

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

Date	27 October2023
Team ID	Team-591084
Project Name	Wanderlust - A Personalized Travel Planning & Tracking App
Maximum Marks	4 Marks

Technical Architecture:

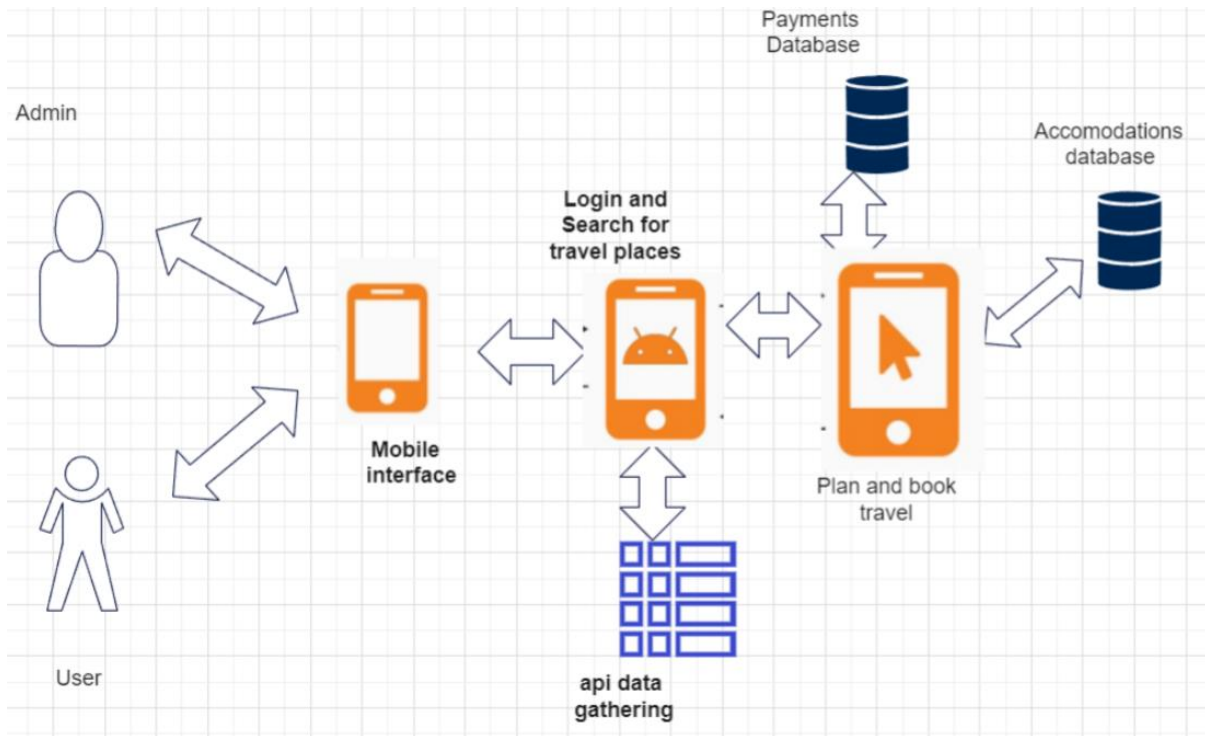


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1	User Interface	Android application for user interaction	Jetpack compose,XML
2	Application Logic	Basic operations and commercial reasoning, such as creating travel itineraries	Kotlin , Android SDK
3	Database	Data storage and retrieval for user profiles, itineraries, etc.	MySQL
4	Cloud Database	Cloud-based database for secure data management	Firebase Realtime Database

S.No	Component	Description	Technology
5	File Storage	Storage of user-generated content and files related to travel documents and media	Firebase Storage
6	External API-1	Integration with third-party services for maps and weather information.	Google Maps API , OpenWeather API, Firebase SDK
7	External API-2	Integration with third-party services hotel booking and flight booking	Payment Gateways, Booking.com API.
8	Security	Implementation of security measures to protect user data and transactions	Android Keystore System, HTTPS, ProGuard
9	Notification System	Push notifications and alerts for users to keep them updated on travel status	Firebase Cloud Messaging (FCM)
10	Machine Learning	Personalized recommendations and insights based on user preferences for travel destinations and activities	TensorFlow Lite for on-device ML models

Table-2: Application Characteristics:

s.no	Characteristics	Description	Technology
1.	Open-Source Frameworks	The application utilizes a variety of open-source frameworks to enhance its functionality and performance.	Retrofit, OkHttp
2.	Security Implementations	The app prioritizes security through the use of ProGuard for code obfuscation, HTTPS for encrypted communication, Encryption for data protection, and Android	ProGuard ,HTTPS ,Encryption, Android Keystore

		<p>Keystore for secure key management.</p> <p>Although conventional firewalls are not directly employed, these robust security measures collectively fortify user data, thwart unauthorized access, and maintain the integrity of the application</p>	
3.	Scalable Architecture	<p>using a scalable architecture that leverages the power of cloud computing and microservices.</p> <p>This architecture enables the app to handle a large number of users and transactions without compromising performance.</p>	<p>Firebase Remote Config, Firebase Realtime Database</p>
4.	Availability	<p>The application uses Firebase Remote Config for dynamic configurations, ensuring that the application remains available even during runtime changes.</p> <p>Additionally, Firebase Realtime</p>	<p>Firebase Remote configurations, Firebase Realtime Database</p>

		Database is employed for real-time data synchronization, enabling the app to provide continuous access to the latest information.	
5.	Performance	The application utilizes caching mechanisms such as SharedPreferences to store frequently accessed data locally, minimizing the number of requests per second to the server. Moreover it enhances the app's performance and reducing load times for users	Efficient RecyclerView usage, Kotlin Coroutines for background tasks