack Recommendations

Audio Capture and Processing

- **BlackHole**: Virtual audio device for system audio capture
- **Core Audio Framework**: For direct access to audio streams
- **AudioKit**: For audio processing and manipulation
- **FFmpeg**: For audio format conversion if needed

Transcription Engine

- **Whisper AI**: Local transcription for privacy and reduced latency
- **Vosk**: Alternative offline speech recognition
- **WebSpeech API**: For browser extension option

Backend/Core

- **Electron**: For cross-platform desktop application
- **Node.js**: For backend processing
- **Express**: For API endpoints if needed
- **SQLite**: For local storage of transcripts and settings

Frontend

- **React**: For building the user interface
- **Tailwind CSS**: For styling
- **Electron Store**: For persistent settings

Cloud Integration

- **OpenAI API**: For summarization and insights
- **Azure AD Authentication Library**: For Microsoft SSO integration
- **Intune SDK**: For corporate security compliance

Mobile Integration (Future)

- **React Native**: For iOS companion app
- **WebSockets**: For real-time communication between desktop and mobile

Standalone App vs. Browser Extension

Standalone App (Electron)

Pros: - Full system-level access to audio devices - Better performance for audio processing and transcription - Works across all meeting platforms without platform-specific code - More robust offline capabilities - Better security control for corporate environments - Can implement deeper OS integration

Cons: - Larger installation footprint - Requires separate installation and updates - More complex deployment in corporate environments - Higher development complexity

Browser Extension

Pros: - Easier distribution through browser stores - Simpler updates - Lighter resource footprint - More familiar installation process for users - Potentially easier approval in some corporate environments

Cons: - Limited access to system audio (browser tab audio only) - Platform-specific implementations needed for each meeting service - Limited offline capabilities - Less control over security aspects - Dependent on browser permissions and API changes

Recommendation

Based on the requirements for system audio capture across multiple meeting platforms and the need for corporate security compliance, the **standalone Electron app** is recommended as the primary approach. This provides the necessary system-level access for audio capture and offers better security controls for corporate environments.

A phased approach could include: 1. Develop the core standalone app first 2. Consider a browser extension as a complementary lightweight option later 3. Develop the iOS companion app as a final phase

Security Considerations

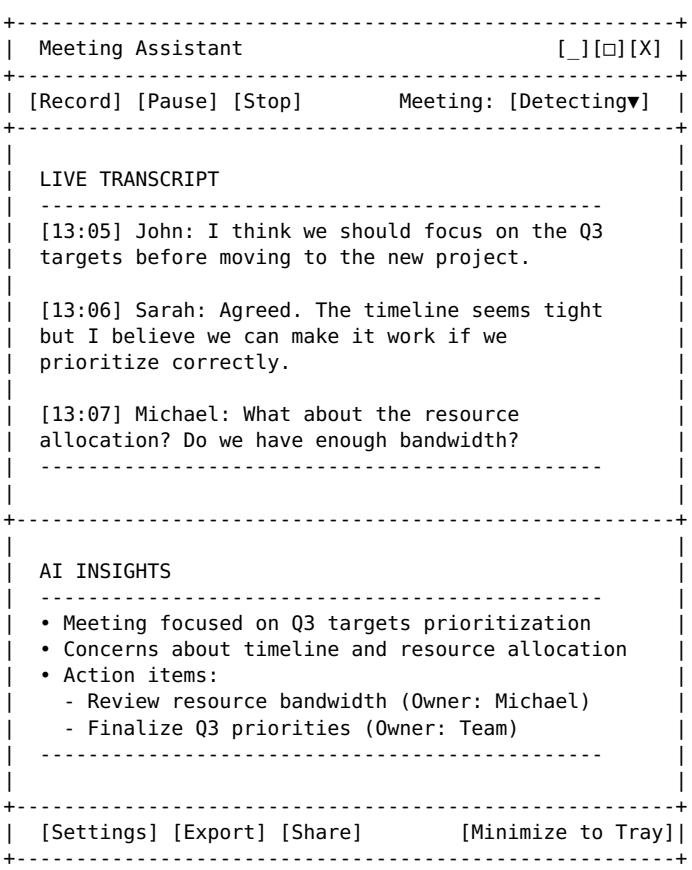
1. **Data Privacy**
   - All audio processing and transcription should happen locally when possible
   - Transcripts should be encrypted at rest
   - Clear data retention policies with options for automatic deletion
2. **Corporate Compliance**
   - Microsoft Intune integration for device management
   - Support for Microsoft SSO authentication
   - Compliance with corporate data loss prevention (DLP) policies
   - Audit logging for security monitoring
3. **API Security**
   - Secure storage of API keys
   - Rate limiting for API calls
   - Minimal data transmission to external services
4. **Application Security**
   - Code signing for application distribution
   - Regular security updates
   - Sandboxed execution environment

    - Permission-based access to system resources
5. **User Control**
    - Clear recording indicators
    - Easy meeting pause/resume controls
    - Ability to redact sensitive information before sharing
    - Granular permission settings

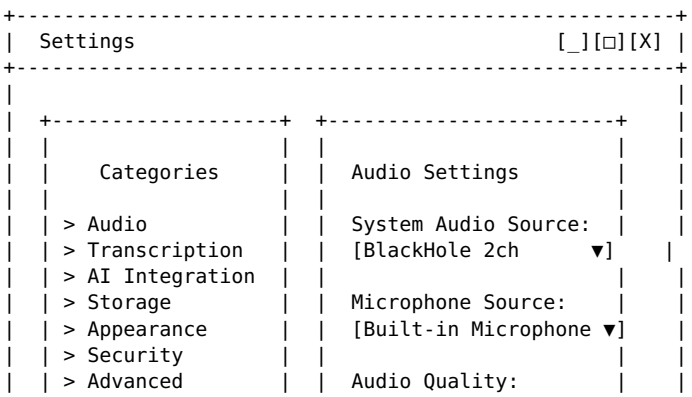
User Interface Design

Wireframes

Main Application Window



Settings Panel



```
|  |                   |  |  [High (16-bit/48kHz) ▼]    |
|  |                   |  |                             |
|  |                   |  |  [ ] Noise Reduction        |
|  |                   |  |  [ ] Echo Cancellation      |
|  |                   |  |                             |
|  |                   |  |  Test Audio Capture:        |
|  |                   |  |  [Test] [Stop]              |
|  |                   |  |                             |
|  +-------------------+  +-----------------------------+
|                                                        |
|                     [Cancel]    [Save]                 |
+--------------------------------------------------------+
```

Meeting Summary View

```
+--------------------------------------------------------+
|  Meeting Summary                              [_][□][X] |
+--------------------------------------------------------+
|  Meeting: Weekly Project Sync                          |
|  Date: April 22, 2025                                  |
|  Duration: 45 minutes                                  |
|  Participants: John, Sarah, Michael, Elena             |
+--------------------------------------------------------+
|                                                        |
|  EXECUTIVE SUMMARY                                     |
|  ---------------------------------------------        |
|  The team discussed Q3 targets and resource            |
|  allocation for the new project. Concerns were         |
|  raised about timeline feasibility, but the team       |
|  agreed to prioritize tasks to meet deadlines.         |
|  ---------------------------------------------        |
|                                                        |
|  KEY POINTS                                            |
|  • Q3 targets need to be finalized by Friday           |
|  • Resource constraints identified in design team      |
|  • New project kickoff scheduled for May 10            |
|  • Budget approval pending from finance                |
|                                                        |
|  ACTION ITEMS                                          |
|  [✓] Review resource allocation (Michael, Apr 24)      |
|  [ ] Finalize Q3 priorities (Team, Apr 26)             |
|  [ ] Schedule budget review (Sarah, Apr 23)            |
|  [ ] Prepare design brief (Elena, Apr 30)              |
|                                                        |
+--------------------------------------------------------+
|  [Full Transcript] [Export to PDF] [Share via Email]   |
+--------------------------------------------------------+
```

User Flow

```
graph TD
    A[Launch App] --> B{First Time?}
    B -->|Yes| C[Setup Wizard]
    B -->|No| D[Main Screen]

    C --> C1[Audio Setup]
    C1 --> C2[Permissions]
    C2 --> C3[Account Login]
    C3 --> C4[Preferences]
    C4 --> D

    D --> E{Meeting Status}
    E -->|No Meeting| F[Standby Mode]
    E -->|Meeting Detected| G[Prompt to Record]

    F --> G
```

```
G -->|User Confirms| H[Recording Mode]
G -->|User Declines| F

H --> I[Live Transcription]
I --> J[AI Processing]

H --> K[User Stops Recording]
K --> L[Generate Summary]
L --> M[Summary View]

M --> N{User Action}
N -->|Edit| O[Edit Summary]
N -->|Export| P[Export Options]
N -->|Share| Q[Share Options]
N -->|New Meeting| D

style A fill:#f9d5e5,stroke:#333,stroke-width:2px
style H fill:#d5e8f9,stroke:#333,stroke-width:2px
style L fill:#e5f9d5,stroke:#333,stroke-width:2px
style M fill:#f9e5d5,stroke:#333,stroke-width:2px
```

Key Features and Controls

1. **Recording Controls**
   - One-click recording start/stop
   - Pause/resume functionality
   - Visual recording indicator
   - Automatic meeting detection (optional)
2. **Transcription Display**
   - Real-time scrolling transcript
   - Speaker identification (when possible)
   - Timestamp markers
   - Search functionality
   - Text highlighting
3. **AI Insights Panel**
   - Live key points extraction
   - Action item identification
   - Decision tracking
   - Topic categorization
   - Sentiment analysis (optional)
4. **Meeting Management**
   - Meeting naming and categorization
   - Calendar integration
   - Meeting history and search
   - Favorites/pinning important meetings
5. **Export and Sharing**
   - Multiple export formats (PDF, DOCX, TXT, HTML)
   - Direct sharing to email, Slack, Teams
   - Selective sharing (full transcript vs. summary only)
   - Access control options
6. **Settings and Configuration**
   - Audio source selection and testing
   - Transcription language and accuracy settings
   - AI customization (summary length, focus areas)
   - Storage and retention policies
   - Appearance customization

Accessibility Considerations

1. **Visual Accessibility**
   - High contrast mode
   - Adjustable text size

    - Screen reader compatibility
    - Keyboard navigation support
    - Color blindness considerations in UI design
2. **Hearing Accessibility**
    - Visual indicators for audio events
    - Transcript as alternative to audio
    - Closed captioning export options
3. **Motor Accessibility**
    - Keyboard shortcuts for all major functions
    - Voice command support (optional)
    - Reduced motion option for animations
4. **Cognitive Accessibility**
    - Clear, consistent interface
    - Simplified mode option
    - Progressive disclosure of complex features
    - Helpful tooltips and contextual help
5. **Language Accessibility**
    - Multi-language support for UI
    - Translation options for transcripts
    - Clear, plain language in all instructions

Implementation Plan

Development Phases and Milestones

Phase 1: Core Functionality (8 weeks)

- **Week 1-2**: Setup development environment and project structure
    - Configure Electron with React
    - Implement basic UI framework
    - Set up build pipeline
- **Week 3-4**: Audio capture implementation
    - Integrate BlackHole for system audio
    - Implement microphone capture
    - Create audio processing pipeline
- **Week 5-6**: Transcription engine integration
    - Implement Whisper AI integration
    - Develop real-time transcription display
    - Create transcript storage system
- **Week 7-8**: Basic UI and testing
    - Implement core UI components
    - Develop settings panel
    - Internal alpha testing

Phase 2: AI Integration and Enhanced Features (6 weeks)

- **Week 9-10**: OpenAI API integration
    - Implement summarization features
    - Develop action item extraction
    - Create insights panel
- **Week 11-12**: Meeting management
    - Implement meeting detection
    - Develop meeting history and search
    - Create export functionality
- **Week 13-14**: Security and corporate integration
    - Implement Microsoft SSO
    - Develop Intune compliance features
    - Security auditing and hardening

Phase 3: Polish and Platform Support (6 weeks)

- **Week 15-16**: UI refinement and accessibility
  - Implement accessibility features
  - Polish user interface
  - User experience improvements
- **Week 17-18**: Cross-platform testing
  - Test on various macOS versions
  - Verify meeting platform compatibility
  - Performance optimization
- **Week 19-20**: Beta release and feedback
  - Limited beta release
  - Gather and implement feedback
  - Final preparations for v1.0

Phase 4: iOS Integration and Advanced Features (Future)

- iOS companion app development
- Advanced AI features (custom training, specialized insights)
- Team collaboration features
- Integration with additional platforms

Testing Strategy

Unit Testing

- Jest for JavaScript components
- Electron-specific component testing
- Audio processing unit tests
- Transcription accuracy validation

Integration Testing

- End-to-end audio capture to transcription flow
- OpenAI API integration testing
- Storage and retrieval testing
- Cross-platform compatibility testing

User Acceptance Testing

- Internal dogfooding program
- Controlled beta with selected users
- Usability testing sessions
- Accessibility compliance testing

Performance Testing

- CPU and memory usage monitoring
- Battery impact assessment
- Transcription latency measurement
- API call optimization

Security Testing

- Penetration testing
- Data privacy audit
- Corporate security compliance verification
- Third-party dependency security scanning

Deployment Considerations

Distribution Methods

- Direct download from website
- Enterprise distribution via MDM
- Future consideration: Mac App Store

Update Mechanism

- Auto-update framework
- Delta updates to minimize bandwidth
- Update notifications with release notes
- Enterprise update control options

Corporate Deployment

- Intune deployment package
- Group Policy configuration options
- Silent installation support
- Enterprise license management

Support Infrastructure

- Documentation and knowledge base
- In-app help system
- Feedback mechanism
- Crash reporting and analytics

Compliance Documentation

- Privacy policy
- Terms of service
- Corporate security compliance documentation
- Data handling and retention policies