Requirement Gathering and Analysis Phase
Solution Architecture

Date	22 June 2024
Team ID	PNT2022TMIDSWTID1721142115
Project Name	Project – Video Conferencing Application
Maximum Marks	3 marks

Solution Architecture:

This solution Architecture facilitates communication and collaboration between geographically dispersed individuals or teams through real-time audio and video and document sharing.

The technology Solution for this could be a video conferencing application that leverages a combination of technologies to achieve this goal.

Here's a breakdown of the key components:

- **Client Application:** This is the user-facing software installed on desktops, laptops, or mobile devices. It captures audio and video input, encodes it, and transmits it to the server. It also receives and decodes video streams from other participants, displaying them on the user's screen. WebRTC is a popular open-source protocol for real-time audio and video communication used in many video conferencing apps.
- **Signaling Server:** This server manages communication between clients. It facilitates connection establishment, participant discovery, and exchange of session information.
- **Media Server:** There are two main architectures for media handling:
    - **Selective Forwarding Unit (SFU):** This server acts as a central hub. Clients send their media streams to the SFU, which then selectively forwards each stream to all other participants in the conference. This reduces bandwidth consumption for clients as they only send one stream instead of multiple.
    - **Multipoint Control Unit (MCU):** This server receives all media streams from participants, mixes them (combining audio and video), and sends a single, optimized stream back to each participant. This approach offers more control over the video layout and features like recording, but requires more server processing power.
- **Database:** Stores user information, conference details, recordings (optional), and other application data.

Solution Characteristics:

- **Scalability:** The architecture should be able to handle a growing number of concurrent users and conferences.
- **Reliability:** The system should be resilient to failures and ensure uninterrupted communication.
- **Security:** Encryption of audio and video streams is crucial to protect sensitive information. Secure authentication and authorization mechanisms are also needed for user access control.
- **Performance:** The application should deliver high-quality audio and video with minimal latency, even on limited bandwidth connections.

Features:

User Authentication and Management

- Users can register and log in using their email or social media accounts.
- User roles and permissions are defined (e.g., admin, host, participant).

Video/Audio Communication

- WebRTC is used for real-time video and audio communication.
- STUN/TURN servers are implemented for NAT traversal.

Screen Sharing and Collaboration

- Users can share their screen with other participants.
- Collaboration tools like whiteboarding and document sharing are integrated.

Chat Functionality

- Real-time chat is implemented using WebSocket.
- Messages are stored in the database for future reference.

Meeting Scheduling and Invitations

- Users can schedule meetings and send invitations via email.
- Calendar integration allows automatic meeting creation in the user's calendar.

Recording and Playback

- Meetings can be recorded and stored in cloud storage.
- Recorded sessions are accessible for playback through the application.

Development Phases:

- **Planning and Analysis:** Define functional and non-functional requirements, identify target platforms, and choose appropriate technologies.
- **Design and Development:** Develop the client application, signaling server, media server (SFU or MCU), and database components. Implement security measures and user management features.
- **Testing and Deployment**: Thoroughly test the application for functionality, performance, and security. Deploy the app to chosen platforms (web, mobile) and integrate with any relevant services (calendar, recording).

Deployment and Maintenance

Continuous Integration/Continuous Deployment (CI/CD)

- Use Jenkins or GitHub Actions for automated testing and deployment.
- Docker containers for consistent environment setup across development, testing, and production.

Monitoring and Logging

- Implement monitoring tools like Prometheus and Grafana for real-time system health monitoring.
- Use ELK Stack (Elasticsearch, Logstash, Kibana) for logging and analysis.

Scalability and Load Balancing

- Utilize auto-scaling groups and load balancers (e.g., AWS Elastic Load Balancer) to manage traffic and ensure high availability.
- Implement CDN (Content Delivery Network) for faster content delivery.

Deliverables:

- Functional video conferencing application for desktops, laptops, and/or mobile devices.
- Server-side infrastructure for managing user connections, media streams, and data storage.
- Documentation outlining system architecture, user guides, and deployment instructions.

Architecture Diagram: