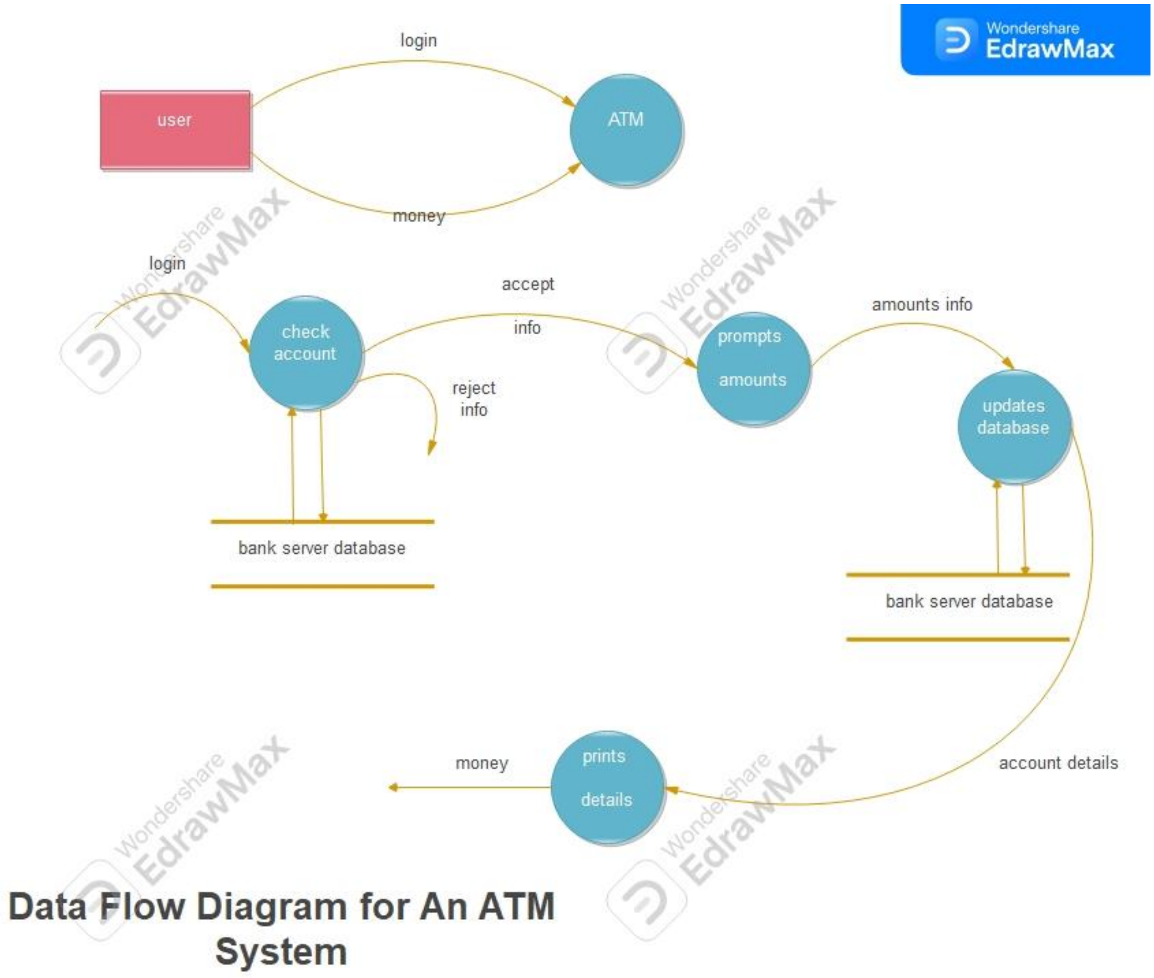
Level 1 Data Flow Diagram (DFD) for an **ATM system**



Question:

Design a **Level 1 Data Flow Diagram (DFD)** for an **ATM system** that illustrates the following processes:

1. User login and account verification.
2. Checking account balance.
3. Handling user requests for cash withdrawal or other transactions.
4. Updating the bank's database after a successful transaction.
5. Providing feedback to the user through prompts, receipts, or error messages.

Instructions:

- Identify and include the following components:
    - External entities (e.g., User).
    - Processes (e.g., Login, Check Account, Update Database, etc.).
    - Data stores (e.g., Bank Server Database).
    - Data flows between components.
- Clearly label all components and ensure the data flow represents a complete transaction lifecycle.

Hint: Refer to the functional steps of an ATM system and structure the diagram to show the interaction between the user, the ATM system, and the bank's backend database.

Description of the ATM System

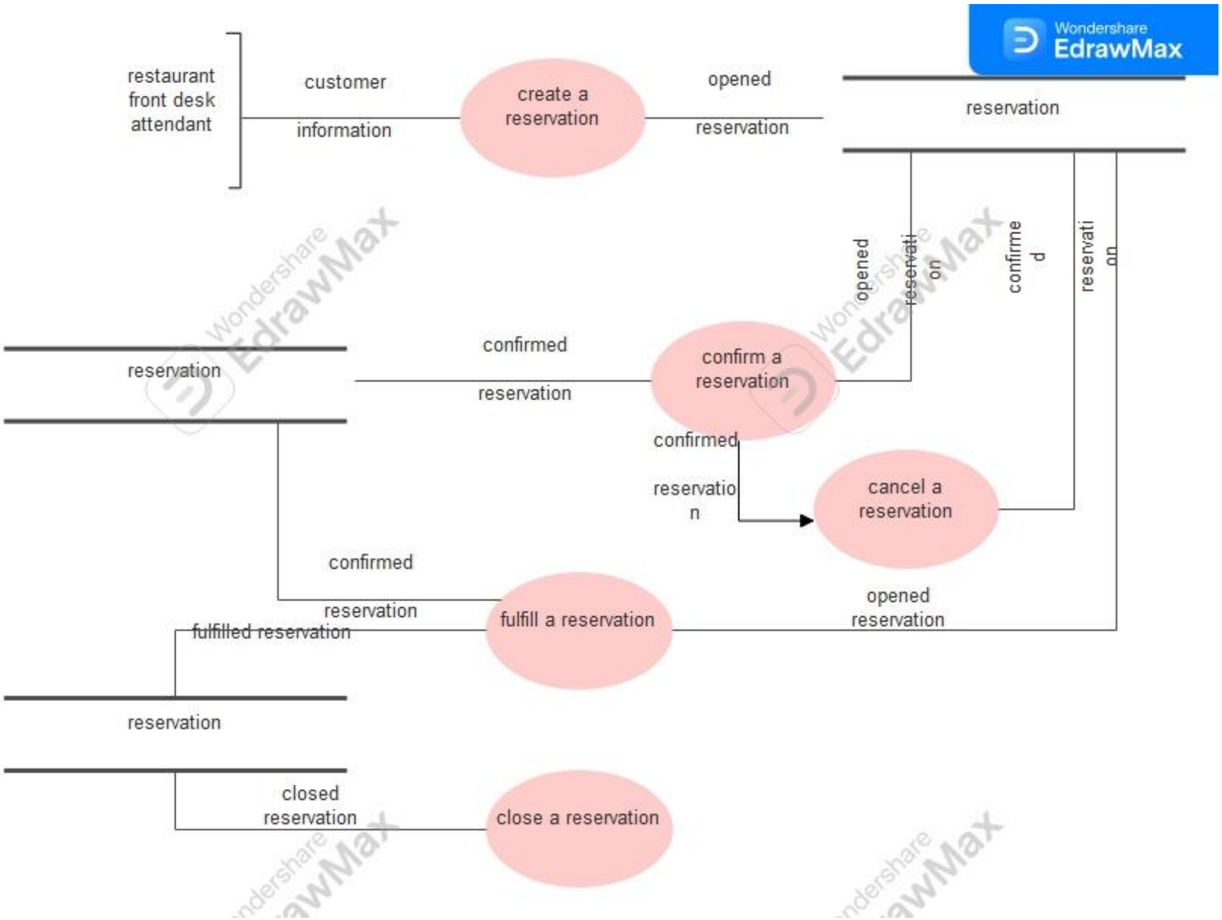
The ATM system facilitates interactions between a user and a bank's backend infrastructure to provide essential banking services. Here's how the system operates:

1. **User Interaction:**
    - The process begins when the user logs in to the ATM.
    - The user provides credentials and requests services such as checking the account balance, withdrawing money, or printing transaction details.
2. **ATM Functionality:**
    - The ATM acts as an intermediary between the user and the bank's database.
    - Upon receiving the user's request, it initiates communication with the bank's server to validate the user's account and execute transactions.
3. **Account Verification (Check Account):**
    - The ATM sends the user's login information to the **bank server database** to verify the account's authenticity.
    - If the account verification fails, the server rejects the request and sends an error message back to the ATM.
    - If the account is valid, the system proceeds to process the user's request.
4. **Prompts and Amount Handling:**
    - The system generates prompts for the user (e.g., "Enter withdrawal amount").
    - Once the user enters the amount, the system checks the account's balance information through the **bank server database**.
5. **Database Updates:**
    - After a successful transaction (e.g., withdrawal), the system updates the **bank server database** with the new account balance and transaction details.
6. **Printing and Money Dispensing:**

    - The ATM dispenses the requested amount of money to the user if sufficient funds are available.
    - The system also prints a receipt containing transaction details for the user's records.
7. **System Feedback:**
    - Throughout the process, the system provides real-time feedback to the user, such as acceptance, rejection of requests, or transaction completion details.

Key Components in the Diagram

- **Entities:**
    - **User**: The primary actor interacting with the system.
    - **ATM**: The device facilitating the interaction and forwarding requests to the backend.
- **Processes:**
    - **Check Account**: Verifies the user’s account credentials and balance.
    - **Prompts Amounts**: Prompts the user for input like withdrawal amount.
    - **Updates Database**: Updates the bank's database after a successful transaction.
    - **Prints Details**: Generates a receipt for the user.
- **Data Stores:**
    - **Bank Server Database**: The central repository for account and transaction information.



Description of the Restaurant Reservation System

The system manages the creation, confirmation, cancellation, fulfillment, and closure of restaurant reservations. It ensures smooth interaction between the **customer**, the **restaurant front desk attendant**, and the system's reservation database.

Key Components and Processes

1. **Entities:**
   - **Customer**: The primary user who provides reservation details and interacts with the system.
   - **Restaurant Front Desk Attendant**: The intermediary who assists with reservations, confirmations, or cancellations.
2. **Processes:**
   - **Create a Reservation**:
     - The customer provides reservation details to the front desk attendant.

        - The system generates an **opened reservation** and stores it in the reservation database.
    - **Confirm a Reservation**:
        - After the reservation is created, it is confirmed, changing its status to **confirmed reservation** in the database.
    - **Cancel a Reservation**:
        - The customer or the front desk attendant can cancel an open or confirmed reservation.
        - The reservation status is updated in the system.
    - **Fulfill a Reservation**:
        - A confirmed reservation is fulfilled once the customer avails the reserved service.
        - The system marks it as a **fulfilled reservation**.
    - **Close a Reservation**:
        - After fulfilling the reservation, it is closed, and the system updates it as a **closed reservation** in the database.
3. **Data Stores:**
    - **Reservation Database**: The central repository that keeps records of reservations in various states (opened, confirmed, fulfilled, and closed).
4. **Data Flows:**
    - Information flows between the **customer** and the **restaurant front desk attendant** to create, confirm, or cancel reservations.
    - Reservation details flow between processes and the **reservation database** to update the reservation's status.

Workflow Summary

1. The customer provides reservation information to the front desk.
2. The system creates a reservation and stores it in the database.
3. The reservation can be:
    - Confirmed,
    - Canceled, or
    - Fulfilled.
4. After fulfillment, the reservation is closed.

This DFD represents a structured and efficient process for managing restaurant reservations, ensuring clarity in customer interactions and database updates.