**Low-Level Design (LLD) Document**

**1. Introduction**

**1.1 Purpose**

The purpose of this Low-Level Design document is to provide a detailed breakdown of the components, their functionalities, and how they interact within the To-Do List Application.

**1.2 Scope**

This document outlines the classes, functions, and data structures used in the application, providing the necessary details for implementation.

**2. Components and Data Structures**

**2.1 Data Structure for Tasks**

The tasks will be stored in an array of objects in local storage. Each task object has the following structure:

{

"name": "Task Name",

"completed": false

}

**2.2 Key Components**

**2.2.1 User Interface (HTML)**

The main components of the UI are outlined below:

* **Header**
  + Contains the title of the application.
* **Input Area**
  + **Input Field**: For entering new tasks.
    - id: new-task
    - type: text
    - placeholder: Add new task
  + **Button**: For adding a new task.
    - id: add-task-btn
    - type: button
    - text: +
* **Task List**
  + An unordered list to display tasks.
    - id: task-list
* **Footer**
  + Contains copyright and additional information.

**2.2.2 JavaScript Functions**

The JavaScript functions manage the application's functionality. Below are the key functions and their responsibilities:

1. **addTask()**
   * **Description**: Adds a new task to the task list.
   * **Input**: Task name from the input field.
   * **Output**: Updates the task list and local storage.
2. **renderTasks()**
   * Description: Renders the list of tasks in the UI.
   * Input: None.
   * Output: Updates the task list display.
3. **renderTasks()**
   * Description: Renders the list of tasks in the UI.
   * Input: None.
   * Output: Updates the task list display.
4. **deleteTask(index)**
   * Description: Deletes a task from the list.
   * Input: Index of the task to be deleted.
   * Output: Updates the task list and local storage.
5. **deleteTask(index)**
   * Description: Deletes a task from the list.
   * Input: Index of the task to be deleted.
   * Output: Updates the task list and local storage.
6. **deleteTask(index)**
   * Description: Deletes a task from the list.
   * Input: Index of the task to be deleted.
   * Output: Updates the task list and local storage.
7. deleteTask(index)
   * Description: Deletes a task from the list.
   * Input: Index of the task to be deleted.
   * Output: Updates the task list and local storage.

**2.3 Event Listeners**

* **Add Task Button**: When clicked, it triggers the addTask() function.
* **Window Load**: When the window loads, it calls loadTasks() to populate the task list.

**3. Sequence Diagram**

**Sequence Diagram for Adding a Task**

User UI JavaScript Local Storage

| | | |

|-- Enter Task -->| | |

| |-- Validate Input -->| |

| |-- Add Task -->|-- Save to Local Storage -->|

| | |<-- Confirmation --|

|<-- Update UI --| | |

**4. Conclusion**

This Low-Level Design document provides a comprehensive breakdown of the components, their functionalities, and interactions within the To-Do List Application. It serves as a guide for developers to implement the system in a structured manner.