# CHAPTER 4

# PROJECT DESIGN

## 4.1. DEFINITION:

The Callify video conferencing application is designed to provide a seamless and secure platform for video calls and meetings. Below is a detailed overview of its project design, including features, architecture, user interface, and technology stack.

## 4.2. Overview of Callify:

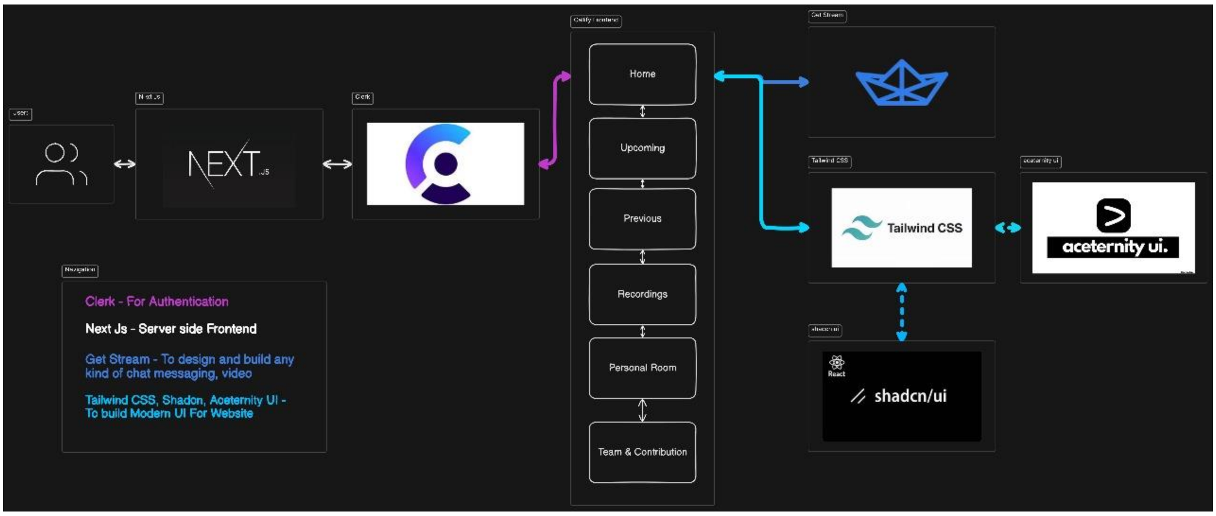

**Figure 4.2.1 - Dataflow Diagram**

- **Purpose**: Callify aims to facilitate hassle-free video calls and meetings, enabling users to connect securely from anywhere.
- **Target Audience**: Businesses, educational institutions, and individuals seeking reliable video communication tools.

## 4.3. Key Features:

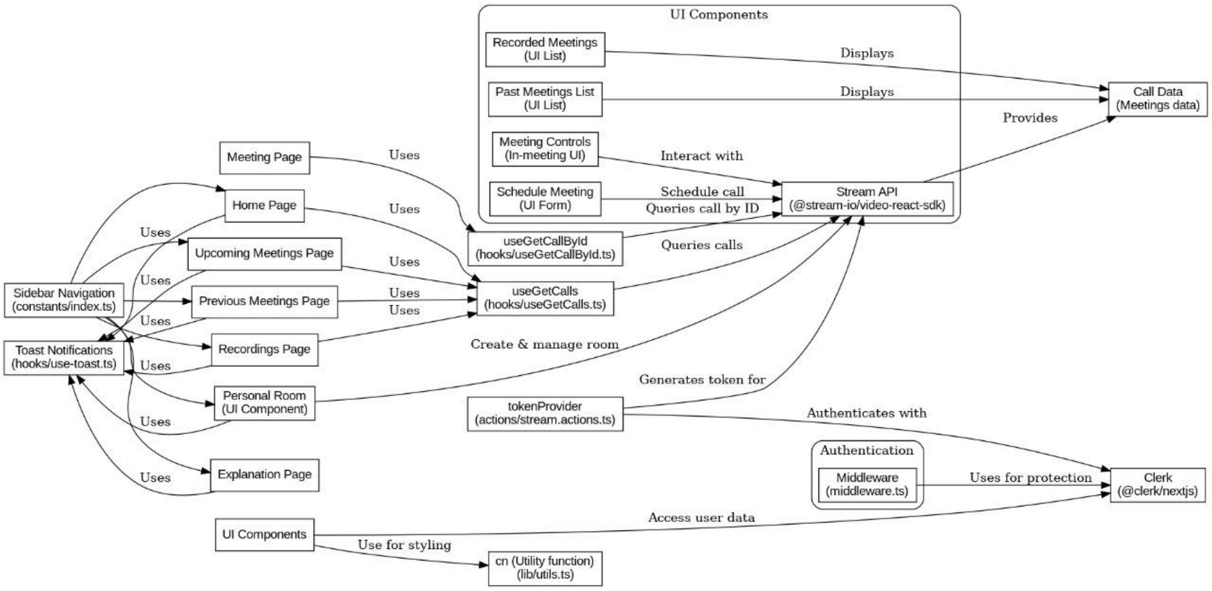

**Figure 4.3.1 – E-R Diagram**

- **User Authentication**:
  - Secure login via social sign-on or traditional email/password methods using Clerk.

- **Meeting Management**:
  - **New Meeting**: Users can quickly start a new meeting with configurable camera and microphone settings.
  - **Schedule Future Meetings**: Users can input meeting details (date, time) and access them on an 'Upcoming Meetings' page.
  - **Past Meetings List**: Access to previously held meetings with details and metadata.

- **Meeting Controls**:
  - Participants can manage various aspects of the meeting, including:
    - Recording
    - Screen sharing
    - Muting/unmuting
    - Emoji reactions
    - Participant management (pinning, muting, blocking)

- **Personal Room**:
    - Each user has a unique meeting link for instant meetings, which can be shared easily.
- **Real-time Functionality**:
    - All interactions are secure and occur in real-time, ensuring user privacy and data integrity.
- **Responsive Design**:
    - The application is designed to be responsive, providing an optimal user experience across devices.

## 4.4. Technology Stack

- **Frontend**:
    - **Next.js**: For server-side rendering and static site generation.
    - **TypeScript**: For type safety and better development experience.
    - **Tailwind CSS**: For styling and responsive design.
    - **Clerk**: For user authentication and session management.
    - **GetStream**: For real-time video capabilities.

- **Backend**:
    - **Node.js**: For server-side logic and API handling.
    - **Database**: A suitable database for storing user data, meeting details, and recordings.

## 4.5. User Interface Design:

- **Landing Page**:
    - Clear value proposition and call-to-action buttons for signing up or logging in.

- **Dashboard**:
    - Overview of upcoming meetings, past meetings, and options to create or join meetings.

- **Meeting Interface**:
    - Intuitive layout with easy access to controls (mute, video on/off, screen share).
    - Chat functionality for participants to communicate during meetings.
- **Modals**:
    - For creating new meetings, joining existing meetings, and viewing meeting details.

## 4.6. Security Measures:

- **Data Encryption**: All data transmitted during meetings is encrypted to protect user privacy.
- **Access Control**: Role-based access control to manage permissions for different participants.
- **Secure Authentication**: Use of Clerk for secure user authentication and session management.

## 4.7. Development and Deployment:

- **Development Process**:
    - Agile methodology with iterative development cycles.
    - Regular user testing to gather feedback and improve the application.
- **Deployment**:
    - Use of Vercel for deploying the Next.js application.
    - Continuous integration and deployment (CI/CD) practices to ensure smooth updates.

## 4.8. Future Enhancements:

- **AI Integration**: Implement AI features such as transcription, facial recognition, and virtual backgrounds.
- **Mobile Application**: Develop a mobile version of the application for iOS and Android platforms.
- **Advanced Analytics**: Provide users with insights into meeting participation and engagement.