**Objectives and purpose.**

**Application Objective**

Task management system aims to provide users with a simple, efficient, and user-friendly way to store, retrieve, update, and delete tasks.

Following is the primary objective of the application:

* **Task Management: To** allow users to add, update, delete, and retrieve tasks.
* **Categorization**: To enable users to categorise tasks according to its priority.
* **Goal setting**: To enable users to set goals and track them.

### **Use Case: Managing Tasks**

**Primary Actors**: User/Administrators/Team member

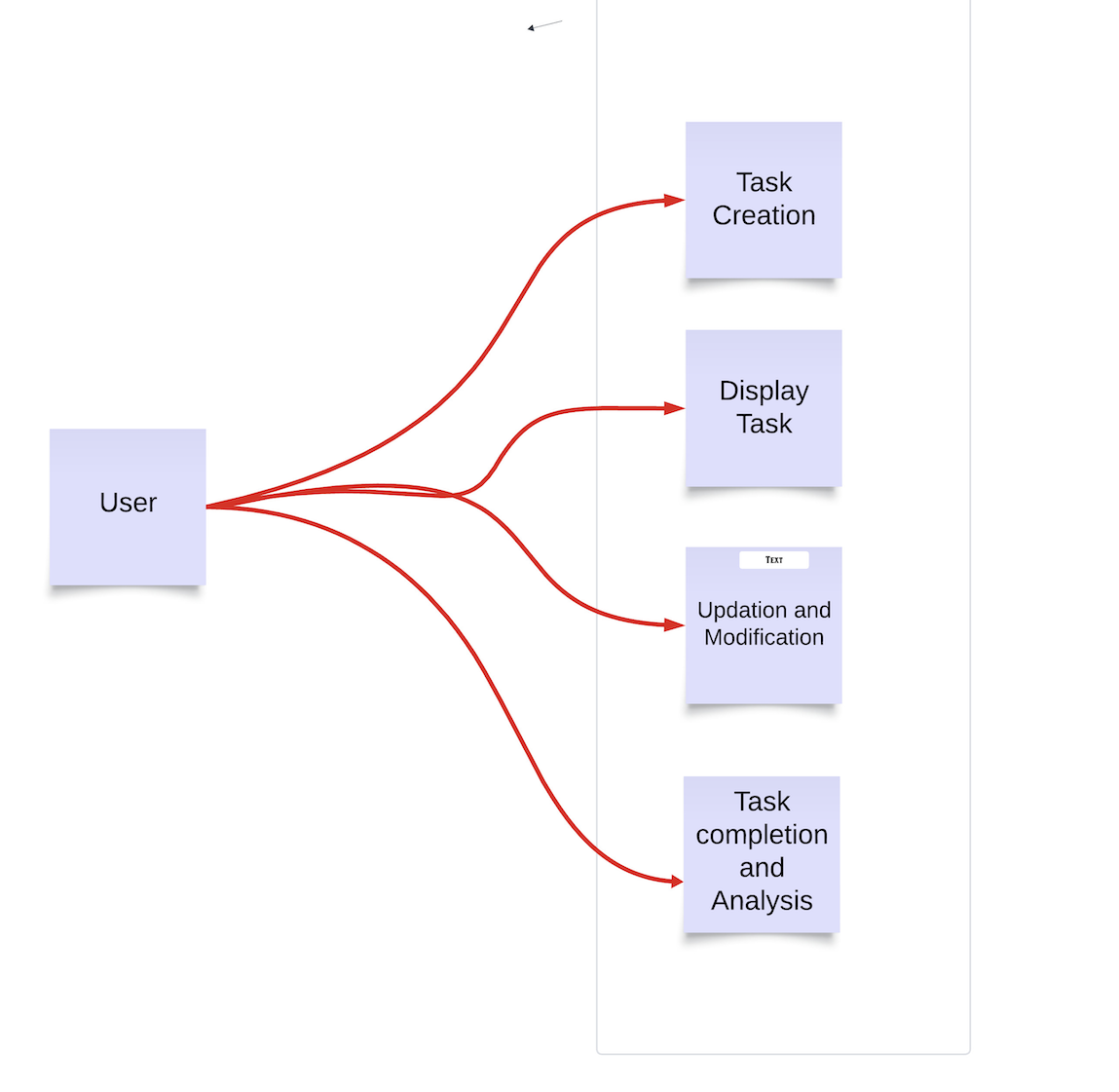
**Pre-condition: Users** must have an account in the task management application. User must login into the system.

**Main Flow**

* **Task creation:** Users create a new task by entering the details such as task title,description,due date at primary level.
* **Display tasks:** Display created task. Filter and sort task by name, due date, category, tags etc.
* **Task Updation:** Change the name of the task,category,due date etc. Update progress of the task.
* **Delete Task:** The user can delete the task if it is no longer needed.
* **Task completion:** Once the task is marked as completed it moves to the completed section.

**Post Condition:**

Tasks are updated, completed or deleted as needed.Users have a clear understanding of the responsibility and deadline, and user goals are met.



**Data structure and Algorithm**

1. **Data structure**

**Task objective diagram**

|  |  |
| --- | --- |
| **Field** | **Description** |
| Task ID  Title  Description  Priority  Deadline  Status | Unique identifier (Integer)  Name (string)  Task Information(string)  Task priority(enum)  Deadline(datetime)  Task status -pending/completed(string) |

**Algorithm**

1.Insertion

2.Deletion/modification (Remove task)

3. Sorting (Sorting based on the priority, deadline)

4.Searching (Search task by ID)

**Pseudocode and flowchart for Insertion Algorithm**

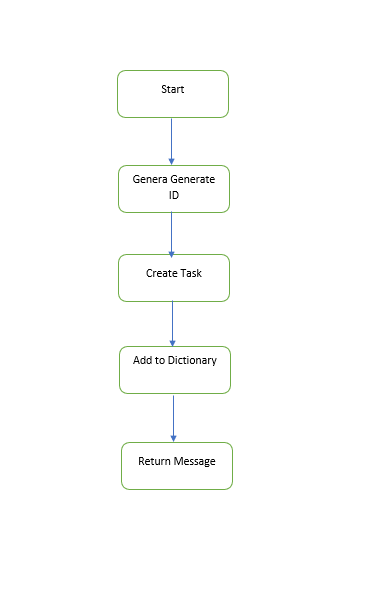
FUNCTION addTask(title, description, priority,deadline)

taskID - unique ID

newTask - Create Task object with input values

taskStorage[taskID] - newTask

RETURN “Task added successfully”



**Pseudocode and flowchart for Deletion Algorithm**

FUNCTION deleteTask(taskID):

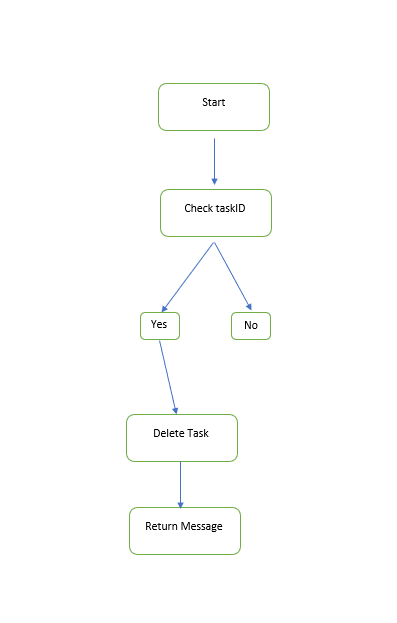
IF taskID IN task storage:

REMOVE taskStorage[taskID]

RETURN “Task deleted successfully”

ELSE:

RETURN”Task not found”



**Pseudocode and flowchart for Sorting Algorithm**

FUNCTION SortTasks(by):

taskList <= Convert taskStorage

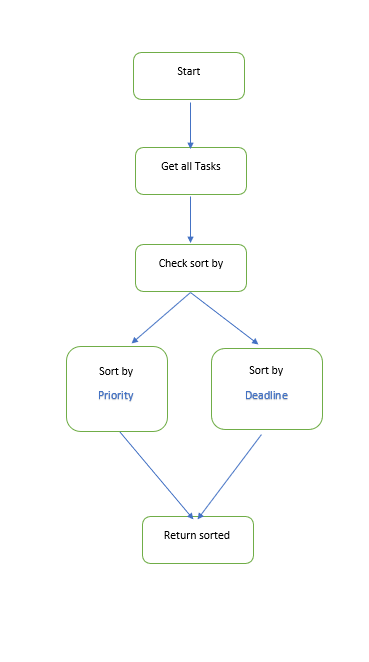
IF by = “priority”:

SORT taskList BY task.priority DEC

ELSE IF by="deadline”:

SORT taskList BY task.deadline ASC

RETURN taskList

****

**Pseudocode and flowchart for Search Algorithm**

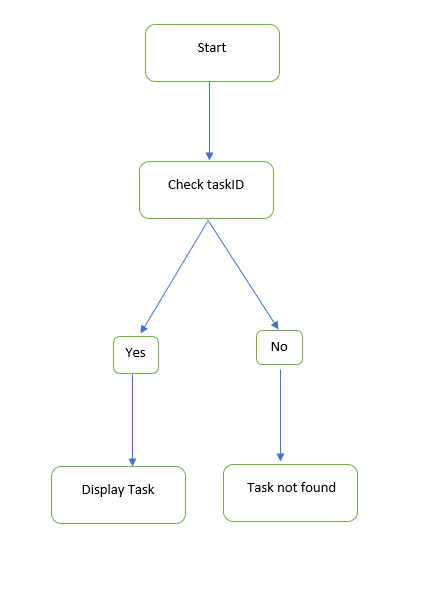
FUNCTION searchTask(taskID):

IF taskID IN taskStorage[taskID]

RETURN taskStorage[taskID]

ELSE:

RETURN”Task not found”



**Testing**

**Test objective**

1. Verify that all functions of task management system working as expected
2. Test the correctness of core functionalities such as insertion, deletion, sorting and searching.
3. Identify the potential bugs and errors

**Test scope**

* Task creation
* Task deletion
* Sorting task
* Searching task

**Test Cases-Insertion**

**Test case 1**: Add new task

**Test Input:** Input valid task information

Click “Add task” or run insertion function

**Expected output:** Task added successfully with unique taskID and confirmation message displayed.

**Test case 2:** Add task with no title

**Test Input:** leave the task title column empty.

Click “add task” or run insertion function

**Expected output:** Error message “Title required” displayed

**Test case 3:** Add task with past deadline

**Test input:** Add task with past deadline

Click “add task” or run insertion function

**Expected output:** Error message "Deadline cannot be in the past." displayed

**Test Cases-Deletion**

**Test case 1**: Deleting existing task

**Test Input:** Input valid taskID

Click “Delete” or run delete function

**Expected output:** Amessage” Task deleted Successfully” displayed.

**Test Cases-Sorting**

**Test case 1:** “Sort task by priority”

**Test Input:** Add multiple tasks with different priorities.

Select “sort by priority” ‘

**Expect output:** Task are displayed descending order of priority

**Test case 2:** “Sort task by deadline”

**Test Input:** Add multiple tasks with different deadlines.

Select “sort by deadline”

**Expect output:** Tasks are displayed ascending order of deadline.

**Test Cases-Searching**

**Test case 1:** “searching an existing task”

**Test Input:** Input valid taskID

Click “search task” or run search function

**Expect output:** Task display with unique taskID

**Test case 2:** “searching a non- existing task”

**Test Input:** Input invalid taskID

Click “search task” or run search function.

**Expect output:** Error message "Task not found" displayed

**Reference**

* Ashishps1(2023). awesome-low-level-design/problems/task-management-system.md at main · ashishps1/awesome-low-level-design. [online] GitHub. Available at:<https://github.com/ashishps1/awesome-low-level-design/blob/main/problems/task-management-system.md> [Accessed 16 Dec. 2024].
* BBC Bitesize. (2020). Designing an algorithm - Revision 2 - KS3 Computer Science - BBC Bitesize.:<https://www.bbc.co.uk/bitesize/guides/z3bq7ty/revision/2>.
* data, W. (2023). What data structure would be the most efficient for a task scheduler? [online] Game Development Stack Exchange. Available at:<https://gamedev.stackexchange.com/questions/208359/what-data-structure-would-be-the-most-efficient-for-a-task-scheduler> [Accessed 16 Dec. 2024].
* GrandQ (2023). Which data structure is suitable for task-based system. Stack Overflow.:<https://stackoverflow.com/questions/77211239/which-data-structure-is-suitable-for-task-based-system>.
* NocoBase (2024). [Tutorial] Chapter 2: Designing a Task Management System. [online] Medium. Available at:<https://medium.com/@nocobase/tutorial-chapter-2-designing-a-task-management-system-0e08896df49f> [Accessed 16 Dec. 2024].