

1. How would you implement an offline mobile money app that allows only one party to be online in the transaction and will be used by at least 100,000 users in the first year of launch?

Designing and implementing an offline mobile money application would require careful planning and consideration of different functionalities and features required for the application. Here are some general steps that can guide the implementation process:

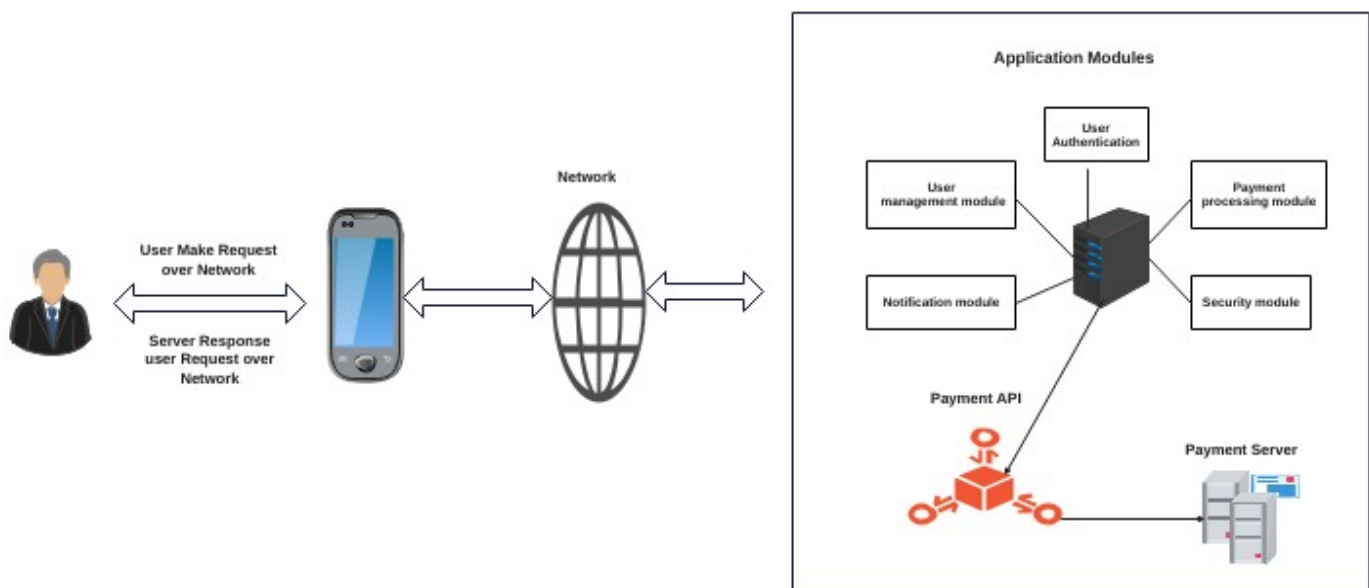
1. **Define the requirements:** Start by defining the requirements of the mobile money application. This includes identifying the target users, their needs, and the types of transactions the application will support. Determine the key features and functionalities, such as account creation, balance inquiry, cash-in/cash-out, money transfer, bill payments, and other value-added services.
2. **Choose a platform:** Decide on the platform on which the mobile money application will run. The platform can be Android, iOS, or both. It is important to ensure that the chosen platform supports offline transactions, as this will be a critical component of the application.
3. **Choose a programming language:** Depending on the chosen platform, choose a programming language that is compatible with the platform. For instance, if you are developing for iOS, you can use Swift or Objective-C.
4. **Develop the application:** Develop the application according to the requirements and platform specifications. Use an offline-first approach to ensure that the application can function even without an internet connection. Implement data synchronization to enable data exchange between the mobile application and the back-end system.
5. **Implement security:** Ensure that the mobile money application is secure. This involves implementing encryption, authentication, and authorization measures to safeguard the user's data and transactions.
6. **Test the application:** Test the mobile money application thoroughly to ensure that it meets the required standards. Test it in different scenarios, including offline and online situations, to ensure that it functions optimally in all situations.

7. **Launch the application:** Once the application has been tested and verified to be working, launch it to the public. Promote the application and monitor its performance to identify areas of improvement.
8. **Provide customer support:** Offer customer support to users of the mobile money application. Provide contact information that users can use to reach out to you for any issues or questions they may have.

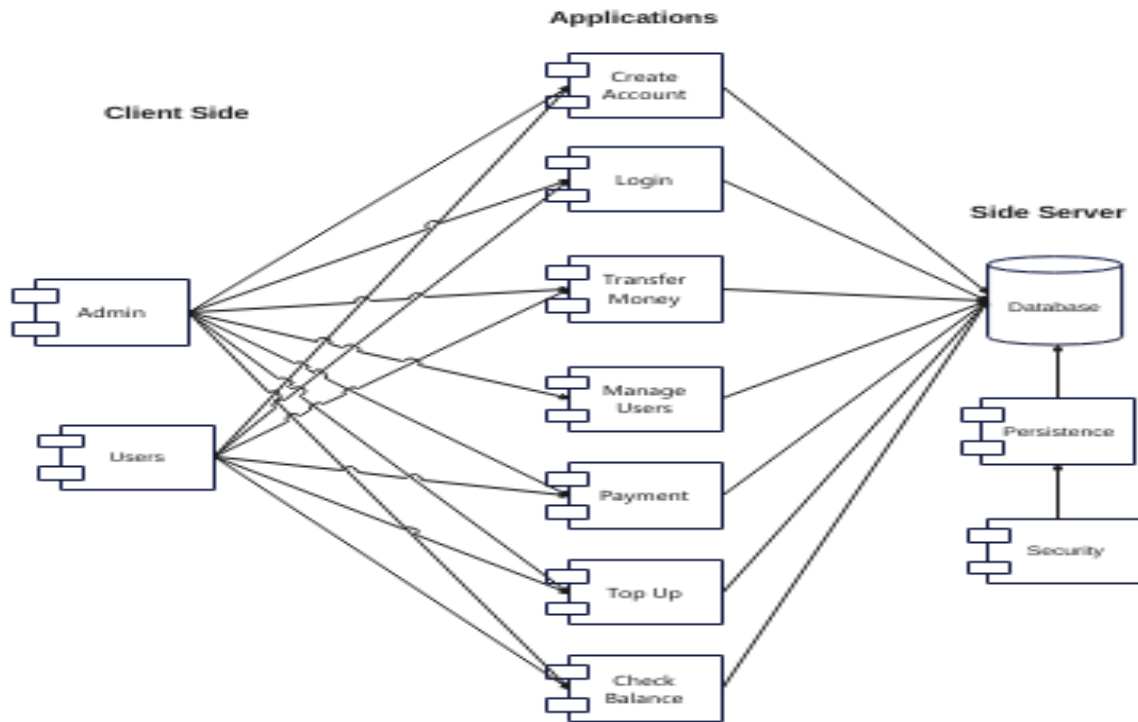
2. Design an application architecture for the above application. The application architecture diagram should include all application modules, components, and the interactions between them that constitute the application. You only need to attach the application architecture diagram & send it to jobs@360ground.com

Some common application modules for mobile money applications include: Authentication module, Payment processing module, Security module, User management module, Notification module, Analytics module etc

2.1 Application Module for Mobile money app



2.2 Component Diagram for Mobile money app



2.3 Class Diagram for Mobile money app

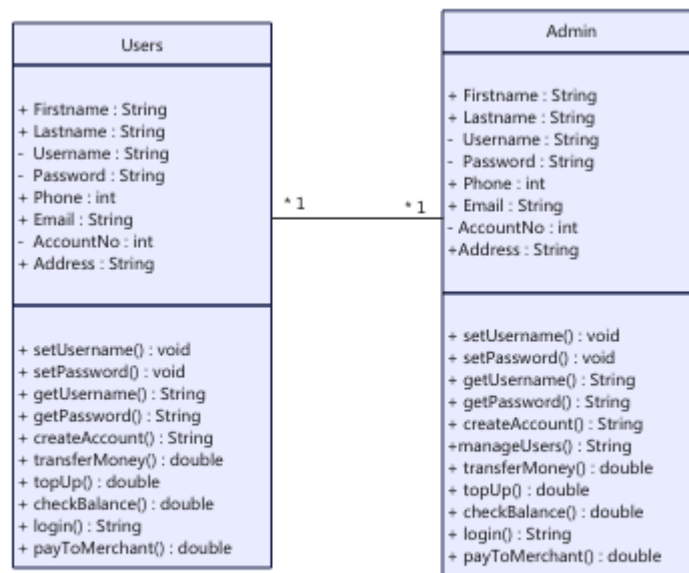


Figure 1.3 Class Diagram for Mobile money app

3. Propose a technology (framework, kit...) to implement the mobile & web app and discuss why you picked it.

Frame works for the developments of web apps		
Front end(Client side)	Back-end(server side)	Back-ed(Storage)
React js	Laravel	Fire-base
The Reason why I selected them is :		
React js is easy to learn	MV C support and Object oriented approach	Fast and save hosting
React js has a Large Development Community	Build in Authentication and Authorization	Reliable and extensive database
Virtual DOMs	Packaging System	Google analytic
JSX increase the performance and efficiency of React js	Eloquent ORM	Free multi platform fire base authentication
Redux maintains data consistency across all components	Events and Broadcasting	Fire base testing service to improve app quality
React Hooks	Testing	Machine learning capabilities

Frame works for the developments of Mobile apps		
Front end(Client side)	Back-end(server side)	Back-ed(Storage)
React js	Laravel	Fire-base
The reason why i selected them is :		
Fire-base integration	MVC support and Object oriented approach	Fast and save hosting
Smooth animation	Build in Authentication and Authorization	Reliable and extensive database
Dart codes	Packaging System	Google analytic
Reusable widgets	Eloquent ORM	Free multi platform fire base authentication
Faster Development	Events and Broadcasting	Fire base testing service to improve app quality
Important flutter tool	Testing	Machine learning capabilities

