

Task Management System

Submitted By

Student Name	Student ID
Tahmid Mustofa	241-15-400
Sadman Yasar Mahib	241-15-678
Arko biswas	241-15-376
Moyen Ahmed	241-15-675
Rakib Hasan	241-15-134

MINI LAB PROJECT REPORT

This Report Presented in Partial Fulfillment of the course **CSE124:Data
Sructure in Computer Science and Engineering Department**



Daffodil
International
University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

December 10, 2024

DECLARATION

We hereby declare that this lab project has been done by us under the supervision of **Amatul Bushra Akhi, Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere as lab projects.

Submitted To:

Amatul Bushra Akhi

Assistant Professor

Department of Computer Science and
Engineering
Daffodil International University

Submitted by

Tahmid Mostafa Student ID: 241-15-400 Dept. of CSE, DIU	
Sadman Yasar Mahib Student ID: 241-15-678 Dept. of CSE, DIU	Arko biswas Student ID: 241-15-376 Dept. of CSE, DIU
Moyen Ahmed Student ID: 241-15-675 Dept. of CSE, DIU	Rakib Hasan Student ID: 241-15-134 Dept. of CSE, DIU

Introduction

Introduces the problem and provides the context for developing the Task Management System. It highlights the need for an efficient task-tracking solution in everyday life or professional settings.

- **Introduction**

Managing tasks manually often leads to inefficiencies, missed deadlines, and difficulty in tracking progress. A systematic approach is needed to add, manage, and prioritize tasks efficiently. The project aims to create a console-based Task Management System that helps users organize their work effectively.

- **Motivation**

Computational Motivation: Developing a system with features like task sorting, undo functionality, and statistics aligns with real-world needs in task management and software development.

Personal Benefit: The project offers practical experience in implementing linked lists, stacks, file handling, and dynamic memory allocation in C programming.

- **Objectives**

- Create a console-based application for task management.
- Implement features such as add, edit, delete, sort, and search tasks.
- Provide file-based persistence for saving and loading tasks.
- Include statistical reporting and advanced options like undo functionality.

- **Feasibility Study**

Popular apps like Microsoft To-Do and Trello focus on GUI-based, online solutions, whereas this project offers an offline, console-based alternative. This system simplifies task management using basic data structures and serves as a learning platform for beginner programmers.

- **Gap Analysis**
 - **Lightweight and Offline:** Many tools need the internet or are too complex. Our system works offline and is simple.
 - **Learning Example:** Current tools don't show how to build something like this in C. Our project teaches useful concepts like memory management and linked lists.
 - **Simple for Small-Scale Use:** Most tools are made for big teams or projects but ours is perfect for personal use and small tasks.
- **Project Outcome**
 - A functional task manager with core and advanced features.
 - Persistent task storage through file handling.
 - Statistical insights for tracking task completion.
 - An extendable codebase for educational purposes and future enhancements.

Proposed Methodology/Architecture

This section will explain how the system is designed to solve the task management problem and the steps taken to implement it.

- **Requirement Analysis & Design Specification**

The task management system needs to be capable of storing, managing, and tracking tasks efficiently. Users should be able to perform basic operations like adding, editing, deleting, and marking tasks as completed. Additionally, the system must support task sorting, searching, and undoing actions. Data persistence is crucial, meaning the tasks must be saved and loaded from a file.

- **Overview:** The system must manage tasks effectively by allowing users to add, edit, delete, and track tasks. It needs to be user-friendly and allow saving/loading of tasks, sorting, and searching.
- **Proposed Methodology/ System Design**

- **UI Design :** The program was designed with a text-based user interface that displays options like adding tasks, deleting tasks, etc. It will be simple to navigate and use, with each action triggered by entering a number corresponding to the option.
- **Overall Project Plan**

The project will be developed step-by-step: starting with the creation of the task list, followed by adding functionalities like sorting and searching, and finally, implementing advanced features such as undo and statistics.

Implementation and Results

Describes how the task management system was built based on the design specifications. The code implements core functionalities such as adding, deleting, editing, completing tasks, and saving/loading tasks from a file.

- **Implementation**

The system uses a linked list to store tasks, and the user interacts with the system through a text-based menu. Each operation is implemented as a separate function, and the system supports basic error handling to ensure smooth operation.

- **Performance Analysis**

The performance of the task management system is evaluated based on its ability to handle operations like adding, deleting, and viewing tasks, as well as loading and saving data from/to a file. The system is expected to perform well for small to medium-sized task lists. The performance may degrade slightly with a large number of tasks, as the linked list operations such as searching or sorting require linear time.

- **Results and Discussion**

The task management system was successfully implemented, and the results show that it works as expected. Users can add, delete, edit, and complete tasks, and the data is saved correctly to

a file. The system also handles basic task sorting and searching. While the system performs well for typical use cases, there may be performance issues with a large number of tasks due to the linear complexity of certain operations. Nonetheless, the system meets the core requirements and provides a simple, efficient way to manage tasks.

Engineering Standards and Mapping

Discusses how the task management system might influence society, the environment, and sustainability.

- **Impact on Society, Environment and Sustainability**

Impact on Life: The system can improve personal productivity, making it easier to organize and track tasks.

Impact on Society & Environment: Can contribute to efficient time management and better work-life balance, benefiting users in the long term.

Ethical Aspects: The system is designed to respect user data and provide transparency in task management. No data was misused.

Sustainability Plan: The system uses minimal resources and can be easily maintained and updated. It requires no special hardware, making it sustainable in terms of low resource consumption.

- **Project Management and Team Work**

Budget: The budget for this project includes development costs such as software tools, time for programming, and testing. An alternate budget could focus on reducing costs by using free open-source software tools and libraries.

Revenue Model: If the system were to be commercialized, revenue could come from offering premium features such as advanced task tracking, integration with other tools, or cloud storage options.

- **Complex Engineering Problem**

Mapping of Program Outcome

Table : Justification of Program Outcomes

PO's	Justification
PO1	The system demonstrates strong problem-solving skills in creating an efficient task management tool.
PO2	The project demonstrates knowledge in software design and development, applying engineering principles in the coding process.
PO3	The system ensures usability and reliability, showcasing practical applications of engineering principles in a real-world context.

- Complex Problem Solving**

Table : Mapping with complex problem solving.

EP1 Depth of Knowledge	EP2 Range of Conflicting Require- ments	EP3 Depth of Analysis	EP4 Familiarity of Issues	EP5 Extent of Applicable Codes	EP6 Extent of Stake- holder Involve- ment	EP7 Inter- dependence
✓	✓	✓	✓	✓	✓	

- Engineering Activities**

Table : Mapping with complex engineering activities.

EA1 Range of re- sources	EA2 Level of Interac- tion	EA3 Innovation	EA4 Consequences for society and envi- ronment	EA5 Familiarity
✓	✓	✓	✓	✓

Conclusion

Provides a brief overview of the findings, limitations, and future directions for the task management system project.

- **Summary**

The task management system successfully implements key features like adding, editing, deleting, sorting, and searching tasks, with a user-friendly console interface and file-based persistence.

- **Limitation**

The current version of the system is limited by its console-based interface, lack of advanced security features, and absence of a cloud-based synchronization option.

- **Future Work**

Future improvements could include implementing a graphical user interface (GUI), adding mobile app compatibility, and introducing cloud synchronization for better user experience and data management.