**Project’s Design**

1. **Problem Identification**

The problem that has been identified is the need to develop a task and reminder management system that allows users to efficiently add, organize, and manage their to-dos and reminders. This problem stems from the lack of an effective tool for organizing tasks and reminders that are easy to use and provide users with the ability to prioritize their activities.

1. **Gathering Necessary Information**

Understanding user needs and expectations is critical to the effective design and development of the task and reminder management system. Knowing the different types of tasks and reminders that users will manage is essential to designing an appropriate data structure.

Understanding how users will interact with the system in their daily workflow is essential to designing an intuitive user interface. Gathering information on how users will assign priorities and categorize tasks is crucial to implementing priority management. The choice of technical aspects, such as programming language, databases, and development tools, should be based on the needs of the project. Data security and privacy are key considerations, and it is important to know the expectations of users in this regard.

If the system is subject to specific regulations, gathering information on the corresponding compliance requirements is necessary. Planning for ongoing maintenance and support is essential to ensure optimal system performance over time. User interface design, usability, and navigation should be informed by user feedback to ensure a positive experience. Ongoing feedback and user testing are necessary to adjust and improve the system.

1. **Search For Creative Solutions**

**To solve this problem, we can use structures such as:**

**ArrayList:**  
  
Utility: An ArrayList is a dynamic data structure that provides a resizable list. It can be used to store tasks and reminders in a flexible way.

Advantages: It allows quick access to items via indexes and is easy to use and manage. It is useful when the size of the list is dynamic and may change over time.

**Hash table:**  
  
Utility: A hash table provides efficient lookup and fast insertion and deletion times. It can be used to store tasks and reminders, using a unique identifier as a key.

Advantages: Provides quick access to data, especially when you need to search for tasks based on a unique identifier, such as a task code.

**Priority Queue (Heap):**  
  
Benefit: A heap (min-heap) can be used to organize tasks according to their level of importance or deadline. Priority tasks are handled first.

Advantages: Ensures that the most important tasks are handled first. Efficient for inserting new tasks and extracting the highest priority task.

**Linked List (FIFO):**  
  
Utility: A linked list can be useful for managing non-priority tasks on a first-come, first-served basis.

Advantages: Allows easy insertion and deletion of items in the middle of the list, while maintaining a specific order for non-priority tasks.

**Stack (LIFO):**

Utility: A stack can be used to implement the "undo actions" functionality. Each action performed by the user can be stacked for later reversal.

Advantages: Allows to undo the last action in an efficient way, following the Last-In-First-Out (LIFO) principle.

**Search Tree (BST):**

Utility: A binary search tree can be useful for organizing tasks by deadline or priority, facilitating efficient search and retrieval.

Benefit: Ensures quick access to tasks based on a specific criterion, such as deadline.

**Graph:**

Utility: A graph can be useful if tasks have complex relationships with each other, such as dependencies. Tasks could be nodes and dependencies could be represented by arcs in the network.

Advantages: Allows modeling and managing complex relationships between tasks, such as dependencies and workflows.

1. **Transition From Idea Formulation to Preliminary Designs**

1st Alternative: Hash Table

2nd Alternative: Stack (LIFO):

3rd Alternative: Queue:

4th Alternative: Priority queue:

1. **Evaluation and Selection of the Best Solution**
2. **Preparation of Reports and Specifications**
3. **Design Implementation**