Track Zone Project

****

Project Name: Track Zone Application

Submitted To : Mr. Md. Monir Hossain (Assistant Professor)

Submitted By : Md. Kamrul Hasan

Roll : 79

Section : A

Lab Group : Q

****

Requirements

- User can set their own time and timezone, this clock can’t be deleted only be edited
- User can create as many clock as they want
- Each clock has their own title or name
- Own Timezone
- Simple Events with time
- Time difference between users timezone and clock timezone in hour and minute
- User can edit or delete a clock
- Timezone could be UTC (standard), GMT, PST, EST
- Only date-fns library is allowed for this project. Rest of the logic should write by yourself
- Every data must be validated

Software Requirement Specification (SRS)

1. Introduction

1.1 Purpose

The **Track Zone Project** is a web-based application built using React that allows users to create and manage multiple clocks with different timezones. The project aims to test and enhance React skills by implementing core functionalities without using external state management libraries or a backend.

1.2 Scope

- Users can create multiple clocks, each with its own title and timezone.
- Users can set their own primary clock, which cannot be deleted but can be edited.
- Users can add simple events to specific clocks.
- The system calculates the time difference between the user's timezone and the clock's timezone.
- Timezones supported: UTC, GMT, PST, EST.
- The interface will be styled using `styled-components`.
- The `date-fns` library will be used for date and time calculations.

2. Functional Requirements

2.1 User Management

- The application does not require authentication or user management.

2.2 Clock Management

- We will have a local clock and a list of clocks
- We will create the initial clock from user timezone
- Clock Object will look like
  - id
  - title
  - timezone
    - type (UTC, GMT, PST, EST)
    - offset (Only for UTC and GMT)
  - events

2.3 Event Management

- Users can add simple events associated with a specific clock.
- Event object will look like
  - id
  - text
  - clockId
  - timezone

    - startTime
    - endTime
- We will use a clock object for local clock
- Use an array of clocks for other clocks
- We will use event id to create events inside clock

2.4 Clock Features

- properties
- update clock
- delete clock
- calculate difference
- update events

2.5 Event Features

- properties
- create event
- delete event
- update event
- filter event by id
- get events by ids

2.6 Time Difference Calculation

- The application calculates the time difference (in hours and minutes) between the user's selected timezone and each added clock.

3. Non-Functional Requirements

3.1 Performance

- The application should run efficiently in modern web browsers.
- All time calculations should be optimized for performance.

3.2 Usability

- The UI should be clean and user-friendly.
- Styled using `styled-components`.

3.3 Maintainability

- Code should be modular and reusable.
- All core logic should be written manually except for date/time operations handled by `date-fns`.

4. Technologies Used

- **Frontend:** React
- **State Management:** `useState`, `useEffect`
- **Styling:** `styled-components`
- **Date/Time Handling:** `date-fns`

5. Constraints

- No backend or external state management libraries (e.g., Redux, Context API) will be used.
- Only `date-fns` is allowed for date and time operations.

6. Future Enhancements

- Support for additional timezones.
- Persistent storage using local storage.
- Enhanced event management with notifications.

7. Assumptions and Dependencies

7.1 Assumptions

- Users will use modern browsers that support ES6+ and React.
- Users will have a stable internet connection.

7.2 Dependencies

- React.js for the frontend.
- `date-fns` and `date-fns-tz` for time and timezone management.
- `styled-components` for styling.

8. Glossary

- **Timezone**: A region of the globe that observes a uniform standard time.
- **UTC**: Coordinated Universal Time, the primary time standard.
- **GMT**: Greenwich Mean Time, often used interchangeably with UTC.
- **PST**: Pacific Standard Time, UTC-8.
- **EST**: Eastern Standard Time, UTC-5.

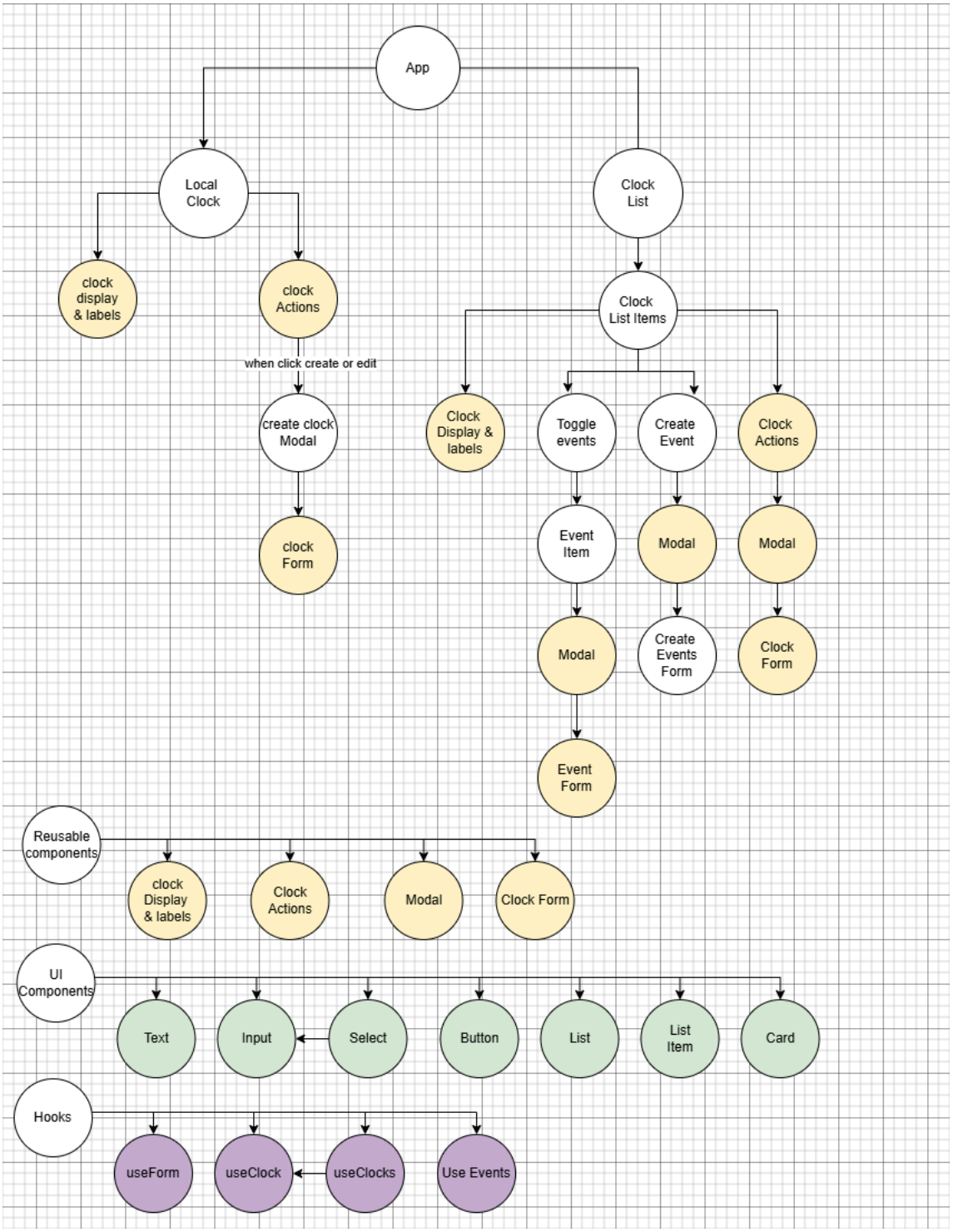
System Design

Creating a **Component Tree** and a **Data Flow Diagram** is essential for visualizing the structure of our React application and understanding how data flows between components. Below, I'll provide both diagrams for the **Track Zone Project**.

---

1. Component Tree

The **Component Tree** represents the hierarchy of components in your React application. Here's how the components are structured:



Component Tree Description

1. App Component (Root)

- The central component managing the application's state.
- Contains the **Local Clock** and **Clock List** components.

2. Local Clock Component

- Represents the user's main clock (cannot be deleted, only edited).
- **Child Components:**
  - **Clock Display & Labels** – Shows the clock title, time, and timezone.
  - **Clock Actions** – Allows editing the primary clock.
  - **Create Clock Modal** (opens when editing)
    - **Clock Form** – The form where users modify the primary clock details.

---

3. Clock List Component

- Displays all additional clocks created by the user.
- **Child Components:**
  - **Clock List Items** – Each clock instance in the list.

Clock List Item Components

- **Clock Display & Labels** – Shows time, timezone, and clock title.
- **Toggle Events** – Expands/collapses the event list.
  - **Event Item** – Displays individual event details.
    - **Modal** – Opens when an event is edited.
      - **Event Form** – Used for adding/editing an event.
- **Create Event** – Button to add a new event.
  - **Modal** – Opens when creating an event.
    - **Create Event Form** – Input fields for event creation.
- **Clock Actions** – Edit/Delete functionality.
  - **Modal** – Opens when editing a clock.
    - **Clock Form** – Form for modifying clock details.

---

4. Reusable Components

These components are used multiple times throughout the application:

- **Clock Display & Labels**
- **Clock Actions**
- **Modal**
- **Clock Form**

---

5. UI Components

Fundamental UI elements used across different components:

- **Text** – For displaying text content.

- **Input** – For user input fields.
- **Select** – Dropdown selection (e.g., choosing a timezone).
- **Button** – Clickable buttons for actions.
- **List** – Container for list-based elements.
- **List Item** – Individual items within a list.
- **Card** – UI container for displaying clocks.

---

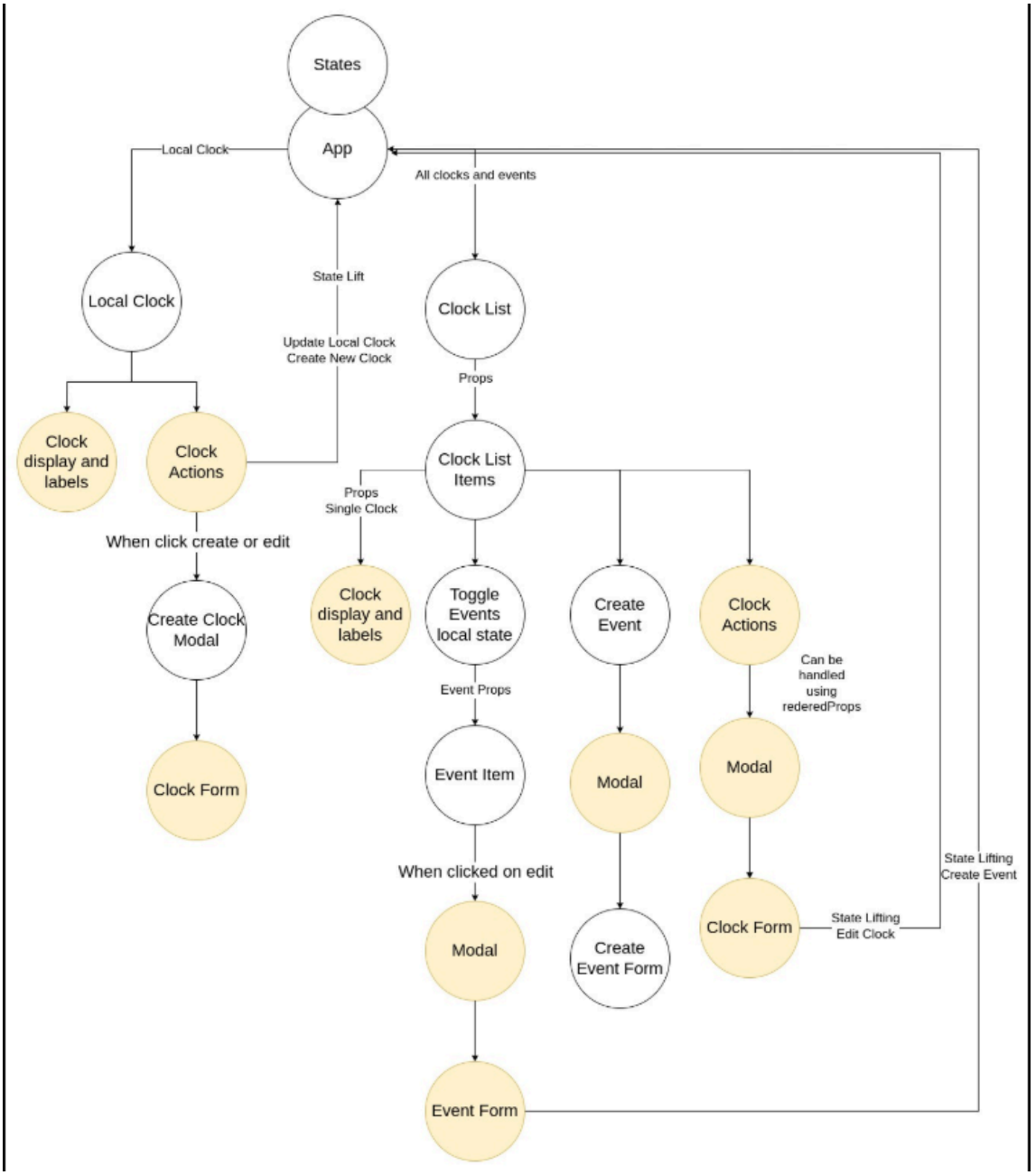
6. Custom Hooks

- **useForm** – Manages form state and validation.
- **useClock** – Handles individual clock logic.
- **useClocks** – Manages multiple clocks in the app.
- **useEvents** – Handles event-related logic.

---

2. Data Flow Diagram

The **Data Flow Diagram** shows how data moves between components. In React, data typically flows **unidirectionally** (from parent to child via props) or is managed globally using **context** or **state management libraries**.



Data Flow & State Management Description

1. States (Global State)

- The **App** component holds the main state.
- Manages:
  - **Local Clock State** (for the primary clock).
  - **All Clocks & Events State** (list of user-created clocks and their events).
- Uses **State Lifting** to manage updates.

2. App Component (Root)

- Acts as the **central controller** for clocks and events.
- **Passes Props** to child components:
  - Local Clock (handles the main clock).
  - Clock List (manages additional clocks).
- Handles **state updates** when:
  - The local clock is updated.
  - A new clock is created.
  - Events are added or edited.

---

3. Local Clock Component

- Displays and manages the **main user clock**.
- **Child Components:**
  - **Clock Display & Labels**: Shows time, title, and timezone.
  - **Clock Actions**: Allows editing the local clock.
  - **Create Clock Modal** (opens on edit action).
    - **Clock Form**: Updates the local clock.

---

4. Clock List Component

- Manages and displays multiple user-created clocks.
- **Passes Single Clock Data via Props** to:
  - **Clock List Items** (each clock instance).
- **Handles Events:**
  - **Props for Events** are passed down to event-related components.

Clock List Item Components

- **Clock Display & Labels** – Shows individual clock details.
- **Toggle Events (Local State)** – Expands/collapses event list.
- **Event Item** – Displays each event, receives **event props**.
  - Clicking "Edit" opens a **Modal** with **Event Form**.
- **Create Event** – Allows adding a new event.
  - Uses **State Lifting** to pass event data back to `App`.
  - A **Modal** with **Create Event Form** opens on action.
- **Clock Actions** – Allows editing/deleting a clock.
  - Uses **State Lifting** to update the clock in `App`.
  - A **Modal** with **Clock Form** is triggered on edit.

---

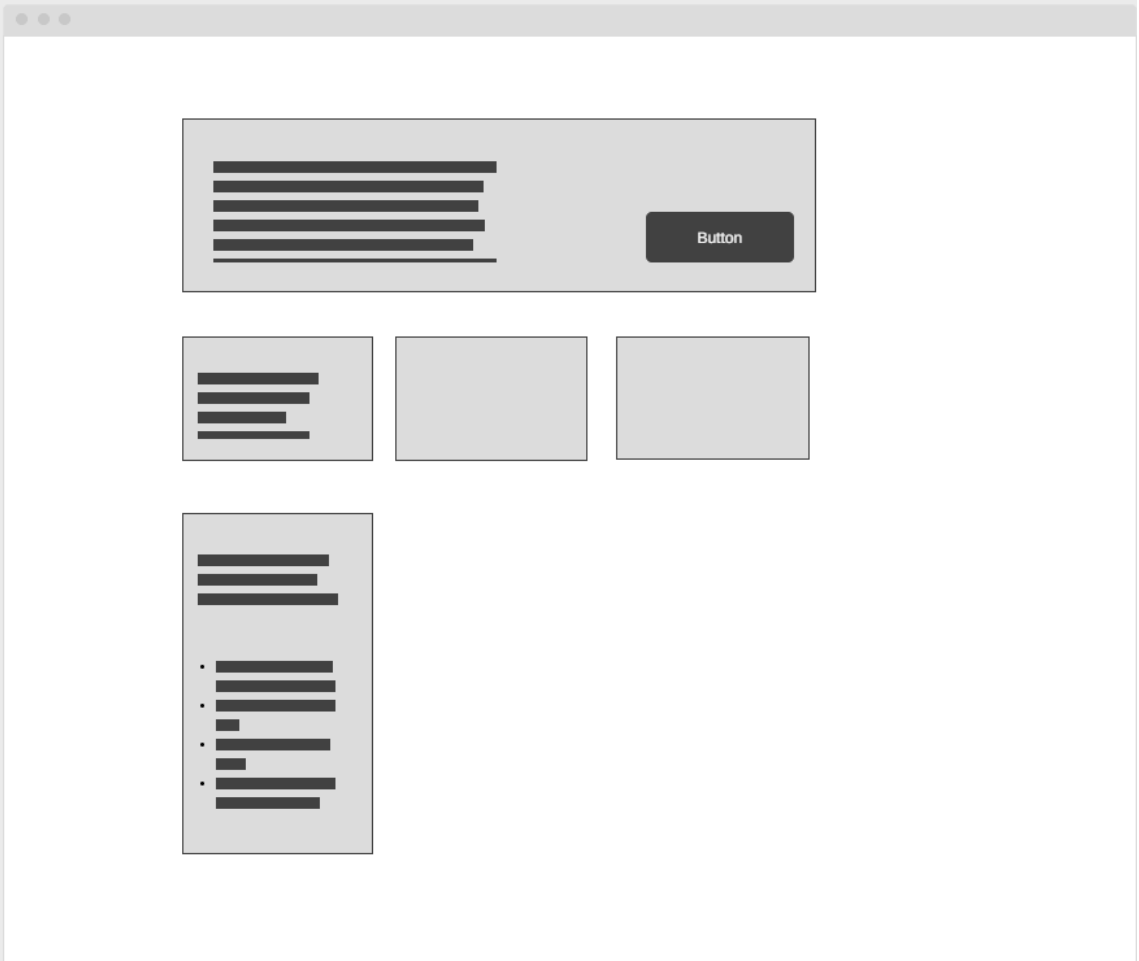
5. State Management & Prop Handling

- **State Lifting:**
  - `App` handles global clock and event state.
  - Updates flow **up** from `Clock Actions` & `Create Event`.
- **Props:**
  - Used to pass **single clock details** to `Clock List Items`.
  - Used to pass **event data** to `Toggle Events` and `Event Item`.
- **Re-render Optimization:**
  - Suggestion: Use **memoization (React.memo)** or **callback props** to reduce unnecessary re-renders.

---

3. UI View

Creating a UI view for the TrackZone application involves designing a user interface that incorporates all the components mentioned in the diagram. Below is a conceptual UI layout based on the provided components:



Main Content Area:

- **Clock Display**:
  - Large clock face showing the current time for the selected clock.
  - Labels for the clock (e.g., "Work Clock").
  - Buttons for "Edit Clock" and "Delete Clock."
- **Event List**:
  - A list of events associated with the selected clock.
  - Each event item shows:
    - Event name (e.g., "Meeting with Team").
    - Event time (e.g., "2:00 PM - 3:00 PM").
    - Buttons for "Edit Event" and "Delete Event."

- **Create/Edit Clock Modal**:
  - Title: "Create Clock" or "Edit Clock."
  - Form Fields:

        - Clock Name (Text Input).
        - Time Zone (Select Dropdown).
        - Labels (Text Input).
    - Buttons: "Save" and "Cancel."
- **Create/Edit Event Modal**:
    - Title: "Create Event" or "Edit Event."
    - Form Fields:
        - Event Name (Text Input).
        - Start Time (Time Picker).
        - End Time (Time Picker).
        - Associated Clock (Select Dropdown).
    - Buttons: "Save" and "Cancel."