

Faculty: Cyber Physical Systems

Department: Internet of Things and Robotics Engineering.

Course Name: Software Engineering Lab

Course Code: ICT 4354

Submitted To:

Shifat Ara Rafiq

Lecturer,

Department: Software Engineering.

Submitted By:

| Tanvir Ahmed | Id: 2001028 |
| --- | --- |
| Akhtarujjaman Siddique | Id: 2001031 |
| Sheikh Sidratul Muntaha Punno | Id: 2001047 |

**Project Name: Meeting app Application Using Flutter, Firebase**

### **1. System Requirements**

**Hardware Requirements:**

* **Client Device:** Smartphones, tablets, or PCs with internet access.
* **Processor:** Minimum 1.5 GHz dual-core CPU.
* **RAM:** Minimum 2 GB RAM for mobile devices, 4 GB for desktops/laptops.
* **Storage:** 100 MB of free space for app installation.
* **Camera and Microphone:** Access to a functioning camera and microphone for video and audio communication.

**Software Requirements:**

* **Operating System:**
  + Android 5.0 (Lollipop) and above for mobile devices.
  + iOS 10.0 and above for Apple devices.
  + Windows 7, macOS 10.12, or Linux-based systems for desktops.
* **Development Environment:**
  + Flutter SDK 3.x
  + Dart 2.x
* **Libraries/Frameworks:**
  + Jitsi Meet SDK for video conferencing functionality.
  + Firebase (optional) for authentication and data storage.

**Network Requirements:**

* **Internet Connection:** A stable internet connection with a minimum bandwidth of 512 kbps for video conferencing.

### 

### 

### **2. Functional Requirements**

**1. User Registration and Login:**

* Users must be able to sign up and log in using email accounts.

**2. Create/Join Meeting:**

* Users can create or join a meeting using a unique meeting ID.
* Option to password-protect meetings.

**3. Video and Audio Conferencing:**

* Real-time video and audio communication during meetings.
* Ability to mute/unmute the microphone and turn the camera on/off.

**4. Screen Sharing:**

* Users can share their screens during the meeting.

**5. Chat Functionality:**

* A live chat feature to allow text-based communication during the meeting.

**6. Meeting Controls:**

* Host controls to manage participants (e.g., muting participants, removing users).
* Ability to lock meetings and prevent new participants.

**7. Participant List:**

* Display the list of participants currently in the meeting.

**8. Raise Hand Feature:**

* Participants should have a "raise hand" feature to signal that they want to speak, which can help manage large meetings.

### **3. Non-Functional Requirements**

**1. Performance:**

* The app must load within 2-5 seconds on all supported devices.
* The video stream should maintain minimal latency (less than 200 ms) under optimal network conditions.

**2. Scalability:**

* The system must support meetings with up to 50 participants with smooth audio/video performance.
* The backend infrastructure must be able to handle growing user traffic.

**3. Security:**

* End-to-end encryption for video and audio streams.
* Secure user authentication (OAuth, token-based).
* Encrypted storage of user data and meeting recordings.

**4. Usability:**

* Intuitive UI for both hosts and participants.
* Accessible for users with varying technical expertise.
* Support for multiple languages (optional).

**5. Reliability:**

* The system should have an uptime of 99.9%.
* Automatic reconnection in case of network failures.

**6. Cross-Platform Support:**

* The app should be functional on Android, iOS, and desktop (Windows, macOS, Linux).

**7. Privacy:**

* Personal data (e.g., meeting recordings, user details) must be anonymized where applicable and should be accessible only by authorized users.

**8. Maintainability:**

* The codebase should be modular and well-documented to allow easy updates and bug fixes.