

Project Design Phase-II

Technology Stack (Architecture & Stack)

Date	26 October 2023
Team ID	PNT2022TMID591106
Project Name	Project - An Android Application for Keeping Up with the Latest Headlines
Maximum Marks	4 Marks

Technical Architecture:

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

An Android Application for Keeping Up with the Latest Headlines :

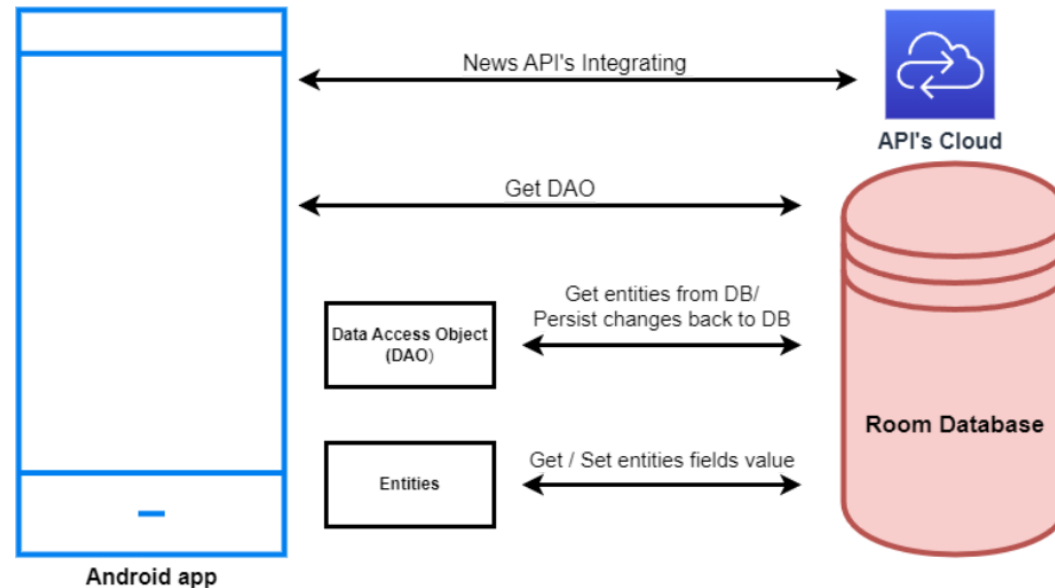


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Users interact with the application through the mobile app.	Jetpack Compose for UI design and interaction.
2.	News Retrieval Logic	Retrieves and processes news data from external sources and display them to the user	Retrofit for network requests, OkHttp for HTTP client.News API for fetching news data
3.	News Caching Logic	Caches news articles locally for offline access and faster loading	Room Database for local data storage.
4.	User Preferences Logic	Manages user settings, favorites, and personalized content	Shared Preferences for storing user-specific data.
5.	Image Loading	Loads images for news articles	Coil Library
6.	Database	Storage of news articles and app-related data.	Room Database for local storage in Android.
7.	Navigation	Easy navigation to browse different news sections and articles	Navigation Component for handling app navigation. Navigation Drawer for section switching.
8.	Custom UI Elements	Unique UI components for an engaging user experience.	Custom Composables built with Jetpack Compose. Material Design components for a consistent UI.
9.	News Data API	Provides access to news articles and headlines from various sources	Eg:News API.
10.	User Authentication	User registration and login functionality	Firebase Authentication, or custom server-side solution

11.	Error Handling	Graceful handling of errors, network issues, and data problems	Exception handling through try-catch blocks. Custom error Composables for displaying error messages.
12.	Permission Handling	Managing app permissions for device access	Android permission system
13.	Testing	Comprehensive testing to ensure app reliability	JUnit for unit testing. Espresso for UI testing. MockWebServer for simulating network requests in tests.
14.	Data Model	Defining data structures for news articles	Kotlin data classes
15.	API Integration	Communicating with a remote server for news data	RESTful API communication
16.	Dependency Management	Managing external libraries and modules	Gradle build system
17.	Manifest Configuration	Setting up app configuration and permissions	AndroidManifest.xml configuration
18.	App Architecture	Designing the app's architecture	Model-View-ViewModel (MVVM) architecture pattern
19.	Firebase Firestore	Cloud-based NoSQL database for storing user preferences and app-related data	Firebase Firestore is a scalable and real-time cloud database provided by Google Firebase. It allows efficient storage and retrieval of structured data, making it suitable for mobile apps like this one.

20.	Infrastructure (Local Deployment)	Running the app on local Android devices for development and testing	Android Emulator: A built-in emulator in Android Studio for local app testing on various device configurations.
-----	-----------------------------------	--	---

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Utilizing open-source tools and libraries for development.	<ul style="list-style-type: none"> Jetpack Compose UI Toolkit: Open-source Android UI framework. Coil: Open-source image loading library. Retrofit: Open-source HTTP client for making network requests. Room Persistence Library: Part of Android Jetpack for local data storage.
2.	Security Implementations	Implementing security measures to protect user data and app integrity.	<ul style="list-style-type: none"> Firebase Authentication: Provides secure user authentication. HTTPS (SSL/TLS): Secure communication with the remote server using HTTPS protocol. Encryption: Data encryption for sensitive information storage and transmission.

S.No	Characteristics	Description	Technology
3.	Scalable Architecture	Designing the app's architecture for scalability, easy maintenance, and future updates.	<ul style="list-style-type: none"> • Model-View-ViewModel (MVVM) architecture: Separation of concerns for easier scalability. • Repository pattern: Isolating data access for flexibility and testability. • Dependency Injection: Dagger Hilt for managing dependencies and scalability.
4.	Availability	Ensuring the app is available to users with minimal downtime.	<ul style="list-style-type: none"> • Load Balancing: Distributing incoming network traffic to maintain availability. • Server redundancy: Multiple server instances to ensure availability in case of server failures.
5.	Performance	Optimizing the app's speed and responsiveness	<ul style="list-style-type: none"> • Caching: Use of local caching to reduce network requests. • Image Compression: Compressing and optimizing images for faster loading. • Background Tasks: Implementing background processing to keep the app responsive. • Profiling and optimization tools: Android Profiler and tools for performance analysis and improvement

