

Project Design Phase-I

Solution Architecture

Date	23 October 2023
Team ID	SI-GuidedProject-587265-1696856877
Project Name	Money Matters: A personal financial management app
Maximum Marks	4 Marks

Solution Architecture:

Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions.

- **Solution** – our project includes a client-side Jetpack Compose UI, a backend API built with Kotlin and a secure database like SQLite or a cloud-based solution (e.g., Firebase) to store user financial data. We will implement user authentication, data encryption, and synchronization for a seamless and secure user experience.
- **Structure, Characteristics & Behaviour** – It comprises a client-side mobile app built with Jetpack Compose and Kotlin, featuring an intuitive user interface for expense tracking and management. It communicates with a backend API, implemented in Kotlin or a compatible technology, which handles data storage, retrieval, and processing. The software ensures data security through user authentication, encryption, and synchronization with a database, enabling users to input, view, and manage their financial expenditures seamlessly while maintaining the privacy and integrity of their financial data.

- **Features, Development phases, and Solution requirements –**

A) Features:

- **Expense Tracking:** record and categorize expenses, including the date, amount etc.
- **Budget Management:** Set and track budget limits for different spending categories
- **User Authentication:** Secure login and registration to protect user data
- **Notifications:** Send reminders for upcoming bills and budget limits
- **Expense History:** Maintain a history of past transactions for reference
- **Secure Data Storage:** Safeguard financial data using encryption

B) Development Phase:

- **UI/UX Design:** Create wireframes and design the user interface using Jetpack Compose.
- **Frontend Development:** Build the mobile app with Jetpack Compose.
- **Backend Development:** Develop the backend API for data storage and retrieval.
- **Authentication:** Implement user authentication and security measures.
- **Database Integration:** Set up the database (e.g., SQLite or a cloud-based solution).
- **Synchronization:** Develop data synchronization mechanisms.
- **Notifications:** Add notification features.

C) Requirements:

- **Security:** Implement strong encryption, secure authentication, and data privacy measures.
- **Scalability:** Design the architecture to handle a growing user base.
- **User-Friendly UI:** Create an intuitive and visually appealing user interface.
- **Performance:** Optimize for speed and responsiveness.
- **Offline Access:** Enable users to access and record data even when offline.
- **Data Backup:** Provide a backup solution to prevent data loss.

- **Solutions delivered via –**

- 1) Authentication – Firebase, Email & Phone Number
- 2) Expense Track, Budget Management, Visualization – Kotlin
- 3) Data Storage, Filter, Sort – SQLite
- 4) Notifications – AndroidX Core
- 5) User Profile, Data Synchronization – Firebase
- 6) Security – Android

Example - Solution Architecture Diagram:

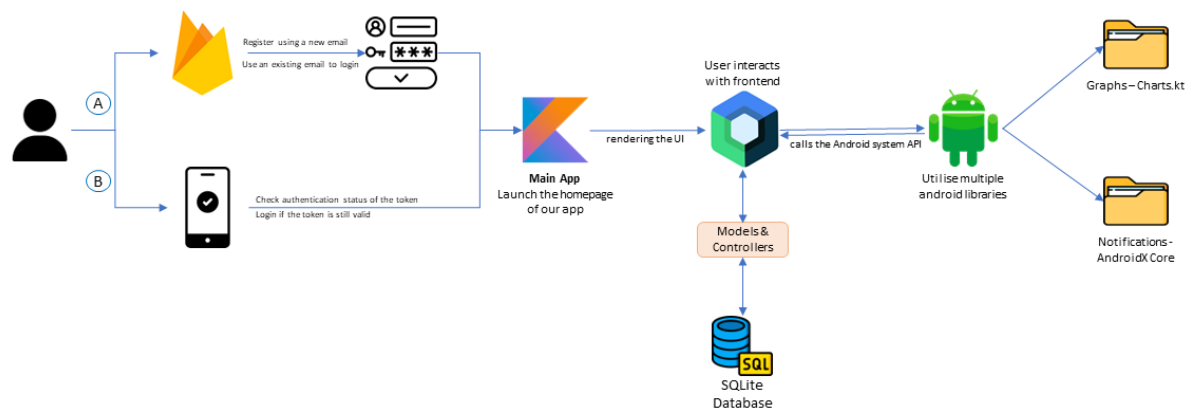


Figure 1: Architecture of an app to manage financial expenditure of user