



**THE IMPLEMENTATION OF TO-DO-LIST SYSTEM TO SUPPORT TIME  
MANAGEMENT**

In Partial Fulfillment of the  
Requirements for the Subject  
Computer Programming I  
and Introduction to Computing

BALANSI, AMANTI  
CABURNAY, RONNEL  
EBARAT, REINZ B.  
PAJE, JERICHO  
SAMILLANO, JOHN PHILIP

APRIL 2025

## **Introduction**

Time management in the contemporary world of everyday living is significant in personal and professional goal achievement. The importance of time management is apparent to many people since it leads to increased productivity and minimal stress. To-Do List system is one of the simplest instruments, which provides users with an ordered format to arrange the tasks, due dates and daily assignments. Furthermore, the rise in the popularity of the digital productivity tools is because people are searching to find effective methods of keeping their already swamped schedules straight.

Human beings are fond of making their time count effectively so as to be able to strike a balance between the areas of work, studies and leisure. Everyone should ensure that time is well spent whether it is on a weekday or weekends or any other time. A To-Do List is also a balanced and useful tool as it includes the task management system that contributes to the daily organization. With the advancement in technology, the world has turned out to be a fast developing and a very advanced place where people use smart phones and computers in many life activities as it saves their time. People are pragmatic and apply technologies to simplify their lives. The concept behind a Console-Based To-Do List System is good, as it will serve as a personal assistant to those who need to manage their responsibilities.

Most people, especially the students are unable to balance their day-to-day activities and forget about some important deadlines. They do not have a system in place that will help them to organize themselves personally. Lack of a time management system can greatly be a deterrence to productivity. They can be unable to follow all of their responsibilities without a digital platform, using a human memory that is not always reliable or the unsystematic paper notes. The potential users can lose critical tasks, which results in more stress and lost chances of succeeding.

The To-Do List System that has been proposed is aimed at assisting individuals in establishing good time management habits. The system is incredibly significant in the personal development of the contemporary world where the majority of humanity works and studies using computers. The system will be user friendly and present significant information regarding tasks, including description, priorities and status. It will also enable the users to revise their list of tasks in a fast and convenient manner.

## **Statement of the Problem**

Generally, this study focuses on exploring the benefits of a To-Do List System for individuals in their daily life. The system is intended to help users organize their tasks across various activities, prioritize responsibilities, and meet deadlines.

Specifically, this study seeks to answer the following research questions:

1. Can users efficiently organize and manage tasks for daily life activities within the system?
2. Does the system help users to prioritize tasks and improve their time management?
3. Does the To-Do List System help users improve their ability to meet deadlines?

## **Objectives of the Study**

This study aims to explore the advantages of the To-Do List System for daily life. Specifically, this study aims to:

1. Ensure that users can effectively organize and manage their daily tasks within the system.
2. Assess the user-friendliness of the interface and the ease of navigating through the system.
3. Evaluate the impact of the system on users' ability to meet deadlines and reduce stress related to time management.

## **Scope and Limitation**

### **Scope:**

The To-Do List System is about making people manage their daily life activities efficiently with the help of tracking and organizing them. With the system, the user is able to add, see, mark as done and delete the tasks depending on what he or she needs. It offers a text-based and easy to use interface written in Python which encourages productivity, organization and time management amongst users. The system is designed to replace manual paper lists with a digital one in which a person has the opportunity to divide all of his daily tasks and track the progress.

### **Limitation:**

The system has only simple functions of managing tasks like adding, seeing, and marking as done and delete tasks. It lacks such high-tech aspects as automated notifications, in-depth progress analytics, or cross-device data synchronization. Also, the system is not graphically user interface (UI) or mobile version and has to be used manually with a Python environment. The system is developed with personal daily use in mind and might be not able to provide a complicated scheduling of professional team or large project management.

### **Hardware Requirements:**

- Processor: Intel Core i3 or equivalent
- Memory (RAM): At least 2 GB
- Storage: Minimum 100 MB of free disk space
- Monitor: Any display that supports text output (console view)
- Keyboard: For input commands

### **Software Requirements:**

- Operating System: Windows or macOS
- Programming Language: Python
- Code Editor: Visual Studio Code

- Console or Terminal: Command Prompt, PowerShell, or Terminal

## User Functional Requirements:

### 1. Add Task

- Allows users to enter a task description, priority, date, and time.
- Validates task description to ensure it is not empty or purely numeric.
- Validates priority, accepting only "High", "Medium", or "Low" (via a 1-3 menu).
- Validates the date (YYYY-MM-DD) and time (HH:MM AM/PM) for correct formatting.
- Prevents time conflicts by checking if another task is already scheduled at the exact same date and time.
- Presents a confirmation screen after all details are entered, allowing the user to review.
- From the confirmation screen, allows the user to edit any of the new task's details (description, priority, or time) before final confirmation.
- Allows the user to cancel the "add task" operation at any input step.
- Asks the user if they want to add another task after a successful addition.

### 2. View Tasks (with Filter)

- Provides a sub-menu to filter the task list by:
  3. Pending Tasks
  4. Done Tasks
  5. All Tasks
- Sorts the displayed tasks chronologically by their date and time.
- Displays each task's number, status ([ Pending ] or [ Done ]), description, priority, and scheduled date/time.

### 3. Mark Task as Done

- First, displays the complete, sorted list of all tasks.
- Allows the user to select a task by its number to mark as "Done".
- Asks for user confirmation before changing the status.
- Allows the user to cancel the operation (by entering 0).

### 4. Delete Task

- First, displays the complete, sorted list of all tasks.
- Allows the user to select a task by its number to delete.
- Asks for user confirmation before permanently removing the task.
- Allows the user to cancel the operation (by entering 0).

### 5. Edit Task

- First, displays the complete, sorted list of all tasks.
  - Allows the user to select a task by its number to edit.
  - Provides a sub-menu to edit:
    3. Description
    4. Priority
    5. Date/Time
  - Re-uses all validation logic (e.g., for description, priority, and datetime formats).
  - Prevents time conflicts when editing a task's datetime, checking against all other tasks.
  - Allows the user to return to the main menu after finishing edits.

## 6. Search Tasks

- Prompts the user to enter a search term.
  - Performs a case-insensitive search matching the term within each task's description.
  - Displays all matching tasks, sorted chronologically.
  - Reports the total number of tasks found.

## 7. Exit System

- Stops the program by breaking the main application loop.

### **Flowchart (INSERT IMAGE)**

