**Project Description:**

The **Counter Application System** tracks user clicks and updates a count dynamically. Users click a button, triggering an event that increments the count, which is then stored and displayed. The system ensures real-time updates using React’s useState hook.

### **Level 0 (Context Diagram)**

At the highest level, we have just one process, which represents the system as a whole.

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

| External Entities |

| - User |

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

|

|

v

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

| Counter System |

| (Process: 1.0) |

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

|

|

v

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

| Data Store |

| (Click Count) |

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

**Explanation**:

* **External Entity (User):** The user interacts with the counter system by clicking a button.
* **Process (Counter System):** The system processes the click event and updates the count.
* **Data Store (Click Count):** Stores the updated click count.

### **Level 1 DFD (Decomposition of Process)**

Now, let's break down the **Counter application System** process (Process 1.0) into more detailed steps.

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

| External Entity | | Data Store |

| (User) | | (Click Count) |

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

| |

v v

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

| Process: 1.1 - Click | | Process: 1.2 - Update Count |

| Button Event | | (Increment by 1) |

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

|

v

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

| Process: 1.3 – Display |

| Updated Count |

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

**Explanation**:

### **Process 1.1 (Click Button Event):** User clicks the button to increment the count***.***

### **Process 1.2 (Update Count):** The system updates the click count by increasing it by 1.

### **Process 1.3 (Display Updated Count):** The updated count is displayed on the screen.

### **Data Flow**

* The user clicks the button.
* The system captures the click event.
* The count is incremented and stored.
* The updated count is displayed.

### **Notes:**

* The system updates the count dynamically using React’s use State hook.
* Future enhancements could include storing the count persistently in a database or local storage.
* Additional features like a reset button or decrement functionality could be added.