To design an application that securely monitors the status and behaviour of ATMs within a bank’s network in real time, will require following elements -

1. **System Design**
2. **Component Design**
3. **API Design**
4. **Activity Flow Diagrams**
5. **Data Model Design**
6. **Data Model Examples**
7. **Pseudo Code**
8. **Task Breakdown and Development Progression**
9. **Additional API Endpoints**

**1. System Design**

**1.1 Overview**

The application will consist of a client-side **ATM Monitoring Agent** that runs on each ATM device. This agent will communicate with a **Bank Monitoring Server** through secured APIs, providing real-time data on the ATM’s status and behaviour, while also receiving system updates, configuration changes, and commands from the server.

**1.2 Key Requirements**

* **Real-time monitoring** of ATM hardware, software, and network status.
* **Secure communication** with centralized bank systems, using tokens for authentication.
* Collection and transmission of customer transaction logs, video footage logs, and other diagnostics.
* Provide APIs for other applications within the bank’s network to interact with the ATM monitoring service.

**1.3 High-Level Components**

1. **ATM Monitoring Agent**: Installed on the ATM, collects status and sends data to the bank server.
2. **Bank Monitoring Server**: Receives and processes data, logs information, sends commands.
3. **Bank Authentication System**: Issues authentication tokens for secure API communication.

**2. Component Design**

**2.1 ATM Monitoring Agent**

* **Modules**:
  + **Hardware Status Module**: Monitors ATM hardware (cash dispensers, card readers, etc.).
  + **Software Status Module**: Checks OS status, application health, software updates.
  + **Network Status Module**: Monitors the ATM's network connection and bandwidth.
  + **Logger**: Collects logs for customer transactions and video footage.
  + **API Client**: Communicates with the Bank Monitoring Server using secured HTTP APIs.

**2.2 Bank Monitoring Server**

* **Modules**:
  + **Status Receiver**: Receives real-time updates from ATMs.
  + **Command Center**: Sends configuration commands, software updates, and shutdown/reboot instructions.
  + **Log Manager**: Stores transaction and video logs in the database.
  + **API Authentication**: Verifies authentication tokens before accepting requests.
  + **Dashboard**: Visual interface for real-time monitoring of all ATMs.

**2.3 Bank Authentication System**

* **Token Issuer**: Provides verifiable authentication tokens.
* **Token Validator**: Ensures each request from ATMs is properly authenticated before processing.

**3. API Design**

**3.1 Authentication and Security**

* **Token-based authentication**: The bank’s authentication system issues tokens. Each request must include this token in the Authorization header.
* **HTTPS** for secure transmission of data.

**3.2 Endpoints**

**3.2.1 ATM Status API**

**Endpoint**: /api/atm/status  
**Method**: POST  
**Description**: Sends the current status of the ATM (hardware, software, network) to the server.  
**Request Body**:

{

"atmId": "ATM123",

"hardwareStatus": "OK",

"softwareStatus": "Running",

"networkStatus": "Connected",

"transactionLogs": [

{

"transactionId": "TX456",

"timestamp": "2024-10-20T12:34:56Z",

"status": "Completed"

}

],

"videoLogs": [

{

"videoId": "VID789",

"timestamp": "2024-10-20T12:34:56Z",

"duration": 30

}

]

}

**Response:**

{

"message": "Status received",

"nextAction": "No action"

}

**3.2.2 ATM Command API**

**Endpoint**: /api/atm/command  
**Method**: POST  
**Description**: Sends commands (restart, update, etc.) to the ATM.  
**Request Body**:

{

"atmId": "ATM123",

"command": "RESTART"

}

**Response**:

{

"message": "Command sent"

}

**3.2.3 Token Validation API**

**Endpoint**: /api/token/validate  
**Method**: POST  
**Description**: Validates the token provided by the ATM during communication.  
**Request Body**:

{

"token": "valid-token-123"

}

**Response**:

{

"valid": true

}

**3.3 Additional API Endpoints**

1. **Software Update API**: Used to send software updates to ATM devices.
2. **ATM Error Logs API**: Allows ATMs to send error logs to the central server.

**4. Activity Flow Diagrams**

**4.1 ATM Monitoring Process**

ATM Device -> Bank Monitoring Server:

1. Monitor hardware, software, and network status.

2. Collect logs for transactions and video footage.

3. Periodically send status updates to Bank Monitoring Server.

**4.2 Token Authentication Flow**

Bank Monitoring Server <-> Bank Authentication System:

1. ATM requests authentication token.

2. Bank system issues token.

3. ATM uses token to authenticate API requests.

**5. Data Model Design**

**5.1 Data Models**

1. **ATM Status**:
   * atmId: Unique identifier for each ATM.
   * hardwareStatus: Status of hardware (OK, Error).
   * softwareStatus: Status of software components.
   * networkStatus: Current network status (Connected, Disconnected).
   * transactionLogs: Collection of transaction logs.
   * videoLogs: Collection of video logs.
2. **Transaction Log**:
   * transactionId: Unique identifier for the transaction.
   * timestamp: When the transaction occurred.
   * status: Status of the transaction (Completed, Failed).
3. **Video Log**:
   * videoId: Unique identifier for the video footage.
   * timestamp: Time of the video recording.
   * duration: Length of the video in seconds.

**6. Data Model Examples**

**ATM Status Example**

{

"atmId": "ATM123",

"hardwareStatus": "OK",

"softwareStatus": "Running",

"networkStatus": "Connected",

"transactionLogs": [

{

"transactionId": "TX456",

"timestamp": "2024-10-20T12:34:56Z",

"status": "Completed"

}

],

"videoLogs": [

{

"videoId": "VID789",

"timestamp": "2024-10-20T12:34:56Z",

"duration": 30

}

]

}

**7. Pseudo Code**

**ATM Monitoring Agent (Client-Side)**

START

Initialize ATM Monitoring Agent

WHILE True:

Get ATM hardware status

Get ATM software status

Get network status

Collect transaction logs

Collect video logs

IF Status or logs updated:

SEND status update to Bank Monitoring Server

WAIT for next interval

END

**Bank Monitoring Server (Server-Side)**

START

Initialize Bank Monitoring Server

WHILE True:

RECEIVE status update from ATM

VALIDATE authentication token

STORE status update and logs in the database

IF commands to send:

SEND commands to ATM

END

**8. Task Breakdown**

1. **Planning and Requirements Gathering**: 1 week
2. **System and Component Design**: 2 weeks
3. **API Design and Documentation**: 1 week
4. **Development Phase**:
   * ATM Monitoring Agent: 3 weeks
   * Bank Monitoring Server: 3 weeks
   * Bank Authentication System Integration: 2 weeks
   * API Security and Validation: 1 week
5. **Testing**: 2 weeks
6. **Deployment**: 1 week
7. **Monitoring and Maintenance**: Ongoing

**9. Additional API Endpoints**

**9.1 Software Update API**

**Endpoint**: /api/atm/update  
**Method**: POST  
**Description**: Sends software updates to the ATM.  
**Request Body**:

{

"atmId": "ATM123",

"updateVersion": "v2.3.1",

"downloadUrl": "https://updates.example.com/atm/v2.3.1"

}

**Response**:

{

"message": "Update received, ATM will install in next maintenance window."

}

**9.2 Error Logs API**

**Endpoint**: /api/atm/error-logs  
**Method**: POST  
**Description**: Allows the ATM to send error logs to the server.  
**Request Body**:

{

"atmId": "ATM123",

"errorCode": "ERR500",

"errorDescription": "Cash dispenser failure",

"timestamp": "2024-10-20T12:45:00Z"

}

**Response**:

{

"message": "Error log received"

}