LinkedIn AI Assistant - System Diagrams

High-Level System Architecture

mermaid

```
graph TB
    subgraph "User Devices"
        PWA[PWA Mobile App<br/>React + TypeScript]
        EXT[Chrome Extension<br/>React + TypeScript]
    end

    subgraph "Supabase Backend"
        AUTH[Supabase Auth]
        DB[(PostgreSQL Database)]
        STORAGE[Supabase Storage]
        EDGE[Edge Functions]
        REALTIME[Realtime Subscriptions]
    end

    subgraph "External APIs"
        OPENAI[OpenAI API<br/>Whisper + GPT-4]
        LINKEDIN[LinkedIn.com]
    end

    PWA -->|Upload Audio| STORAGE
    PWA -->|Create Records| DB
    PWA -->|Authenticate| AUTH
    EXT -->|Query Data| DB
    EXT -->|Manipulate DOM| LINKEDIN
    EXT -->|Subscribe| REALTIME
    EDGE -->|Transcribe| OPENAI
    EDGE -->|Analyze| OPENAI
    EDGE -->|Generate Responses| OPENAI
    STORAGE -->|Trigger| EDGE
    DB -->|Notify| REALTIME
```

Complete Data Flow: Recording to LinkedIn Action

mermaid

```
sequenceDiagram
    participant User as User (PWA)
    participant QR as QR Scanner
    participant Recorder as Audio Recorder
    participant Storage as Supabase Storage
    participant Edge as Edge Function
    participant AI as OpenAI API
    participant DB as Database
    participant RT as Realtime
    participant Ext as Chrome Extension
    participant LI as LinkedIn DOM

    User->>QR: Scan LinkedIn QR code
    QR->>DB: Create contact record
    DB-->>QR: Contact created

    User->>Recorder: Start recording
    Note over Recorder: Records conversation audio
    User->>Recorder: Stop recording

    Recorder->>Storage: Upload audio file
    Storage-->>Recorder: Upload complete

    Recorder->>DB: Create conversation record
    DB->>Edge: Trigger process-audio function

    Edge->>Storage: Download audio file
    Storage-->>Edge: Audio file

    Edge->>AI: Transcribe audio (Whisper)
    AI-->>Edge: Transcription text

    Edge->>AI: Analyze conversation (GPT-4)
    AI-->>Edge: Analysis (summary, topics, etc)

    Edge->>DB: Update conversation + contact
    DB->>RT: Notify changes

    RT->>Ext: Push update
    Ext->>AI: Generate connection message
    AI-->>Ext: Personalized message

    Ext->>LI: Click Connect button
    Ext->>LI: Insert message text
    Ext->>LI: Click Send button
```

```
    Ext->>DB: Log AI action
```

Auto-Pilot Message Response Flow

mermaid

```
sequenceDiagram
    participant LI as LinkedIn Page
    participant CS as Content Script
    participant BG as Background Worker
    participant DB as Supabase DB
    participant Edge as Edge Function
    participant AI as OpenAI API

    Note over LI: Contact sends message
    LI->>CS: DOM mutation detected
    CS->>CS: Extract sender & message

    CS->>BG: New message event
    BG->>DB: Check auto-pilot status
    DB-->>BG: Contact info + auto_pilot=true

    BG->>Edge: Call generate-response
    Edge->>DB: Fetch conversation context
    DB-->>Edge: Conversation summary
    Edge->>DB: Fetch message history
    DB-->>Edge: Last 10 messages

    Edge->>AI: Generate response with context
    AI-->>Edge: AI-generated message

    Edge->>DB: Log AI action
    Edge->>DB: Save messages
    Edge-->>BG: Return AI message

    BG-->>CS: Send message to LinkedIn
    CS->>LI: Insert text in message box
    CS->>LI: Trigger input event
    CS->>LI: Click send button

    Note over LI: Message sent automatically
```

Database Entity Relationship Diagram

mermaid

```
erDiagram
    users ||--o{ contacts : "has"
    users ||--o{ conversations : "records"
    users ||--o{ ai_actions : "performs"
    users ||--o{ linkedin_messages : "exchanges"

    contacts ||--o{ conversations : "involved_in"
    contacts ||--o{ ai_actions : "receives"
    contacts ||--o{ linkedin_messages : "exchanges"

    users {
        uuid id PK
        string email
        timestamp created_at
    }

    contacts {
        uuid id PK
        uuid user_id FK
        string name
        string linkedin_url
        jsonb linkedin_profile_data
        string relationship_type
        text custom_instructions
        boolean auto_pilot_enabled
        string status
        timestamp created_at
        timestamp updated_at
    }

    conversations {
        uuid id PK
        uuid user_id FK
        uuid contact_id FK
        string audio_url
        text transcription
        text summary
        text[] key_topics
        jsonb ai_analysis
        integer duration_seconds
        timestamp recorded_at
        boolean processed
    }

    ai_actions {
        uuid id PK
```

```
    uuid user_id FK
    uuid contact_id FK
    string action_type
    text action_content
    boolean success
    jsonb metadata
    timestamp created_at
}

linkedin_messages {
    uuid id PK
    uuid user_id FK
    uuid contact_id FK
    string sender
    text message_text
    boolean ai_generated
    timestamp sent_at
}
```

User Flow: Record Conversation

mermaid

```
flowchart TD
    A[User opens PWA] --> B{Authenticated?}
    B -->|No| C[Login/Signup Screen]
    C --> D[Enter credentials]
    D --> E[Authenticate with Supabase]
    E --> F[Main Recording Screen]
    B -->|Yes| F

    F --> G[Tap 'Scan QR Code']
    G --> H[Camera opens]
    H --> I{QR Code detected?}
    I -->|No| H
    I -->|Yes| J[Extract LinkedIn URL]
    J --> K[Create contact in DB]
    K --> L[Display contact info]

    L --> M[Tap 'START RECORDING']
    M --> N[Recording in progress<br/>Show duration & waveform]
    N --> O[Tap 'STOP RECORDING']

    O --> P[Create audio Blob]
    P --> Q[Upload to Supabase Storage]
    Q --> R[Create conversation record]
    R --> S[Show 'Processing' screen]

    S --> T[Edge Function processes]
    T --> U[Whisper transcribes]
    U --> V[GPT-4 analyzes]
    V --> W[Update DB with results]

    W --> X[Show Success Screen]
    X --> Y{User action}
    Y -->|View Details| Z[Full conversation view]
    Y -->|Record Another| F
    Y -->|Done| AA[Close app]
```

User Flow: Auto Connection Request

```
mermaid
```

```
flowchart TD
    A[User opens LinkedIn] --> B[Extension loads]
    B --> C[Sidebar shows contacts list]
    C --> D{New contact from PWA?}

    D -->|No| E[Display existing contacts]
    D -->|Yes| F[Contact marked as 'NEW']

    F --> G[Extension detects new contact]
    G --> H[Call send-connection-request API]
    H --> I[GPT-4 generates message ≤300 chars]
    I --> J[Receive personalized message]

    J --> K[Content script activates]
    K --> L[Navigate to LinkedIn profile]
    L --> M[Find 'Connect' button]
    M --> N[Click Connect]

    N --> O[Modal opens]
    O --> P[Click 'Add a note']
    P --> Q[Insert AI message]
    Q --> R[Click 'Send']

    R --> S[Update DB: status = 'connection_sent']
    S --> T[Log AI action]
    T --> U[Update sidebar UI]
    U --> V[Show 'Connection Sent' badge]
```

User Flow: Auto-Pilot Response

mermaid

```
flowchart TD
    A[User browsing LinkedIn] --> B[Extension monitoring]
    B --> C{New message detected?}
    C -->|No| B
    C -->|Yes| D[Extract message data]

    D --> E[Query DB for contact]
    E --> F{Contact exists?}
    F -->|No| G[Ignore message]
    F -->|Yes| H{Auto-pilot enabled?}

    H -->|No| G
    H -->|Yes| I[Save incoming message to DB]

    I --> J[Call generate-response API]
    J --> K[Fetch conversation context]
    K --> L[Fetch message history]
    L --> M[Build GPT-4 prompt with context]
    M --> N[GPT-4 generates response]

    N --> O[Receive AI message]
    O --> P[Content script finds message box]
    P --> Q[Insert AI text]
    Q --> R[Trigger input event]
    R --> S[Wait 500ms]
    S --> T[Click send button]

    T --> U[Message sent on LinkedIn]
    U --> V[Save AI message to DB]
    V --> W[Log AI action]
    W --> X[Update sidebar UI]
    X --> Y[Show notification:<br/>'AI responded to Contact']
```

Component Architecture: PWA

mermaid

```
graph TD
    A[App.tsx<br/>Root Component] --> B[Router]

    B --> C[LoginScreen]
    B --> D[RecordingScreen]
    B --> E[SuccessScreen]

    D --> F[QRScanner<br/>Component]
    D --> G[ContactDisplay<br/>Component]
    D --> H[RecordButton<br/>Component]
    D --> I[AudioWaveform<br/>Component]
    D --> J[ProcessingScreen<br/>Component]

    D --> K[useAudioRecorder<br/>Hook]
    D --> L[useQRScanner<br/>Hook]
    D --> M[useAuth<br/>Hook]

    K --> N[Supabase Client]
    L --> N
    M --> N

    C --> M
    E --> N
```

Component Architecture: Chrome Extension

```
mermaid
```

```
graph TD
    A[Chrome Extension] --> B[Sidebar<br/>React App]
    A --> C[Content Script<br/>linkedin.ts]
    A --> D[Background Worker<br/>service-worker.ts]

    B --> E[ContactList<br/>Component]
    B --> F[ContactDetail<br/>Component]

    E --> G[ContactCard<br/>Sub-component]
    F --> H[AutoPilotToggle<br/>Sub-component]
    F --> I[MessageHistory<br/>Sub-component]
    F --> J[AIActionLog<br/>Sub-component]

    B --> K[useContacts Hook]
    B --> L[useRealtime Hook]
    K --> M[Supabase Client]
    L --> M

    C --> N[MutationObserver]
    C --> O[DOM Manipulation]

    D --> M
    D --> P[API Coordination]

    C -.Message.-> D
    D -.Response.-> C
    B -.Query.-> M
    M -.Realtime.-> B
```

LinkedIn DOM Manipulation Flow

mermaid

```
flowchart LR
    A[MutationObserver] --> B{Change detected?}
    B -->|Yes| C[Check if new message]
    B -->|No| A

    C --> D{Is incoming message?}
    D -->|No| A
    D -->|Yes| E[Extract message data]

    E --> F[Get sender LinkedIn URL]
    E --> G[Get message text]
    E --> H[Get timestamp]

    F --> I[Send to background script]
    G --> I
    H --> I

    I --> J[Check auto-pilot status]
    J --> K{Auto-pilot ON?}
    K -->|No| L[Do nothing]
    K -->|Yes| M[Generate AI response]

    M --> N[Find message input box]
    N --> O[Clear existing content]
    O --> P[Insert AI text]
    P --> Q[Trigger input event]
    Q --> R[Wait 500ms]
    R --> S[Find send button]
    S --> T[Click send]
    T --> U[Message sent!]
```

Data Sync Between PWA and Extension

```
mermaid
```

```
sequenceDiagram
    participant PWA as PWA Mobile App
    participant DB as Supabase Database
    participant RT as Realtime Channel
    participant Ext as Chrome Extension

    Note over PWA: User records conversation
    PWA->>DB: Create conversation record
    PWA->>DB: Upload audio file

    Note over DB: Edge Function processes
    DB->>DB: Update conversation.processed = true
    DB->>DB: Update contact with summary

    DB->>RT: Broadcast changes
    RT->>Ext: Push notification

    Note over Ext: Extension receives update
    Ext->>DB: Fetch latest contact data
    DB-->>Ext: Return contact + conversation

    Note over Ext: UI updates automatically
    Ext->>Ext: Show new contact in list
    Ext->>Ext: Mark as 'NEW'

    Note over Ext: Auto-actions may trigger
```

Edge Function Processing Pipeline

mermaid

```
flowchart TD
    A[Audio file uploaded] --> B[Trigger process-audio function]

    B --> C[Download audio from Storage]
    C --> D[Send to Whisper API]
    D --> E[Receive transcription]

    E --> F[Build GPT-4 prompt]
    F --> G[Include transcription]
    F --> H[Request: summary, topics, sentiment]

    G --> I[Call GPT-4 API]
    H --> I
    I --> J[Receive AI analysis]

    J --> K[Parse JSON response]
    K --> L[Update conversation table]
    K --> M[Update contact table]

    L --> N[Set processed = true]
    M --> O[Set suggested relationship]

    N --> P[Trigger Realtime notification]
    O --> P

    P --> Q[Chrome Extension receives update]
```