**Table of Contents**

**1. Introduction**

1.1 Overview 4

1.2 Purpose 4

1.3 Scope 4

1.4 Outline of SRS 4

1.5 References 4

1.6 Definitions, Acronyms and Abbreviations 5

**2. Overall description**

2.1 Product Perspective 7

2.2 Product Function 7

2.3 User Classes and Characteristics 7

2.4 User Documentation 7

2.5 Operating Environment 8

2.6 Assumptions and Dependencies 8

2.7 Design Implementation 8

**3. Specific Requirements**

3.1 Functional Requirements 9

3.2 Non-Functional Requirements 10

3.2.1 Performance 10

3.2.2 Security 10

3.2.3 Reliability 11

3.2.4 Maintainability 11

3.3 Constraints 11

**4. External Interfaces**

4.1 User Interface 12

4.1.1 Home Page User Interface 12

4.1.2 Login Interface for User 13

4.1.3 Customer’s Login Interface 13

4.1.4 Login interface for Seller 14

4.1.5 Seller’s Login Interface 14

4.2 Hardware Interface 15

4.3 Communication Interface 15

4.4 Analytics Interface 15

**5. Software Design Specification**

5.1 Data Flow Diagram 15

5.2 Unified Modelling Language 21

5.2.1 Use Case Diagram 21

5.2.2 Sequence Diagram 22

5.2.3 Class diagram 26

5.2.4 Component Diagram 27

5.2.5 Activity Diagram 28

5.2.6 State Chart Diagram 30

5.2.7 Deployment Diagram 31

5.3 Database design

5.3.1 Table Design 32

5.3.2 Entity Relationship Diagram for User 35

5.3.3 Entity Relationship Diagram for Seller 36

**6. Testing of Software**

6.1 Unit testing 37

6.2 Integration testing 37

6.3 Validation Testing 37

6.4 Test Cases 38

6.4.1 Objectives 38

6.4.2 Description 38

6.4.3 Features to be tested 39

6.5 Performance Testing 44

6.6 Suspension and Resumption Requirements 44

6.6.1 Suspension Criteria 44

6.6.2 Resumption Requirements 44

**1. Introduction**

**1.1 Purpose**

This document specifies the requirements for developing a video conferencing application (Connectify) modeled after Zoom. The application aims to provide users with a platform for virtual meetings, webinars, and online collaboration, file sharing, meeting recordings, screen sharing.

**1.2 Scope**

The system will support functionalities for user registration, video and audio conferencing, real-time chat, screen sharing, meeting scheduling ,participant details, File sharing system, meeting recording and management. It will also include features for recording meetings, managing users, and handling different user roles.

**1.3 Definitions, Acronyms, and Abbreviations**

* Connectify: The video conferencing application being developed.
* UI: User Interface
* DB: Database
* API: Application Programming Interface
* RTC: Real-Time Communication
* SSL: Secure Sockets Layer
* P2P: Peer-to-Peer

**1.4 References**

* Zoom API Documentation
* Socket.io for real-time video and audio communication
* WebRTC for ream-time communication
* Node.js and Express.js for server-side development
* MongoDb for database management
* HTML,CSS, JavaScript and EJS view engine for front-end development

**2. Overall Description**

**2.1 Product Perspective**

The Connectify will be a video conferencing platform utilizing modern technologies to provide a reliable and interactive communication experience through video conferencing. It will employ Node.js and Express.js for backend operations, MongoDb for database management, and HTML, CSS and EJS view engine for the user interface.

**2.2 Product Functions**

* User Registration and Authentication: Allow users to create accounts and log in.
* Video and Audio Conferencing: Facilitate real-time video and audio communication.
* Screen Sharing: Enable users to share their screens with meeting participants.
* Group audio and video calling: Multiple people can connect through this feature.
* Chat: Provide real-time text communication during meetings.
* Meeting Scheduling: Allow users to schedule and manage meetings.
* Meeting Recording: Record meetings for later playback.
* User Management: Manage user roles and permissions.
* Admin Dashboard: Provide administrative access to manage users, meetings, and settings.

**2.3 User Characteristics**

* Admin: Full access to all system functionalities, including user and meeting management.
* Host: Users who schedule and manage meetings, with control over meeting settings and participant permissions.
* Participant: Users who join meetings and interact with other participants.
* Guest: Users who can join meetings without creating an account (if guest access is supported).

**2.4 Constraints**

* Ensure compliance with data protection and privacy regulations.
* The application must be mobile-responsive and compatible with major browsers.
* Real-time communication must be low-latency and reliable.

**3. Specific Requirements**

**3.1 Functional Requirements**

3.1.1 User Account Management

* Registration and Login: Users must be able to create accounts, log in, and reset passwords. Support for social media logins (e.g., Google, Facebook) should be considered.
* Profile Management: Users should be able to update their profiles, including name, email, and profile picture.

3.1.2 Meeting Management

* Scheduling: Users should be able to schedule one-time or recurring meetings with specific date, time, and duration.
* Meeting Creation: Users can create instant meetings and invite participants via email or direct links.
* Meeting Reminders: Automatic email and/or push notifications should be sent to participants before the meeting starts.
* Meeting Link: A unique URL should be generated for each meeting, which users can share with participants.

3.1.3 Video and Audio Communication

* HD Video: Support for high-definition video (720p or higher).
* Audio: High-quality audio with echo cancellation, noise suppression, and adaptive audio quality based on network conditions.
* Virtual Backgrounds: Users should be able to choose or upload virtual backgrounds.

3.1.4 Participant Management

* Roles and Permissions: Define roles such as host, co-host, and participant, with specific permissions (e.g., mute/unmute, video on/off).
* Participant Controls: Hosts should have the ability to mute/unmute participants, enable/disable video, and remove participants from the meeting.

3.1.5 Screen Sharing

* Full Screen and Application Sharing: Users can share their entire screen or specific applications with meeting participants.
* Annotation Tools: Provide tools for annotating shared screens (e.g., drawing, highlighting).

3.1.6 Chat and Collaboration

* In-Meeting Chat: Real-time text chat available to all participants.
* File Sharing: Ability to share files within the chat or during the meeting.
* Whiteboard: Interactive whiteboard feature for collaborative drawing and note-taking.

3.1.7 Recording and Playback

* Meeting Recording: Ability to record meetings, including video, audio, and shared screens.
* Cloud Storage: Store recordings in the cloud with secure access.
* Playback: Users can access and play back recorded meetings.

3.1.8 Security and Privacy

* End-to-End Encryption: Ensure video and audio communications are encrypted.
* Meeting Security: Implement features like waiting rooms, password protection, and meeting locks.
* User Authentication: Secure login methods, including multi-factor authentication (MFA).

3.1.9 Integration and APIs

* Calendar Integration: Integration with popular calendars (e.g., Google Calendar, Outlook) for scheduling and reminders.
* API Access: Provide APIs for integrating with third-party applications and services.

**3.2. Non-Functional Requirements**

3.2.1 Performance

* Scalability: The system must handle a high number of concurrent meetings and participants with minimal latency.
* Latency: Ensure low latency for real-time communication to provide a smooth user experience.

3.2.2 Reliability

* Availability: System must have high availability and redundancy to handle failures and maintain uptime.
* Backup: Regular backups of user data and meeting recordings to prevent data loss.

3.2.3 Usability

* User Interface: Intuitive and user-friendly interface for ease of use.
* Accessibility: Ensure the platform is accessible to users with disabilities, including screen reader support and keyboard navigation.

3.2.4 Security

* Data Protection: Compliance with data protection regulations such as GDPR or CCPA.
* Vulnerability Management: Regular security assessments and updates to address potential vulnerabilities.

3.2.5 Compatibility

* Cross-Platform Support: The application must work on various platforms, including web browsers, mobile devices (iOS and Android), and desktop applications.
* Browser Compatibility: Ensure compatibility with major web browsers (e.g., Chrome, Firefox, Safari).

3.2.6 Documentation

* User Guides: Provide comprehensive user guides and help documentation.
* API Documentation: Detailed API documentation for developers integrating with the platform.

**4. Additional Considerations**

**4.1. Video Conferencing Features**:

* Real-Time Communication: Support for HD video and audio, real-time chat, and virtual backgrounds.
* Meeting Management: Scheduling, recurring meetings, and meeting reminders.
* Participant Management: Host controls for muting, video enabling/disabling, and participant management.

**4.2. Collaboration Tools**:

* Screen Sharing: Options for sharing entire screen, specific applications, or files.
* Whiteboard: Interactive whiteboard for drawing and note-taking during meetings.
* File Sharing: Ability to share files within meetings and during chat.

**4.3. Security and Privacy**:

* Encryption: End-to-end encryption for video and audio communications.
* Authentication: Secure login methods including single sign-on (SSO) and multi-factor authentication (MFA).
* Meeting Security: Features like waiting rooms, meeting locks, and participant authentication.

**4.4. Administrative Controls**:

* Dashboard: Admin tools for user management, meeting analytics, and system configuration.
* User Roles: Differentiated access levels for hosts, co-hosts, and participants.

4.5. **Integration Capabilities**:

* API Integration: Support for third-party integrations and plugins.
* Calendar Integration: Synchronization with popular calendar applications (e.g., Google Calendar, Outlook).

4.6. **User Experience Enhancements**:

* Breakout Rooms: Ability to create and manage breakout rooms during meetings.
* Recording: Features for recording meetings and storing recordings securely.

4.7. **Performance Requirements**:

* Scalability: System must handle a large number of concurrent meetings and users.
* Latency: Low-latency performance for smooth real-time communication.

4.8. **Compliance**:

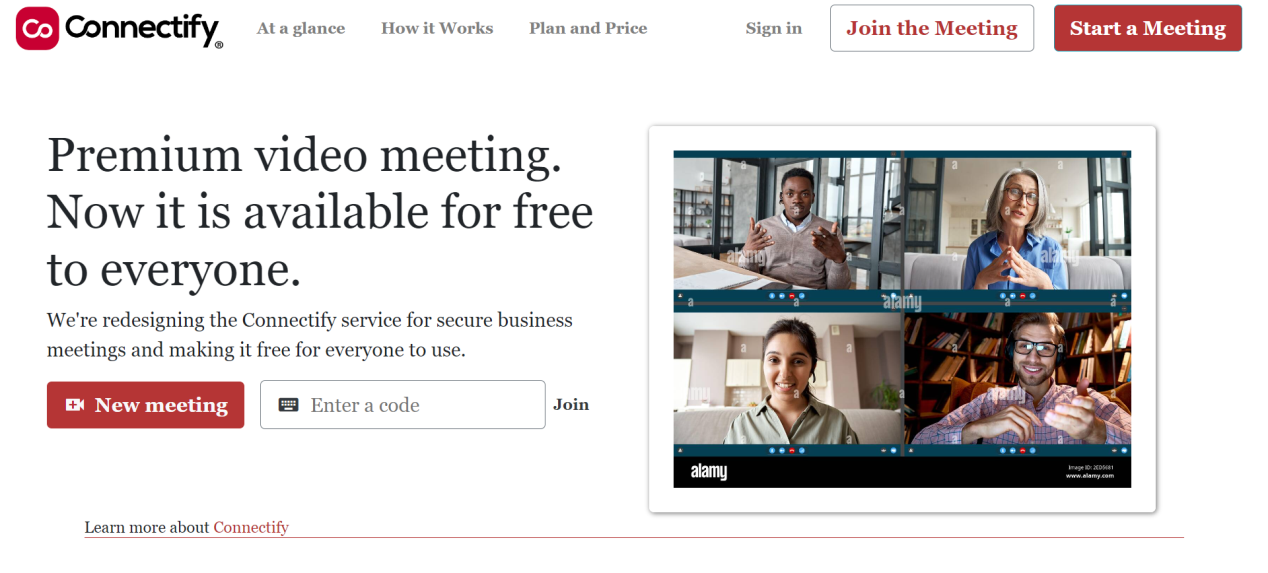
* Regulations: Adherence to data protection regulations (e.g., GDPR, CCPA) for user data.

**5. Interface Requirements**

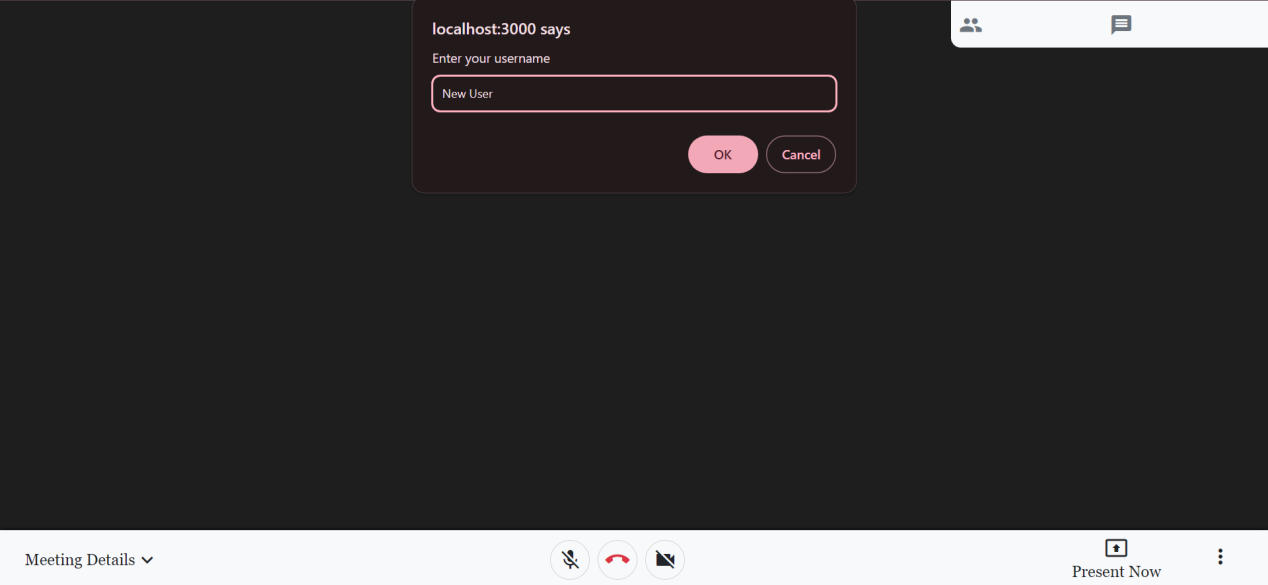
**5.1. User Interfaces**

5.1.1 Web and Mobile Interfaces

* Login/Registration Screen:
  + Elements: Email/username, password fields, login button, registration link, password recovery link.
  + Validation: Real-time feedback for invalid input, such as incorrect email format or weak passwords.
* Dashboard:
  + Elements: Navigation menu (Schedule Meeting, Join Meeting, Settings), upcoming meetings list, meeting creation button.
  + Functionality: Ability to view, schedule, and join meetings from a central location.



* Meeting Interface:
  + Elements: Video display, audio controls (mute/unmute), video controls (start/stop video), participant list, chat window, screen sharing button.
  + Functionality: Real-time video and audio communication, screen sharing, chat, and meeting control features (e.g., end meeting, record).



* Settings and Profile Management:
  + Elements: Profile picture upload, personal information fields (name, email), notification settings, account security options.
  + Functionality: Users can update their profile details, change passwords, and configure notification preferences.

5.1.2 Admin Interface

* Admin Dashboard:
* Elements: Overview of system performance, active users, recent meetings, and system alerts.
* Functionality: Monitoring system status, user management, and access to logs and reports.
* User Management:
  + Elements: List of users, user roles, search and filter options.
  + Functionality: Admins can add, remove, or modify user roles and permissions.
* Meeting Management:
  + Elements: List of scheduled meetings, meeting logs, and analytics.
  + Functionality: View and manage meetings, including rescheduling or canceling.

**5.2. System Interfaces**

5.2.1 Backend Services

* Authentication Service:
  + Interface: RESTful API endpoints for user login, registration, and password recovery.
  + Data: User credentials, authentication tokens.
* Video and Audio Service:
  + Interface: Real-time communication protocols (e.g., WebRTC) for handling video and audio streams.
  + Data: Video/audio streams, participant metadata.
* Meeting Management Service:
  + Interface: API endpoints for scheduling, updating, and retrieving meeting details.
  + Data: Meeting schedules, participant lists, meeting links.

5.2.2 Database

* User Data:
* Interface: Database schema for storing user profiles, login credentials, and account settings.
* Data: User information, authentication tokens, meeting history.
* Meeting Data:
  + Interface: Database schema for storing meeting details, schedules, and recordings.
  + Data: Meeting metadata, participant lists, meeting recordings.
  + Chat and Collaboration Data:
    - Interface: Database schema for storing chat messages and shared files.
    - Data: Chat logs, file metadata.

**5.3. External Interfaces**

5.3.1 Third-Party Integrations

* Calendar Integration:
  + Interface: API endpoints for integrating with calendar services (e.g., Google Calendar, Microsoft Outlook).
  + Data: Meeting schedules, event reminders.
* Payment Gateway (If applicable):
* Interface: API endpoints for processing payments and managing subscriptions.
* Data: Payment details, transaction history.
* Content Delivery Network (CDN):
* Interface: Integration for streaming video content and handling high-traffic scenarios.
* Data: Video content delivery, load balancing.

5.3.2 API Integration

* Public API:
* Interface: RESTful API endpoints for third-party developers to integrate with the platform.
* Data: Meeting creation, user management, and reporting.

**5.4. Usability Considerations**

5.4.1 Responsive Design

* Web Interface: Must be responsive and adapt to various screen sizes (desktops, tablets, smartphones).
* Mobile Interface: Should provide a user-friendly experience optimized for touch interfaces.

5.4.2 Accessibility

* Web Interface: Compliance with web accessibility standards (e.g., WCAG) to support users with disabilities.
* Mobile Interface: Support for screen readers and other assistive technologies.

**6. System design specifications**

**6.1. Data flow Diagram (DFD):**

A Data Flow Diagrams is a structured analysis and design tool that can be used for flowcharting inplace of, or in association with, information-oriented and process-oriented systems flowcharts. A DFD is a network that describes the flow of data and the processes that change, or transform, data throughout a system. This network is constructed by using a set of symbols that do not imply a physical implementation. It has the purpose of clarifying system requirements and identifying major transformations that will become programs in system design. So it is the starting point of the design phase that functionality decomposes the requirement specifications down to the lowest level of detail.

6.1.1 Data flow diagram symbol:

|  |  |
| --- | --- |
| **Symbol** | **Description** |
|  | **Data Flow:** Data flow are pipelines through the packets of information flow. |
|  | **Process:** A Process or task performed by the system |
|  | **Entity:** Entity are object of the system. A source or destination data of a system |
|  | **Data Store**: A place where data to be stored |

6.1.2 Context level DFD – 0 level:

The context level data flow diagram (DFD) is describe the whole system. The 0-level DFD describe the all user module who operate the system. Below data flow diagram of video confrencng site (Connectify), shows the two user can operate the system Admin and Member user.