**Business Requirement Document (BRD) Error Boundary in React**

1. Objective

Implement an Error Boundary using componentDidCatch in a React application to catch runtime errors in child components, log them via reportError from utils.js, and display a fallback UI instead of crashing the entire application.

1. Functional Requirements

ErrorBoundary Component:

* Uses componentDidCatch to handle errors in child components
* Calls reportError from utils.js to log errors
* Displays a fallback UI ('There was a problem') when an error occurs

BombButton Component:

* Renders a button with a bomb emoji (💣)
* Has proper accessibility attributes (role="button", aria-label)
* Contains a span element with role="img" and aria-label="bomb"
* When clicked, triggers an error state
* Uses ErrorBoundary to prevent app crashes

1. Acceptance Criteria

Component Rendering:

* BombButton renders correctly with bomb emoji
* Proper accessibility attributes are present
* Button is clickable and interactive

Error Handling:

* App continues running after an error occurs
* Error message ('There was a problem') is shown instead of crashing
* reportError function is called with error details
* Console errors are properly managed and don't flood the console

Accessibility Requirements:

* Button has proper role and aria-label
* Emoji is wrapped in span with role="img"
* All interactive elements are keyboard accessible

State Management:

* Error state is properly maintained
* Component recovers after error resolution
* State changes don't cause unnecessary re-renders

1. Testing Requirements

Component Testing:

* Initial render state verification
* Error state transition testing
* Error reporting verification
* Accessibility compliance testing

Environment Compatibility:

* Tests must pass in local development environment
* Tests must pass in CI/CD pipeline
* Consistent behavior across environments

1. Risks & Mitigation

Risks                         Mitigation

Uncaught errors              Ensure key components are wrapped in ErrorBoundary Performance impact           Optimize error handling to minimize re-renders Inconsistent test behavior   Implement robust, environment-agnostic tests Accessibility issues         Regular accessibility audits and testing

1. Technical Specifications

File Structure:

* Component location: src/component-did-catch.js
* Utils location: src/utils.js
* Test file location: src/tests/

Dependencies:

* React Testing Library
* Jest
* Required utility functions (reportError)

This implementation ensures that errors are handled gracefully, improving app stability and user experience while maintaining error logging and accessibility standards.