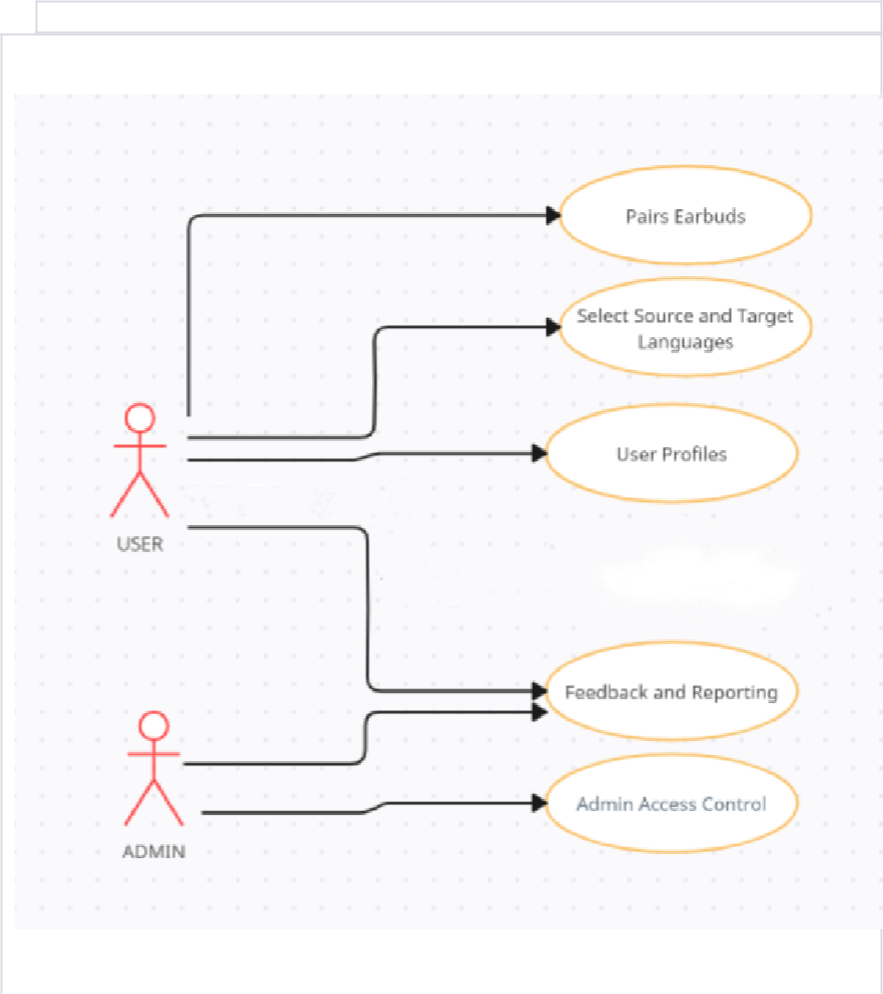
**Assignment Details: Design Document**

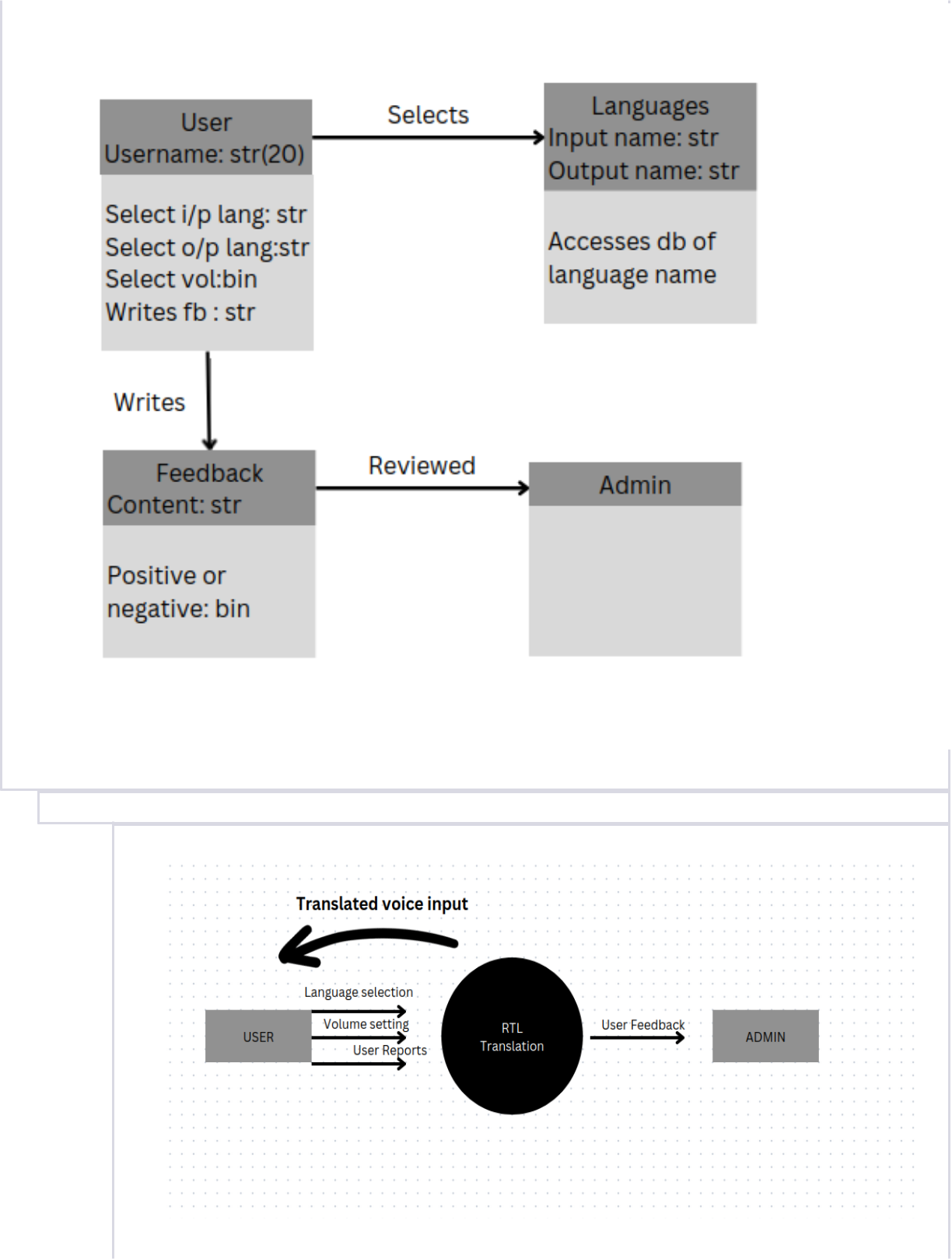
**Team Members:**

|  |  |
| --- | --- |
| Swayam Chhaba | PES2UG21CS566 |
| Tarun Sairaj Vepamaniti | PES2UG21CS572 |
| Vismay Rallabandi | PES2UG21CS612 |
| Y Teresha | PES2UG21CS618 |



1. **Create a Use case and Class Diagram**

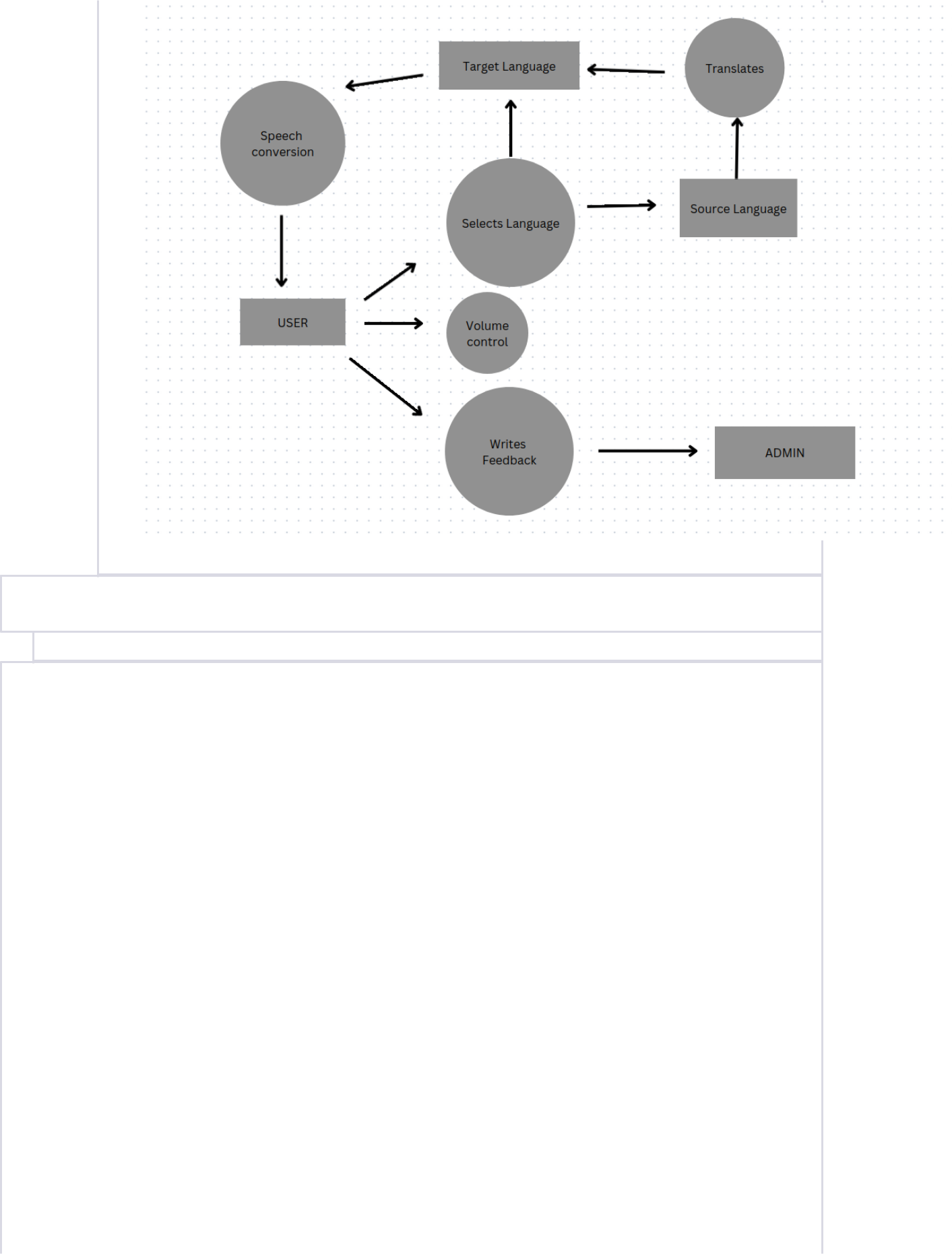
**Fig 1.1 Use Case Diagram**



**1.2 Class diagram**

1. **Incorporate DFDs:**
   * Develop a Data Flow Diagram (DFD) for your project:

Level 0 diagram 2.1



Level 1 diagram 2.2

1. **Architectural Style Integration:**

For the Real-Time Language Translation Earpiece project, the most suitable architectural style is the Client-Server architectural style. This choice aligns with the project's goals and requirements.

In the Client-Server architectural style, the system is divided into two primary components: the client and the server. The client represents the mobile application used by end-users to interact with the Real-Time Language Translation Earpiece. The server encompasses various backend services, APIs, and cloud-based components that facilitate language translation and system functionality.

=> Describe and justify the chosen architectural style for your system. Explain how this style is suitable for your system’s requirements and how it will beneft the project in terms of scalability, maintainability, performance, etc.

Chosen Architectural Style: Client-Server

Justifcation:

The Client-Server architectural style is the most appropriate choice for the Real-Time Language Translation Earpiece project due to several key factors that align with the project's requirements and objectives.

1. Scalability:

The project aims to provide real-time language translation services to a wide user base, including travelers, business professionals, language learners, and cultural enthusiasts. As the user base grows, the architecture allows for the easy addition of server resources to handle increasing translation demands. This scalability is essential for accommodating a global audience and future growth.

2. Centralized Control:

The project's success depends on effective language translation, user profle management, feedback processing, and data privacy. The Client-Server style offers centralized control over these critical components. Language translation services, data privacy measures, and feedback review can be effciently managed and monitored from the server. This centralized control ensures consistent service quality and user experience.

3. Effective Communication:

Real-time language translation requires effcient communication between the

client (mobile application) and the server (backend services). The

Client-Server style is well-suited for this purpose. It enables seamless

communication between the two components, ensuring that user input is

processed quickly and translated in real-time. This is fundamental to the

project's goal of breaking down language barriers and facilitating effortless

communication.

4. Data Privacy and Security:

Data privacy and security are paramount in a project that involves speech recognition, translation, and user profles. With the Client-Server architecture, data privacy measures can be concentrated within the server. This ensures that sensitive user data and translations are protected, and robust security protocols can be implemented to safeguard information.

5. Maintainability:

The separation of client and server components in the Client-Server style enhances maintainability. Updates and enhancements to the system can be made on the server side without requiring modifcations to the client application. This reduces development complexity and makes it easier to deliver updates, improvements, and new language models to users.

6. Performance:

Real-time translation demands effcient and high-performance processing.

The Client-Server style is optimized for this purpose. Backend services can

leverage the processing power of cloud-based solutions, and by having

dedicated server resources, the system can deliver rapid and accurate

translations without overburdening the user's mobile device.

7. User Experience:

For a project focused on enhancing user experience by breaking down language barriers, the Client-Server style provides the necessary infrastructure for seamless real-time translation. The centralized control, scalability, and effcient communication ensure that users can engage in multilingual conversations effortlessly.

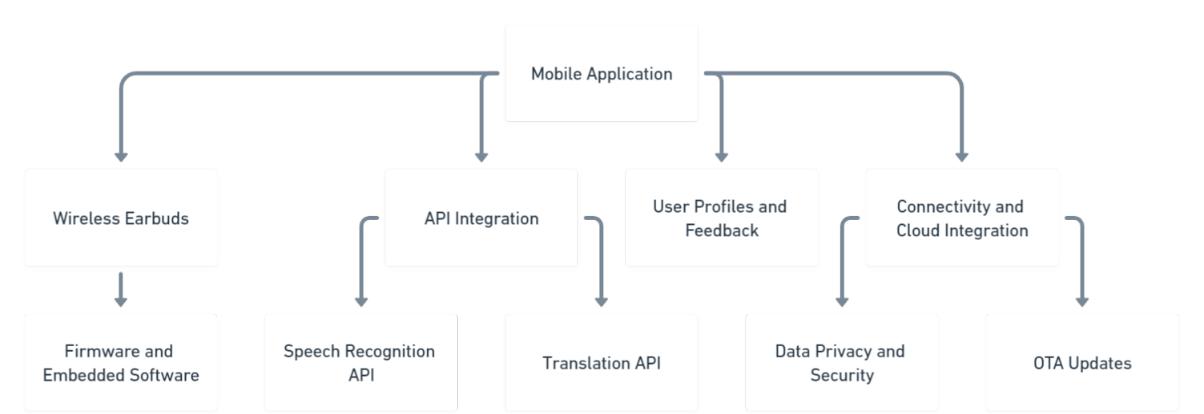


Fig 4.1 Architecture

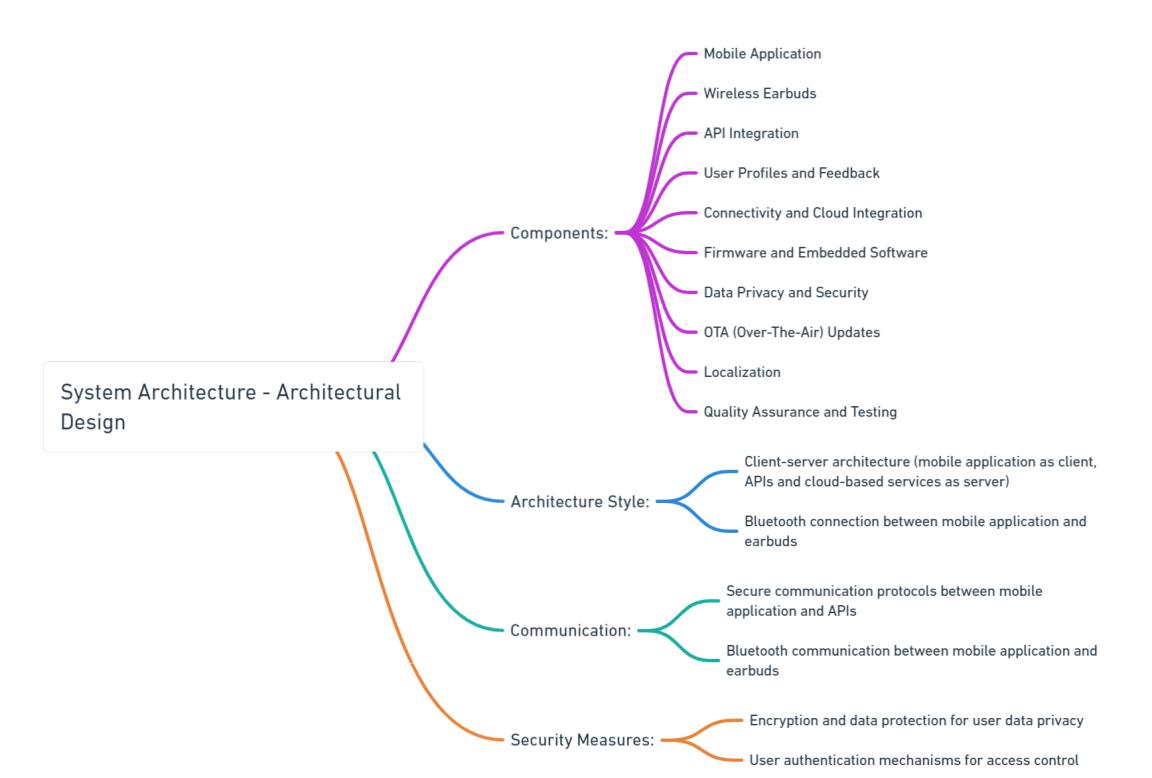


Fig 4.2 Architecture