

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

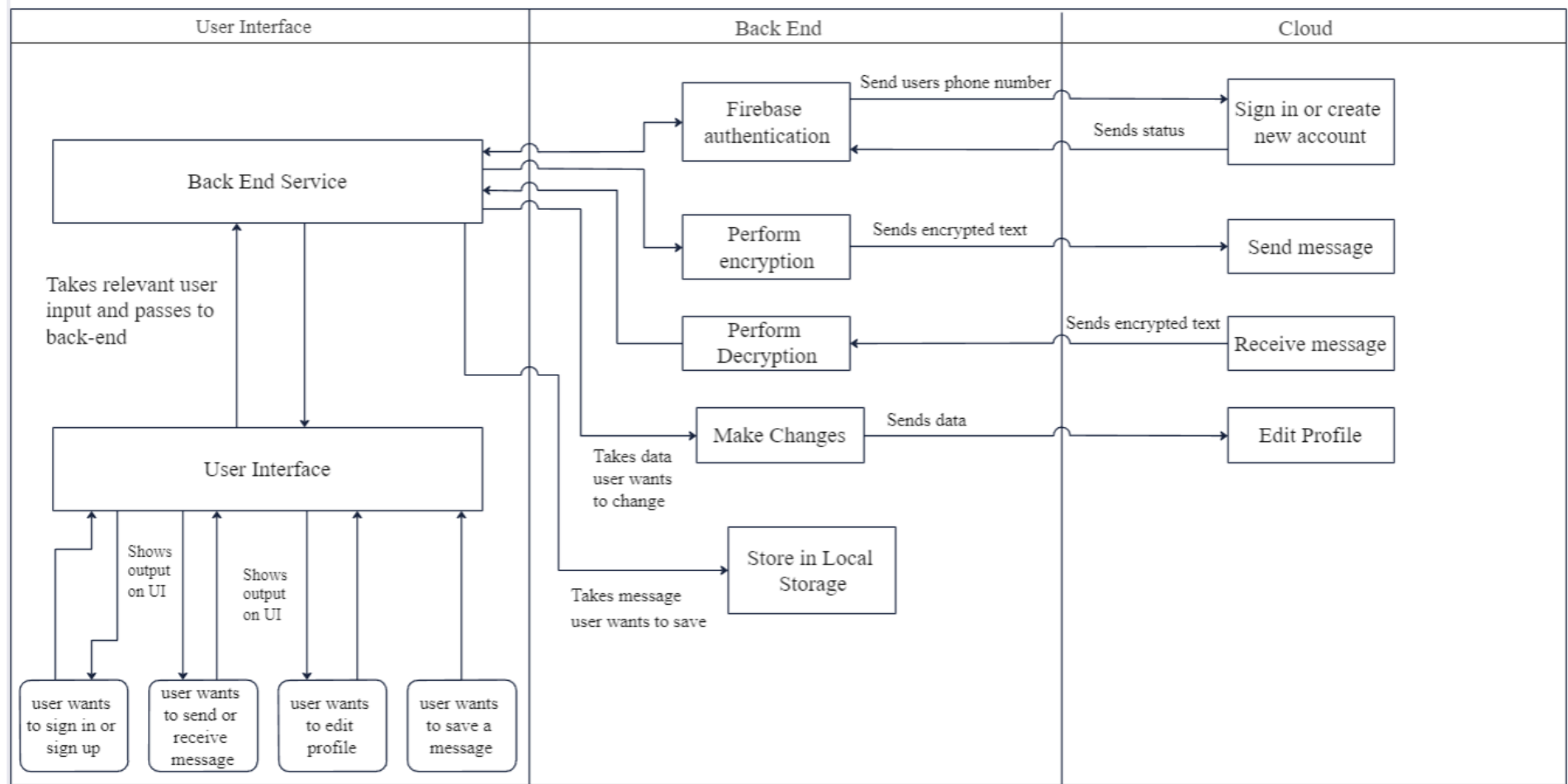
Date	27 October 2023
Team ID	590913
Project Name	ChatConnect - A Real-Time Chat and Communication App
Maximum Marks	4 Marks

**Technical Architecture:**

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

**Guidelines:**

1. Include all the processes (As an application logic / Technology Block)
2. Provide infrastructural demarcation (Local / Cloud)
3. Indicate external interfaces (third party API's etc.)
4. Indicate Data Storage components / services
5. Indicate interface to machine learning models (if applicable)



**Table-1 : Components & Technologies:**

S.No	Component	Description	Technology
1.	User Interface	How user interacts with application e.g. Mobile App	Jetpack Compose, XML
2.	Sign in/up Logic	Take user phone number to login user or create new account	Kotlin, NoSQL
3.	Send/receive message Logic	Take message input then encrypt it and send it to database. Receive message from database and then decrypt it.	Kotlin, NoSQL, RSA Cryptography
4.	Edit profileLogic	Take user input for changes to be made and send data to database	Kotlin, NoSQL
5.	Save message Logic	Take a message to save and store it in local storage.	Kotlin, JSON
6.	Cloud Database	Database Service on Cloud	Firebase
7.	File Storage	File storage requirement for storing messages and private key	Local Filesystem
8.	External API	Firebase used for user authentication, store user related data and handle messages	Firebase API
9.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Local, Firebase

**Table-2: Application Characteristics:**

<b>S.No</b>	<b>Characteristics</b>	<b>Description</b>	<b>Technology</b>
1.	Open-Source Frameworks	List the open-source frameworks used	Firebase
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Authentication and authorization, End-to-End Encryption
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Kotlin, jetpack compose, NoSQL, Firebase
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Google Cloud
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Database Indexing

**References:**

<https://c4model.com/>

<https://developer.ibm.com/patterns/online-order-processing-system-during-pandemic/>

<https://www.ibm.com/cloud/architecture>

<https://aws.amazon.com/architecture>

<https://medium.com/the-internal-startup/how-to-draw-useful-technical-architecture-diagrams-2d20c9fda90d>