SimpleForm Data Flow

1. Introduction

This document provides a detailed Data Flow Diagram (DFD) representation of the SimpleForm React application.

It explains how form handling, state management, and UI updates are managed dynamically.

2. System Overview

The React application consists of the following key components:

- App.js: The main file that initializes the form, manages input fields, and handles form submission.
- useState Hook: Maintains the form input values and submission state.
- Form Component: Displays the form and processes user input.

3. Data Flow Diagram (DFD)

Level 0 (Context Diagram)

At the highest level, the system consists of external users interacting with the form.

External Entities:

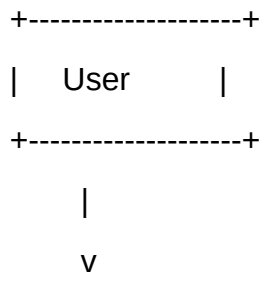
- User: Provides input (name and email) and submits the form.

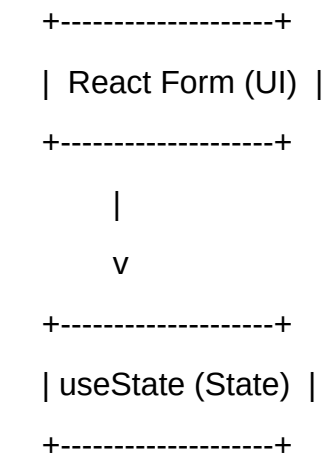
Processes:

- React Application: Handles input changes, manages state, and updates the UI.

Data Stores:

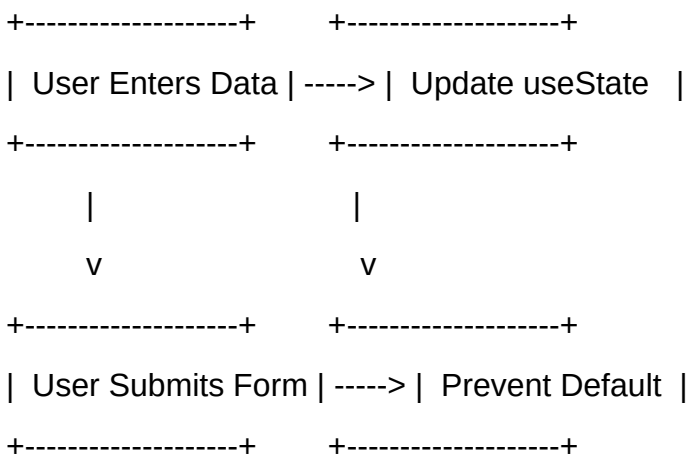
- useState Hook: Stores form data and submission status.





Level 1 DFD (Form Submission Process)

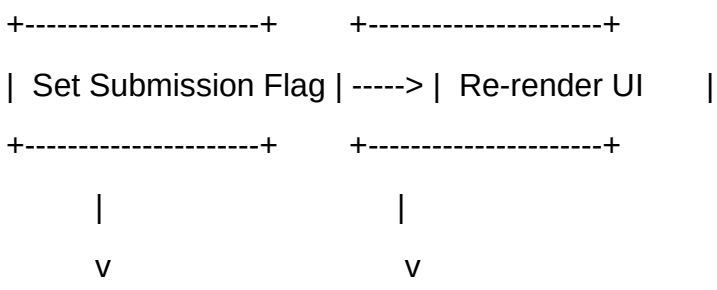
This level breaks down how user input is processed and the form submission is handled.

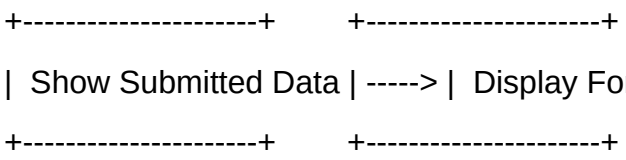


Process Explanation:

1. The user enters their name and email.
2. The input values are updated using `useState`.
3. When the form is submitted, the default form submission is prevented.

Level 2 DFD (State Update and UI Changes)





Process Explanation:

1. When the form is submitted, `setSubmitted(true)` is triggered.
2. React detects the state change and re-renders the component.
3. The submitted name and email are displayed instead of the form.

4. Explanation of Data Flow

1. The user interacts with input fields in the form.
2. The component manages input state using `useState`.
3. When submitted, the form prevents default behavior and updates the submission state.
4. The application conditionally renders the submitted data.

5. Conclusion

This document outlines the detailed data flow in a simple React-based form handling application.
The application dynamically manages form state and user interactions using `useState`.