Software Architecture (40353103-1)

Assignment #1- Graded (15%)

Given Online Banking System / Digital Wallet to represent client interaction with banking system or digital wallet as follows

- **Client Presentation**: Web access to account/balance/transactions
- **Servers and Web Services (Physical or Virtual or Cloud)**:

    1. **Applications server (banking system and ATM services)**
    2. **WS1**: User Account Management
    3. **WS2**: Payment/Transfer Service
    4. **WS3**: Transaction History / Reporting

- **DB Server (Physical or Virtual or Cloud)**:

    1. **C1**: Secure banking DB (account, balance)
    2. **C2**: REST APIs for internal access

- **Extras**: HTTPS, FTP (statement downloads), caching (dashboard data), load balancing.

Q1: Software Architecture Design Tasks. You have to achieve the following 4 tasks with considering the applicable architecture and optimal QA

1. **Client-Server Based Architecture Design**

    - Include **Load Balancer** and **Caching Service** in your architecture.
    - Show all major modules and their interaction using HTTP/HTTPS and REST.
    - Clearly diagram how requests are routed through the Load Balancer and processed efficiently with caching.

2. **Workflow Architecture: From Request to Reply**

    - Diagram a complete user interaction:

        - A user sends a request.
        - Load Balancer routes it to a service.
        - The service queries the cache and DB.
        - A response is sent back through the stack.
    - Highlight caching and load balancing behavior during this flow.
3. **Module Description**
    - Provide a **detailed write-up** of:
        - Each major module (DB Server, App Server, Client Presentation).

        - All sub-components (C1, C2, WS1, WS2, WS3, Load Balancer, Cache).
        - Their individual functions and technology suggestions (optional).

4. **Communication Patterns (C&C)**
    - Show how components interact using connectors:
        - HTTP/HTTPS for RESTful APIs
        - FTP for file transfer
        - Internal connectors (e.g., between WS1 and Cache)
    - Define protocols, interface types, and directions.

Q2: Component & Connector View – Detailed Explanation

- Provide a **Component & Connector** view of the full architecture:

    1. Describe each component, its responsibilities, interfaces.
    2. Describe connectors and communication protocols (including LB and cache).
    3. Indicate synchronous/asynchronous calls where relevant.

Q3: Mapping Quality Attributes to Architecture

Explain how your architecture supports key **software quality attributes (if applicable)**:

Quality Attribute	Architectural Feature Supporting It
Performance	
Scalability	
Availability	
Security	
Modifiability	
Maintainability	
Cost Efficiency	