Project Design Document - ATM Protocol

1. Group Members

- **Member 1**: Abhay Kumar Upparwal (21110004)
- **Member 2**: Anushk Bhana (21110031)
- **Member 3**: Husain Malwat (21110117)
- **Member 4**: Naman Dharmani (21110136)
- **Member 5**: Sahil Das (21110184)
- **Member 6**: Srujan Kumar Shetty (21110214)
- **Member 7**: Yashraj J Deshmukh (21110245)

2. Languages/Frameworks

- **Programming Languages**:
  - Python for both ATM and Bank programs.
- **Frameworks/Libraries**:
  - SQLAlchemy - ORM model for database connection.
  - PostgreSQL as the primary database.
  - Rate Limiters to prevent DOS attacks.
  - Alternatively, JWT
  - OS Library for handling system calls and network programming.
- **Build System**:
  - Gradle, Maven, or Ninja (for compiling and managing dependencies)

3. System Architecture

The system consists of 3 main components: the **ATM Client**, the **Bank Server**, and the **Database.**

ATM Client

- **Network Interface**: Establishes and maintains secure connections with the Bank Server, ensuring reliable communication and data transfer.
- **Authentication Module**: Verifies the authenticity of the ATM using the auth file, ensuring only authorized devices can access the Bank Server.
- **Transaction Processing**: Handles user interactions, including:
  - Account creation
  - Deposit and withdrawal transactions
  - Balance inquiries
  - Account maintenance

Bank Server

- **Connection Manager**: Listens for incoming TCP connection requests from ATM clients on a specified port, establishing new connections as needed and managing existing connections.
- **Authentication Service**: Verifies the authenticity of the ATM client using the auth file, ensuring only authorized devices can access the Bank Server.
- **Transaction Processor**: Manages account data, handles transactions, and maintains balance integrity, ensuring accurate and secure processing of user requests.
- **Security Framework**: Ensures secure communication and protects sensitive data, implementing measures such as encryption, access controls, and secure protocols.

Database

- **Data Storage**: Manages the storage and retrieval of account data, additionaly:
    - Account numbers
    - Card numbers
    - Current balances
    - Authentication files
- Ensures data persistence and supports the Bank Server in handling transactions and authentication.

4. Functionalities and Responsibilities

- **ATM Client**:
    - **Account Creation**: Allows customers to open a new bank account with an initial balance.
    - **Deposit**: Allows customers to deposit money into their accounts.
    - **Withdrawal**: Allows customers to withdraw money from their accounts, ensuring they don't withdraw more than the available balance.
    - **Balance Check**: Allows customers to check their account balance.
- **Bank Server**:
    - **Transaction Handling**: Processes and verifies transactions (deposit, withdrawal, balance check) received from the ATM.
    - **Secure Communication**: Ensures that communication with the ATM is secure and encrypted using the auth file.
    - **User Verification**: Verifies the user with the help of its user id and associated card file.
    - **Account Management**: Manages customer accounts and ensures data integrity.
    - **Database Management**: Establishes a connection with the database to fetch and update account balances.Once the request is completed.
    - **Key Management**: Generates, stores, and rotates cryptographic keys for secure communication and data encryption, ensuring high security and regulatory compliance.

5. Security Considerations

- **ATM Client**:
    - **Authentication**: The ATM will use the auth file to authenticate with the bank before performing any transactions.
    - **Data Encryption**: All communication between the ATM and the Bank will be encrypted using SSL/TLS to prevent man-in-the-middle attacks.
    - **Card File Security**: The card file is treated as sensitive information, and measures will be implemented to prevent unauthorized access.
- **Bank Server**:
    - **Authentication Verification**: The server will verify the authenticity of the ATM client before processing any transactions.
    - **Card File Verification**: The server will verify the card file of the user to verify its authenticity.
    - **Data Integrity**: All transactions will be atomic, ensuring that partial transactions do not corrupt account balances.
    - **Error Handling**: The server will handle protocol errors gracefully, ensuring no unauthorized access or data corruption occurs.

6. Additional Features:

- **Daily Transaction Limit per User**: Enforce a maximum number of transactions allowed per user each day to mitigate excessive or fraudulent activity.
- **Temporary Account Locking After Failed Attempts**: Implement a mechanism to temporarily lock user accounts after a specified number of consecutive failed login attempts.
- **Database Failure Recovery**: Establish a robust recovery process for database failures to ensure minimal data loss and quick restoration of services.
- **Transaction Logging**: Maintain detailed logs of all transactions to enable troubleshooting.