1. Quality Attributes Analysis for Candidate Architectures

	Advantages	Disadvantages
Layered System	1. It is used to control and encapsulate the complexity of large applications. 2. It is intuitively easier to achieve the separation of concerns from the user interface to the business logic and the data access logic. 3. This architecture enables the development of loosely coupled systems. Different components deployed, maintained, and updated independently. 4. Suitable for interactive systems. 5. Testability is high as unit testings can be performed in each component. 6. Maintainability: enhancement of functionality can be achieved by introducing new components without affecting existing ones. 7. Reusability: Lower layers can be reused to support various upper layers or multiple applications.	1. More upfront efforts are required to design systems in a layered fashion. 2. If there are too many layers, changes will pass slowly to higher layers as more overheads of passing through layers will be generated. 3. It adds extra complexity to simple applications. 4. It is difficult to exactly assign functionalities to the correct and appropriate layer.
Pipe and Filter	1. The simplicity of the design allows the developers and designers to quickly understand the overall input/output data flow logic. 2. Reusability is high as each filter can be called overtimes 3. The concurrent execution feature allows each filter to be implemented with a single and separate task in parallel. It guarantees efficiency and maintainability.	1. Not suitable for interactive systems 2. Difficulties in handling the cooperations between the filters 3. Need to balance the workload to avoid one bottleneck to block the whole pipeline

	4. The loose coupling structure ensures flexibility between each filter.	
Client & Server	1. This architecture provides a central administration. The server helps control system set-up and user access rights and resource allocation. 2. This architecture stores all the data on the server. The design highly improves its robustness, assisting efficient recovery, and preventing data loss from a cascaded break-down. 3. Security can be ensured by setting rules and access rights at the system set-up. 4. Scalability and up-gradation are guaranteed by only making changes and feature additions on the server-side.	1. The server can be heavily loaded if all clients request data simultaneously, which may lead to network congestion and lagging system response. 2. Failure of the server may leave all client requests unfulfilled. 3. The cost of setting and maintaining the server is high due to its high computational ability requirements. 4. Complicated protocols are required to achieve communication between one client to another since all clients are independent.

2. Conclusion

Layered System is chosen as the architecture pattern of our application as the nature of our application requires a lot of user interactions which can not be handled well by the Pile and Filter architecture. According to our software design, new features will be triggered by clicking on the mobile application. This request will be handled by the front end which will then pass on the requests to the APIs of the back end servers. The APIs will unwrap the request and call the backend services to query the database. This mechanism naturally forms a layered structure. The server needs to be guaranteed not to be heavily loaded as there are some simultaneous requests. Hence, Client & Server architecture is not suitable. Reusability and maintainability are essential factors that we considered during the application design stage. Therefore, the layered system architecture is the best choice for our application.