**Task Management System Documentation**

**1. Overview**

The Task Management System is a simple Java application that allows users to manage tasks using a linked list data structure. The system provides functionality to add tasks, view all tasks, and delete tasks based on their unique task ID.

**2. Classes**

**2.1 Task**

The Task class represents an individual task in the system.

**Attributes:**

* taskId (int): Unique identifier for the task
* taskName (String): Name or description of the task
* status (String): Current status of the task
* next (Task): Reference to the next task in the linked list

**Constructor:**

* Task(int taskId, String taskName, String status): Initializes a new Task object

**2.2 TaskList**

The TaskList class manages the collection of tasks using a linked list structure.

**Attributes:**

* head (Task): Reference to the first task in the list
* scanner (Scanner): Used for user input

**Methods:**

* isTaskIdExists(int taskId): Checks if a task with the given ID already exists
* addTask(): Adds a new task to the end of the list
* traverseTasks(): Displays all tasks in the list
* deleteTask(): Deletes a task based on its ID

**2.3 TaskManagementSystem**

The TaskManagementSystem class contains the main method and serves as the entry point for the application.

**Methods:**

* main(String[] args): Runs the main program loop, presenting a menu to the user and calling appropriate TaskList methods based on user input

**3. Functionality**

1. **Add Task**: Users can add a new task by providing a unique task ID, task name, and status. The system ensures that task IDs are unique.
2. **View All Tasks**: Users can view all tasks currently in the system, displaying the task ID, name, and status for each task.
3. **Delete Task**: Users can delete a task by providing its task ID. The system removes the task from the linked list.
4. **Exit**: Users can choose to exit the program.

**4. Data Structure**

The system uses a singly linked list to store and manage tasks. Each Task object contains a reference to the next task in the list, allowing for efficient traversal and manipulation of the list.

**5. User Interface**

The system provides a simple command-line interface with a menu-driven approach. Users interact with the system by entering numeric choices and providing task details when prompted.

7**. Implementation**

**Link:** [**Click here for the Code**](https://github.com/Akashmondal55/Akash_5016855/tree/main/Week-1/DSA/Exercise-5)**.**

**8. Conclusion**

The Task Management System provides a simple yet effective solution for managing tasks using a linked list data structure in Java. This system demonstrates the practical application of fundamental programming concepts such as:

1. Object-Oriented Programming (OOP): The system utilizes classes and objects to represent tasks and manage their collection.
2. Data Structures: The implementation of a singly linked list showcases how basic data structures can be used to organize and manipulate data efficiently.
3. User Input Handling: The system interacts with users through a command-line interface, demonstrating input/output operations in Java.
4. Control Structures: The use of loops and conditional statements to control program flow and implement the menu-driven interface.
5. Modular Design: The separation of concerns between the Task, TaskList, and TaskManagementSystem classes promotes code organization and maintainability.

While this system serves as a good starting point for task management, there are several areas where it could be enhanced for real-world use:

* Persistence: Implementing data storage to save tasks between program executions.
* Error Handling: Adding more robust error checking and exception handling.
* Task Editing: Including functionality to modify existing tasks.
* Sorting and Filtering: Adding capabilities to sort tasks or filter them based on various criteria.
* User Authentication: Implementing a multi-user system with login functionality.

Overall, this Task Management System provides a solid foundation for understanding linked list implementations and basic CRUD (Create, Read, Update, Delete) operations in Java. It can serve as a starting point for more complex project management or to-do list applications.