

S.No	Component	Description	Technology
1.	User Interface	Mobile App	Kotlin
2.	login screen	<p>a. Check if both the username and password fields are not empty. If they are empty, display an error message asking the user to fill in all fields.</p> <p>b. If both fields are filled, query the user database to find a user with the username entered.</p>	Android, Kotlin, Compose, SQLite Database, Intents, Material Design, Image Loading, State Management, UI Layout.
3.	registration screen	<p>a. Check if all the fields (username, email, and password) are not empty. If any of them are empty, display an error message asking the user to fill in all fields. b. If all fields are filled, create a User object with the provided information. c. Insert the user into the database using the databaseHelper.insertUser(user) method. d. Display a success message ("User registered successfully"). e. Redirect the user to the login screen (LoginActivity).</p>	Android, Kotlin, Compose, SQLite Database, Intents, Material Design, Image Loading, State Management, UI Layout.
4.	Database	UserDao is an interface that defines methods for performing database operations on a User entity., @Dao	Android, Kotlin, Compose, SQLite Database, Intents, Material Design, Data Persistence
5.	Infrastructure (Server / Cloud)	This infrastructure includes server resources, databases, content delivery networks, and load balancing for optimized performance. Security measures, authentication, and authorization are critical to protect user data.	Amazon Web Services (AWS)

Table-2: Application Characteristics:

S.NO	Characteristics	Description	Technology
1.	Security Implementations	Use encryption protocols (e.g., TLS/SSL) to secure data transmission between the app and the server. This prevents eavesdropping on sensitive information during transit.	TLS/SSL, SHA-256, Encryptions
2.	Scalable Architecture	Implement a load balancer to evenly distribute incoming traffic across multiple servers. This ensures high availability and even resource utilization.	Amazon Web Services (AWS),
3.	Availability	99.9% uptime, good scalability	Load Balancing Algorithms, Server Redundancy
4.	Performance	The number of requests per second, the use of caching, and content delivery networks (CDNs), are essential to ensure a smooth and efficient user experience.	horizontal scaling, Caching, Database Optimization