**Level 0 (Context Diagram)**

At the highest level, we represent the system as a whole.

+-----------------------+

| External Entity |

| - User |

+-----------------------+

|

v

+------------------------+

| Login System |

| (Process: 1.0) |

+------------------------+

|

v

+------------------------+

| Authentication System |

+------------------------+

**Explanation:**

* **External Entity (User)**: The user interacts with the login system by entering a login ID.
* **Process (Login System)**: The system processes the input and verifies the login credentials.
* **Data Store (Authentication System)**: The system checks whether the entered ID matches the stored login credentials.

**Level 1 DFD (Decomposition of Process 1.0 - Login System)**

+-----------------------+ +-----------------------------+

| External Entity | | Data Store |

| User | | (Authentication System) |

+-----------------------+ +-----------------------------+

| |

v v

+-----------------------+ +-------------------------------+

| Process: 1.1 - Input | | Process: 1.2 - Verify Login |

| User Login ID | | Validate ID against stored |

+-----------------------+ | credentials in the system |

| +-------------------------------+

v |

+------------------------+ |

| Process: 1.3 - Display|<---------------+

| Login Status |

+------------------------+

**Explanation:**

* **Process 1.1 (Input User Login ID)**: The user enters a login ID in the input field.
* **Process 1.2 (Verify Login Credentials)**: The system checks whether the input ID matches the stored LOGIN\_ID.
* **Process 1.3 (Display Login Status)**: The system updates the UI to show whether the login was successful or not.

**Data Flow:**

1. **User enters the login ID** in the input field.
2. **The system verifies the entered ID** against the stored LOGIN\_ID.
3. **If the ID matches**, a success message is displayed.
4. **If the ID does not match**, an error message prompts the user to try again.

**Additional Notes:**

* This system is a basic authentication simulation and does not involve a backend or secure authentication methods.
* In a real-world application, login credentials should be validated through a backend API, not stored in the frontend.
* Future improvements could include password authentication, session handling, and API-based login verification.