Data Structures and Algorithms

SuperSet ID:6412063

Exercise 5: Task ManagementSystem

Code:

class Task {

    int taskId;

    String taskName;

    String status;

    public Task(int taskId, String taskName, String status) {

        this.taskId = taskId;

        this.taskName = taskName;

        this.status = status;

    }

    public String toString() {

        return "Task ID: " + taskId + ", Name: " + taskName + ", Status: " + status;

    }

}

class Node {

    Task task;

    Node next;

    public Node(Task task) {

        this.task = task;

        this.next = null;

    }

}

public class TaskManagementSystem {

    private Node head;

    public void addTask(Task task) {

        Node newNode = new Node(task);

        if (head == null) {

            head = newNode;

        } else {

            Node curr = head;

            while (curr.next != null) {

                curr = curr.next;

            }

            curr.next = newNode;

        }

        System.out.println("Task added.");

    }

    public Task searchTask(int taskId) {

        Node curr = head;

        while (curr != null) {

            if (curr.task.taskId == taskId) {

                return curr.task;

            }

            curr = curr.next;

        }

        return null;

    }

    public void displayTasks() {

        if (head == null) {

            System.out.println("No tasks found.");

        } else {

            Node curr = head;

            while (curr != null) {

                System.out.println(curr.task);

                curr = curr.next;

            }

        }

    }

    public void deleteTask(int taskId) {

        if (head == null) {

            System.out.println("No tasks to delete.");

            return;

        }

        if (head.task.taskId == taskId) {

            head = head.next;

            System.out.println("Task deleted.");

            return;

        }

        Node prev = head;

        Node curr = head.next;

        while (curr != null) {

            if (curr.task.taskId == taskId) {

                prev.next = curr.next;

                System.out.println("Task deleted.");

                return;

            }

            prev = curr;

            curr = curr.next;

        }

        System.out.println("Task not found.");

    }

    public static void main(String[] args) {

        TaskManagementSystem tms = new TaskManagementSystem();

        tms.addTask(new Task(101, "Design Module", "Pending"));

        tms.addTask(new Task(102, "Code Review", "Completed"));

        tms.addTask(new Task(103, "Test Cases", "In Progress"));

        System.out.println("\nAll Tasks:");

        tms.displayTasks();

        System.out.println("\nSearching for Task ID 102:");

        Task found = tms.searchTask(102);

        System.out.println(found != null ? found : "Not found");

        System.out.println("\nDeleting Task ID 101:");

        tms.deleteTask(101);

        System.out.println("\nTasks after deletion:");

        tms.displayTasks();

    }

}

Output:

A white background with text

Description automatically generated