**Architecture**

The Architecture of the Simple Banking Application can be described as a simple console-based application following a layered architecture approach. Here's an overview of the architecture:

**Presentation Layer:**

This layer is responsible for interacting with the user and displaying information.

It handles user input and presents the output on the console.

In the Simple Banking Application, the presentation layer is represented by the main method in the SimpleBankingApplication class and the usage of the Scanner class to capture user input.

**Business Logic Layer:**

This layer contains the core logic and functionality of the banking operations.

It performs calculations, validates user inputs, and updates the data accordingly.

In the Simple Banking Application, the business logic layer is represented by the checkBalance(), deposit(double amount), and withdraw(double amount) methods.

**Data Layer:**

The data layer is responsible for managing and storing the application's data.

In the Simple Banking Application, since the data is not persisted permanently, there is no dedicated data layer. The balance variable is stored in memory during the application's runtime.

**Interaction Flow:**

The user interacts with the presentation layer, providing input and receiving output.

The presentation layer passes the user's input to the business logic layer, which performs the required operations and updates the data.

The updated data is then returned to the presentation layer, which displays the relevant output to the user.

**Error Handling:**

The application includes basic error handling to handle invalid inputs, insufficient funds during withdrawals, and other potential errors.

Error messages are displayed to the user when errors occur, guiding them to take appropriate actions.

**Future Enhancements:**

The architecture allows for future enhancements, such as implementing user authentication and multiple accounts, adding a persistent data storage mechanism, and supporting additional banking operations.

These enhancements would involve introducing additional layers, such as an authentication layer and a data access layer, to handle more complex functionalities.

Overall, the Simple Banking Application follows a simple and straightforward architecture, focusing on the presentation layer for user interaction, the business logic layer for core operations, and the absence of a dedicated data layer for the temporary storage of data during runtime.