`Documentation.md`

Modular and Scalable To-Do List App

Structure:

- **App.tsx:** The entry point, having the primary component `AppMain`.
- **src/:** Contains the app's core logic (`AppMain.tsx`) and other components.

AppMain (src/AppMain.tsx):

- Manages app state with React's `useState`:

  - Tasks (list of tasks)
  - Task input (current task being entered)
  - Filters ("All", "Done", "Not Done")
  - Dark/light mode toggle

- Handles tasks:

  - `addTask` (adds a new task)
  - `deleteTask` (removes a task)
  - `toggleTaskCompletion` (marks tasks as done/undone)
  - `startEditing` (initiates task editing)
  - `saveTask` (saves edits)
  - `cancelEditing` (discards edits)

- Renders UI elements:

  - Header
  - Task input field
  - Action buttons
  - Task filters
  - `TaskList` component

Components (src/components/):

- **TaskList (TaskList.tsx):**
  - Renders a `FlatList` of tasks, mapping each to a `TaskItem`.
  - Manages task actions (edit, delete, toggle completion).
- **TaskItem (TaskItem.tsx):**
  - Represents a single task.
  - Enables task editing, completion toggling, and deletion.
- **Filters (Filters.tsx):**
  - Provides filter options ("All", "Done", "Not Done") for task filtering.

Testing (tests/):

- Unit tests for components (e.g., `AppMain.test.tsx`, `TaskItem.test.tsx`) using `@testing-library/react-native`.
- Covers component rendering and user interactions (adding tasks, toggling completion).

Key Decisions:

1. **State Management:** Used `useState` for component-level state, for reactivity and efficient changes.
2. **Modularity:** Components (AppMain, TaskList, TaskItem, Filters) are modular for reusability and maintainability, alignings with React's component-based architecture.
3. **UI/UX Design:** Implements a straightforward and user-friendly interface with dark mode support, enhancing user experience and accessibility.
4. **Animation:** Employs `LayoutAnimation` for smooth transitions during task operations (adding, deleting, editing), creating a polished user interface.
5. **Testing:** Includes unit tests to guarantee component functionality and user interactions work flawlessly, upholding code quality and reliability.

Overall, the application prioritizes:

- **Modular Design:** Components are separate and reusable, facilitating maintenance and future expansion.
- **Efficient State Management:** `useState` simplifies state handling within components for optimal performance.
- **User-Centric Design:** The interface is clear and intuitive, with dark mode for better accessibility.
- **Interactive Experience:** `LayoutAnimation` provides smooth animations, enhancing user engagement.
- **Reliable Testing:** Unit tests ensure the application's robustness and responsiveness.

This structure offers a well-organized and scalable foundation for a to-do list app, prioritizing clean code, user experience, and maintainability.