**Project Description:**

The **State Management Application System** efficiently handles navigation, theme switching, and lazy-loaded components using React, React.lazy(), and Suspense. Users select pages (Home, About, Contact), triggering lazy loading for optimized performance. Theme changes (Light/Dark) are managed via Theme Context, dynamically updating the UI. A Loader (Fallback Component) provides a smooth transition while modules load asynchronously.

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

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

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

| External Entities |

| - User |

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

|

|

v

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

| React App |

| (Process: 1.0) |

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

|

|

v

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

| Lazy Loaded Modules |

| (About, Contact) |

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

**Explanation**:

* **External Entity (User):** The user interacts with the React application.
* **Process (React App):** The system manages navigation, theme switching, and lazy loading.
* **Lazy Loaded Modules (About, Contact):** Components that are dynamically loaded using React Lazy() and Suspense**.**

### **Level 1 DFD (Decompos ition of Process)**

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

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

| External Entity | | Data Store |

| (User) | | (Theme Context) |

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

| |

v v

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

| Process: 1.1 - Select | | Process: 1.2 - Toggle |

| Page (Home, About) | | Theme (Light/Dark) |

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

| |

v v

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

| Process: 1.3 - Load | | Process: 1.4 - Apply |

| Component (Lazy Load) | | Theme UI Changes |

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

|

v

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

| Loader (Fallback Component) |

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

**Explanation**:

### **Process 1.1 (Select Page):** The user selects a page (Home, About, or Contact).

### **Process 1.2 (Toggle Theme):** The user switches between Light and Dark themes using the theme toggle button.

### **Process 1.3 (Load Component):** The app uses React.lazy() and Suspense to load components dynamically.

### **Process 1.4 (Apply Theme UI Changes):** The app updates the UI based on the selected theme.

### **Loader (Fallback Component):** If a module is not yet loaded, a loading message is displayed.

### **Data Flow**

* The user opens the React app.
* The app initializes with the current theme.
* The user selects a page (Home, About, or Contact).
* The selected component is lazy-loaded and displayed.
* If the user toggles the theme, the app updates the UI accordingly.
* The fallback loader displays a message until the component is fully loaded.

### **Notes:**

* Lazy loading reduces initial page load time.
* Suspense provides a fallback while components load asynchronously.
* Theme switching enhances user experience with dynamic UI updates