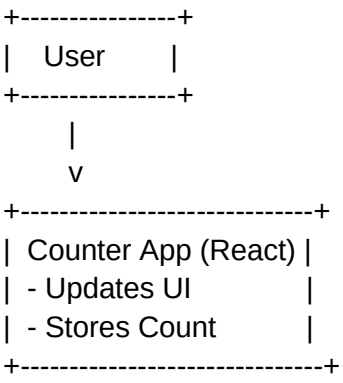
Data Flow Diagram (DFD) for React Counter App (Increment & Decrement)

This Counter App allows the user to increment or decrement a counter value using React's useState hook.

Level 0 (Context Diagram)

At the highest level, the **user interacts** with the **Counter App**, which updates the count state and displays the result.

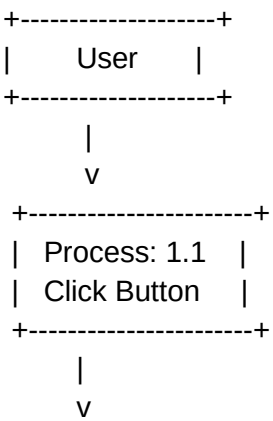


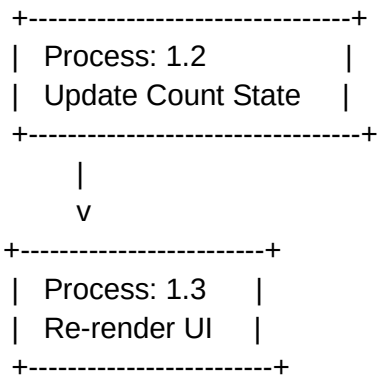
Explanation:

- The User interacts with the Counter App by clicking the Increment and Decrement buttons.
- The Counter App processes user actions and updates the displayed count.

Level 1 DFD (Decomposition of Process)

Breaking down the **Counter App** into core processes:





Explanation:

1. **Process 1.1 - Click Button**
    - The user clicks the Increment or Decrement button.
2. **Process 1.2 - Update Count State**
    - setCount(count + 1) (for increment) or setCount(count - 1) (for decrement) updates the state.
3. **Process 1.3 - Re-render UI**
    - React re-renders the component, displaying the updated count.

Data Flow

Entities:

User – Clicks the increment or decrement button.

1. Counter App (React Component) – Manages state and updates the UI.

Processes:

1. Click Button – Triggers the function to update count.
2. Update Count State – Increments or decrements the count in React state.
3. Re-render UI – Displays the updated count.

Data Stores:

1. React State (count) – Stores the counter value.

Additional Notes & Improvements

- **Optimizations:**
    - If multiple clicks happen quickly, use functional updates:

      setCount(prevCount => prevCount + 1);

      setCount(prevCount => prevCount - 1);
- **Enhancements:**
    - Add a Reset button to set the counter back to 0.
    - Store the count value in local storage to persist after refresh.
    - Add a minimum limit (e.g., prevent count from going below 0).