Project Design Phase-I
Solution Architecture

Date	23 October 2023
Team ID	Team 590895
Project Name	Chat connect- A real time chat and communication app
Maximum Marks	4 Marks

Solution Architecture:

User Authentication:

- Components:
    1. User Database: Store user account information, including user ID, email, password, and authentication status.
    2. Authentication Process: Use Firebase Authentication or other methods for user sign-up, login, and password reset.
    3. User Profile: Display user information, including their name, email, and profile picture.
    4. Front-end: Implement UI for user registration, login, and profile management.
    5. Security Rules: Set up Firebase Security Rules to secure user data and authentication.
- Workflow:
    1. Users create accounts by providing their email and password.
    2. Email verification may be used to ensure the authenticity of user accounts.
    3. Users can update their profiles, including their name and profile picture.

Status/Story:

- Components:
    1. Firestore Database: Store user status updates, including user ID, timestamp, and content.
    2. Storage: Store media files (images, videos) related to status updates.
    3. User Interface: Implement a UI for users to view and post status updates.
- Workflow:
    1. Users can post status updates, which are stored in Firestore.
    2. Media files associated with status updates are stored in Firebase Storage.
    3. Users can view and interact with status updates in the app.

Profile Picture:

- Components:
    1. Firebase Storage: Store user profile pictures.
    2. User Profile: Display the user's profile picture.
- Workflow:
    1. Users can upload their profile pictures to Firebase Storage.

2. The profile picture URL is stored in the user's data in Firestore.
3. The user's profile picture is displayed in their user profile.

Chatbot (Predefined Responses):

- Components:
  1. Realtime Database or Firestore: Store chat messages between users and the chatbot, including predefined responses.
  2. User Interface: Implement a chat interface for users to interact with the chatbot.
  3. Chatbot Logic: Define predefined responses for specific user inputs.
- Workflow:
  1. Users can send messages to the chatbot.
  2. The app checks if user messages match predefined responses in the database.
  3. If a match is found, the predefined response is sent; otherwise, the message is stored, and a generic response is provided.

Video Calling Feature:

Components:

- User Interface (UI):
  Video call buttons, user profiles, and the video call screen.
- Signalling Servers:
  Technologies: WebSockets or SignalR.
  Functions: Call initiation, call acceptance, and user presence updates.
- Media Server:
  Technology: WebRTC.
  Supporting Servers: STUN and TURN servers for NAT traversal and firewall support.
- Codecs and Media Processing:
  Features: Echo cancellation, noise reduction, and bandwidth adaptation.
- User Authentication and Authorization:
  Process: User login and authentication.
- Data Synchronization:
  Purpose: Call history, call status (e.g., ongoing, missed, completed), and call-related data.
- Push Notifications:
  Function: Notify users of incoming video call requests or missed calls.
- Security and Encryption:
  Priority: Implement end-to-end encryption for privacy.

Workflow:

- User initiates a video call from the UI.
- Signalling server handles the call initiation and notifies the recipient.
- The recipient accepts the call request through the UI.
- Signalling server establishes a connection between caller and recipient.
- Media server (WebRTC) handles audio and video streaming directly between clients.

- Codecs and media processing enhance call quality.
- User authentication and authorization ensure only authorized users can initiate calls.
- Call data is synchronized and stored for future reference.
- Push notifications notify users of incoming video call requests or missed calls.
- End-to-end encryption is implemented for secure video call content.

Audio Calling Feature:

Components:

- User Interface (UI):
  Call buttons, user profiles, and the in-call screen.
- Signalling Server:
  Technology: WebSockets or SignalR.
  Functions: Call initiation, call acceptance, and user presence updates.
- Media Server:
  Technology: WebRTC or SIP (Session Initiation Protocol).
  Supporting Servers: STUN and TURN servers for NAT traversal.
- Audio Codecs:
  Usage: Audio codecs (e.g., Opus, G.711) for audio stream compression and decompression.
- Audio Quality Enhancement:
  Features: Echo cancellation and noise reduction.
- User Authentication and Authorization:
  Process: User login and authentication.
- Data Synchronization:
  Purpose: Call history, call status (e.g., ongoing, missed, completed), and call-related data.
- Push Notifications:
  Function: Notify users of incoming audio call requests or missed calls.
- Security and Encryption:
  Security Measure: Implement end-to-end encryption for audio content privacy.

Workflow:

- User initiates an audio call from the UI.
- Signalling server handles the call initiation and notifies the recipient.
- The recipient accepts the call request through the UI.
- Signalling server establishes a connection between caller and recipient.
- Media server (WebRTC or SIP) handles audio streaming between clients.
- Audio codecs compress and decompress audio streams.
- Audio quality enhancement features improve call quality.
- User authentication and authorization ensure only authorized users can initiate audio calls.
- Call data is synchronized and stored for future reference.
- Push notifications notify users of incoming audio call requests or missed calls.
- End-to-end encryption is implemented for secure audio call content.

Voice Messaging:

Components:

- User Interface (UI): Voice messaging button in chat interface. List of voice messages in chat history.
- Voice Messaging Server: Responsible for recording, storing, and delivering voice messages.
- Recording and Playback: Components for recording and playing voice messages within the app.
- Data Synchronization: Ensure voice messages are synchronized across devices and users.

Workflow:

- User selects the voice messaging button in the chat interface.
- The app prompts the user to record a voice message.
- The recorded message is sent to the voice messaging server for storage.
- The recipient is notified of the incoming voice message.
- The recipient can play the voice message from the chat interface.
- Data synchronization keeps track of sent and received voice messages.

Chat Themes Feature:

Components:

- User Interface (UI): Themes selection screen.Theme customization options.
- Theme Storage: Store theme preferences and customizations.

Workflow:

- Users access the chat themes feature from the UI.
- They can select from pre-defined themes or customize their own.
- Theme preferences are stored for the user to apply in the chat interface.

Verified Accounts (Tick Marks):

Verified accounts are typically used to confirm the authenticity of an account or user. This can be crucial for ensuring trust and security in a chat application.

Components:

- User Database: This is where user account information is stored, including account status (verified/unverified).
- Verification Process: A verification process should be in place to confirm user identity. This may include document submission, email verification, or phone number verification.
- User Profile: Each user should have a user profile page where their verification status is displayed, typically with a tick mark symbol to indicate a verified account.

- Front-end: The application's user interface should display tick marks next to verified users' names in the chat interface.
- Verification Badge Management: An admin panel or system should be available for managing and updating verification badges.

Workflow:

- User initiates the verification process by providing necessary information and documents.
- The application's backend verifies the submitted information through an automated or manual review process.
- If verification is successful, the user's account status is updated to "verified," and a tick mark badge is displayed on their profile and next to their name in chats.

Text Reading:

The text reading feature allows users to have messages read aloud, which can be beneficial for accessibility or when users are unable to read the text.

Components:

- Text-to-Speech (TTS) Service: You need a Text-to-Speech service or API that can convert text messages into spoken words. There are several TTS services available, such as Google Text-to-Speech, Amazon Polly, or Microsoft Azure Cognitive Services.
- Chat Interface: Within the chat interface, you should provide a button or option for users to initiate the text reading feature.
- Audio Output: Ensure that the spoken text is played through the user's device's audio output (speaker or headphones).
- User Settings: Allow users to customize the TTS settings, such as voice selection, speed, and volume.
- Accessibility Features: Implement accessibility features to make it easier for users with disabilities to access the text reading functionality.

Workflow:

- User selects a message in the chat interface they want to be read aloud.
- The selected text is sent to the TTS service, which converts it into speech.
- The generated speech is played through the user's device's audio output.

Considerations:

- Ensure that the text reading feature is accessible and customizable for users with different needs.
- Implement privacy controls to allow users to opt-in or opt-out of having their messages read aloud.

Solution Architecture Diagram: