Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	12 October 2023
Team ID	Team - 593383
Project Name	ChatConnect
Maximum Marks	5 Marks

Technical Architecture:

The Deliverable shall include the architectural diagram as below and the information as per the table1 & table 2.

Architecture

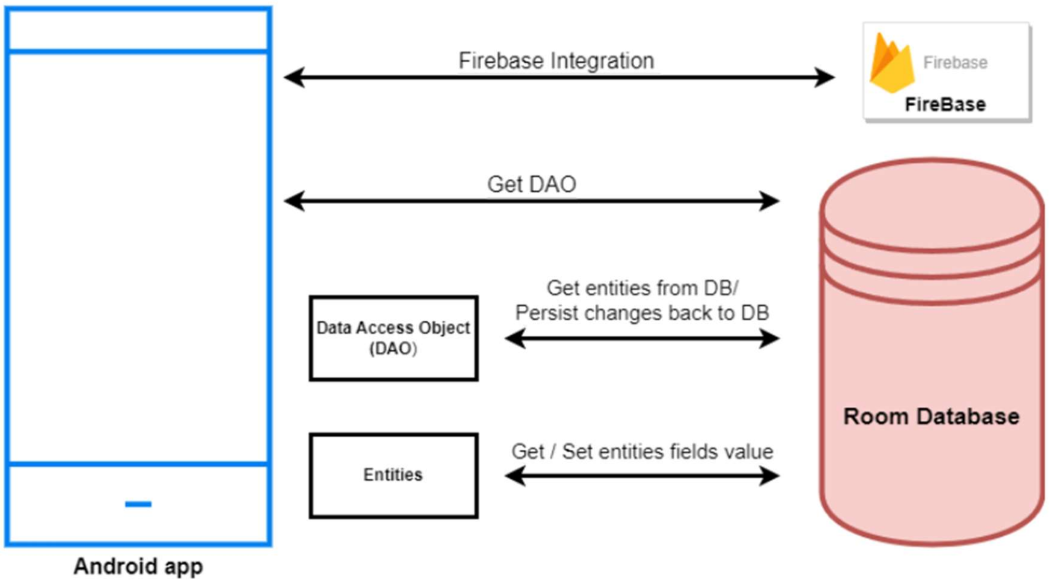

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface Design	Create a modern UI with Jetpack Compose.	Jetpack Compose
2.	User Experience	Optimize the app's user interface.	Jetpack Compose animations
3.	Authentication	Secure user login and identity verification.	Firebase Authentication.
4.	Application Logic-1	Logic for a process in the application	Kotlin
5.	Application Logic-2	Chat app functionalities, such as sending notifications or handling specific events, require server-side logic.	Firebase Cloud Functions
6.	Database	Data Type, Configurations etc..	NoSQL
7.	File Storage	Store images and videos in chats.	Firebase Cloud Storage
8.	External API-1(Storage API)	Enable users to share and access files from cloud storage services.	Google Drive
9.	External API-2 (Emoji API)	Expanding chat app's emoji library by integrating additional emoticon packs.	Twemoji API.
10.	Cloud Functions	Implement server-side logic.	Firebase Cloud Functions
11.	Testing and Debugging	Test and debug the app.	Android Studio tools
12.	Infrastructure Server	For Seameless chatting and connectivity	Firebase server
13.	Documentation	Document code and app structure.	Github

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	MVIKotlin
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Firebase security rules and authentication.
3.	Scalable Architecture	Ability to handle growing user and message volume.	Firebase scalability features.
4.	Real-time Messaging	Instant message delivery and updates.	Firebase Realtime Database
5.	Push Notifications	Real-time alerts for new messages.	Firebase Cloud Messaging (FCM).
6.	Offline Access	Use the app without an internet connection.	Firebase Realtime Database or Cloud Firestore
7.	Testing & Debugging	Ensure app stability and resolve issues.	Android Studio tools, Firebase testing.
8.	Chat History	Access and search past messages.	Firebase Realtime Database or Cloud Firestore.
9.	Availability	Ensures uninterrupted service with load balancing, distributed servers, and automatic failover.	Firebase Realtime Database, Firebase Cloud Functions
10.	Performance	Ensures the app is accessible at all times. Achieved through distributed servers and Firebase's real-time syncing.	Firebase Realtime Database or Cloud Firestore for synchronized data across multiple servers and regions.
11.	Search and Chat History	Enabling users to search for older messages and access their chat history.	Firebase Realtime Database