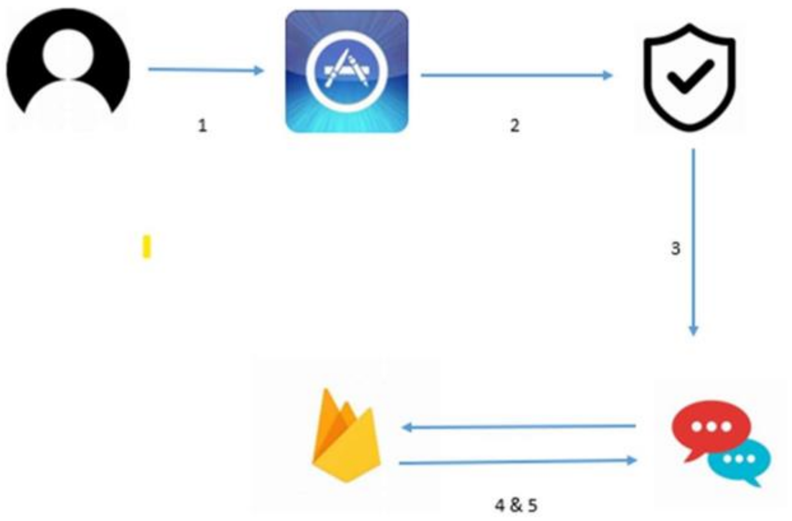
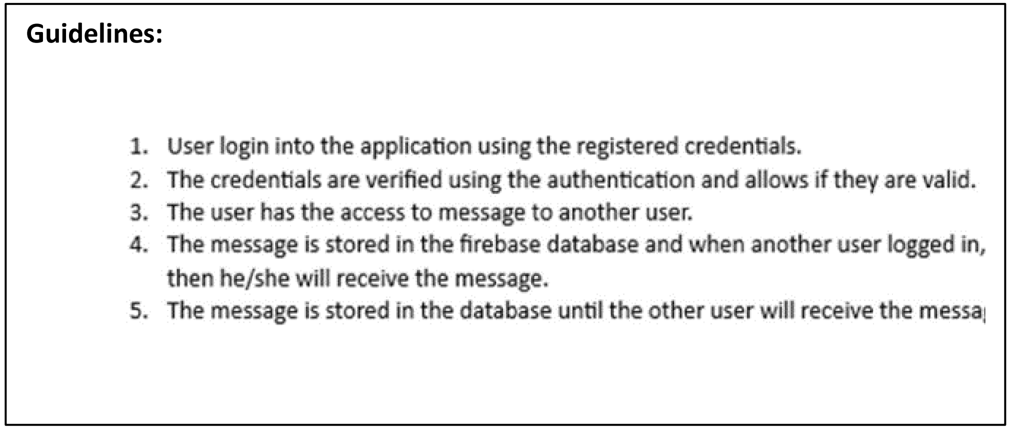
Project Design Phase-II
Technology Stack (Architecture & Stack)

Date	24 October 2023
Team ID	Team-591070
Project Name	CHATCONNECT- A Real time chat and communication app
Maximum Marks	4 Marks

Technical Architecture:

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





Guidelines:

1. User login into the application using the registered credentials.
2. The credentials are verified using the authentication and allows if they are valid.
3. The user has the access to message to another user.
4. The message is stored in the firebase database and when another user logged in, then he/she will receive the message.
5. The message is stored in the database until the other user will receive the messa

Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	user interaction with application	Kotlin Jetpack Compose
2.	Application Logic-1	Logic for a process in the application	KOTLIN
3.	Application Logic-2	Logic for a process in the application	Android Compose UI Tool KIT
4.	Database	Data Type, Configurations	Shared Preference
5.	Cloud Database	Database Service on Cloud	Firebase (REALTIME DATABASE)
6.	File Storage	File storage requirements	LOCAL Filesystem
7.	External API-1	Purpose of External API used in the application	Google Firebase
8.	Application Type	Purpose of Application	Real time messaging application
9.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration:	Firebase, Local

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Declarative framework for sharing UIs across multiple platforms
2.	Security Implementations	List all the security / access controls implemented, use of firewalls	Encrypt data in transit using HTTPS and logically isolate user data
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Database uses a JSON-based, hierarchical data structure. Data is organized into paths and nodes, with real-time synchronization and security rules. It scales automatically and

S.No	Characteristics	Description	Technology
			supports offline capabilities for responsive apps.
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	application is available through all Google's global infrastructure, with load balancing and distributed servers across multiple regions. This reduces downtime and ensures reliability. Load balancing and data replication enhance fault tolerance for seamless user experiences.
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	For optimal performance, consider efficient data structuring to minimize requests, utilize Realtime Database's caching, and leverage Content Delivery Networks (CDNs) for fast content delivery. These strategies help handle high request rates, reduce server load, and enhance user experiences.