



**DAFFODIL INTERNATIONAL UNIVERSITY**

**Dhaka, Bangladesh**

**LAB PROJECT REPORT**

.....

**CSE222 || Object Oriented Programming II Lab**  
**Department: Computer Science and Engineering Department**

.....

**‘TO DO LIST APPLICATION’**

**Submitted By**

<b>Student Name</b>	<b>Student ID</b>
Titas Sarker	0242220005101864

# DECLARATION

I hereby declare that this project report has been completed under the supervision of **Ms. Nasima Islam Bithi, Lecturer**, Department of Computer Science and Engineering, Daffodil International University. This report is original and has not been submitted elsewhere for academic purposes.

**Submitted To:**

-----

**Ms. Nasima Islam Bithi**

Lecturer

Department of Computer Science and Engineering

Daffodil International University

**Submitted by:**

-----

Titas Sarker

0242220005101864

Dept. of CSE,DIU

# Table of Contents

- 1. Project Overview**
  - 1.1 Introduction
  - 1.2 Purpose of the Application
- 2. Objectives**
  - 2.1 Key Goals of the Project
- 3. Technologies Used**
  - 3.1 Programming Language
  - 3.2 Libraries and Tools
- 4. Features of the Application**
  - 4.1 Task Creation and Management
  - 4.2 Task Filtering and Sorting
  - 4.3 Data Persistence
- 5. System Design and Architecture**
  - 5.1 Class Structure and Relationships
  - 5.2 Functional Components
- 6. User Interface and Interaction**
  - 6.1 Command-Line Interface (CLI)
  - 6.2 User Navigation Flow
- 7. User Flow and Functionality**
  - 7.1 Adding, Viewing, and Updating Tasks
  - 7.2 Saving and Exiting
- 8. Benefits of the Application**
  - 8.1 Enhanced Productivity
  - 8.2 Simplified Task Management
- 9. Challenges and Future Enhancements**
  - 9.1 Identified Challenges
  - 9.2 Suggested Improvements
- 10. Conclusion**
  - 10.1 Final Remarks

## References

# Project Report: To Do List Application

## 1. Project Overview

The **To Do List** application is a task management tool developed to help users organize, track, and manage their tasks efficiently. The application allows users to categorize their tasks, prioritize them based on urgency, set due dates, and mark them as completed when done. By offering a simple yet comprehensive interface, it aims to boost productivity and assist users in achieving their goals, whether in academic, personal, or professional settings.

The goal of this project is to create an easy-to-use, functional, and intuitive platform where users can manage their daily, weekly, or long-term tasks in one place. The system also provides features to save and load tasks from a persistent data file, ensuring that users' tasks are stored and can be retrieved whenever needed.

## 2. Objectives

The main objectives of the **To Do List** project are:

- To create a user-friendly task management system that allows adding, updating, deleting, and viewing tasks.
- To categorize tasks into different groups such as "Student," "Family," and "Other" for better organization.
- To allow users to set priorities for tasks (High, Medium, Low) and filter tasks based on their completion status or priority level.
- To support the inclusion of special tasks with tags like "Urgent" or "Important" for better task prioritization.
- To store task data in a file that persists across program runs, using a simple yet effective format (JSON).
- To provide a summary of tasks using basic statistics, like counts of tasks by priority, for better tracking and analysis.

## 3. Technologies Used

The **To Do List** application is developed using the following technologies:

- **Programming Language:** Python 3
- **Libraries Used:**
  - **json:** For reading and writing data to a JSON file, ensuring tasks are saved and loaded properly.
  - **numpy:** For generating task statistics, such as counting tasks by their priority level.
  - **datetime:** For handling and manipulating task due dates.
- **Storage Format:** The application uses a simple JSON format to store tasks persistently. This allows the application to load and save tasks even after it is closed and reopened, ensuring that no task data is lost.

## 4. Features of the Application

The **To Do List** application offers the following features:

- **Task Creation:** Users can easily add new tasks by providing a description, category, priority, and due date. Optional tags such as "Urgent" or "Important" can be added to tasks to give them special significance.
- **Task Management:** Tasks can be updated and deleted as required. Users can change the task

description, category, due date, or priority at any time, ensuring flexibility in task management.

- **Task Completion:** Users can mark tasks as completed when they are done. This action is reflected in the task list by a visual marker (✓ or ✗).
- **Task Filtering:** Users can filter tasks based on:
  - Completion status (All, Completed, Pending).
  - Priority level (High, Medium, Low).
  - Due date.
- **Task Summary:** The application provides a summary of tasks based on priority levels. Using the numpy library, it can count how many tasks fall under each priority category, providing a simple statistical overview.
- **Data Persistence:** Tasks are saved to a tasks.json file, allowing users to maintain their tasks between sessions. When the application is restarted, it loads the task data from this file, ensuring that no data is lost.

## 5. System Design and Architecture

The application is designed using **Object-Oriented Programming (OOP)** principles. The main components of the system include:

- **Task Class:** This class defines the attributes of a task, including the task ID, description, category, due date, priority, and completion status. Methods within this class enable task creation, completion marking, and string representation for displaying tasks.
- **SpecialTask Class:** This class inherits from the **Task** class and adds the ability to tag tasks with labels like "Urgent" or "Important." This class leverages inheritance and polymorphism to extend the functionality of the basic task.
- **ToDoList Class:** This class manages a collection of **Task** and **SpecialTask** objects. It includes methods to add tasks, show tasks with filtering options, remove tasks, update task details, and save/load tasks to/from a JSON file.

The **ToDoList** class serves as the core of the application, interacting with the task data and providing methods to manage and interact with tasks. It also handles the task loading and saving mechanisms, ensuring data persistence across sessions.

## 6. User Interface and Interaction

The application operates through a **command-line interface (CLI)**. Upon launching, the user is presented with a text-based menu that includes several options to interact with the application. The menu allows users to:

- Add new tasks with details like description, category, due date, and priority.
- View all tasks or filter them by completion status (completed or pending) or by priority level (High, Medium, Low).
- View tasks by specific due dates.
- Mark tasks as completed.
- Update task details or delete a task from the list.
- View a summary of tasks categorized by their priority levels (using simple statistics).
- Save tasks and exit the program.

## 7. User Flow and Functionality

1. **Start:** The user launches the program, and the task list is loaded from a tasks.json file (if available).

2. **Menu Navigation:** The user selects an option from the main menu to interact with tasks:
  - Option to add a new task.
  - Option to filter and view tasks.
  - Option to update, delete, or mark tasks as completed.
3. **Task Management:** Tasks are displayed on the screen, and users can add, modify, or delete tasks as needed.
4. **Saving Data:** After making changes, the user can save tasks back to the tasks.json file.
5. **Exit:** The program closes after saving tasks, and all data is saved for the next session.

## 8. Benefits of the Application

- **Organized Task Management:** Users can efficiently organize their tasks by category and priority, ensuring that they focus on important tasks first.
- **Improved Productivity:** By providing tools to filter, prioritize, and track task completion, the application helps users stay on top of their responsibilities.
- **Simple and Flexible Interface:** The application is intuitive and easy to use. It provides flexibility in managing tasks without overwhelming the user.
- **Data Persistence:** The JSON file storage ensures that tasks are not lost between sessions, making it easy for users to continue where they left off.

## 9. Challenges and Future Enhancements

While the **To Do List** application provides a basic task management solution, there are several areas where it could be improved:

- **Error Handling:** There could be more advanced error handling mechanisms to prevent the program from crashing when users provide incorrect inputs (e.g., invalid dates, incorrect task IDs).
- **Graphical User Interface (GUI):** A graphical interface would make the application more accessible and user-friendly. Adding buttons, date pickers, and task views would enhance the user experience.
- **Task Reminders and Notifications:** Adding reminders or notifications for upcoming tasks would make the application more functional, helping users stay on track with deadlines.
- **Synchronization:** Syncing tasks across multiple devices or platforms would allow users to access their to-do list on the go.

## 10. Conclusion

The **To Do List** application is an efficient and effective tool for managing tasks. It is simple to use, highly customizable, and powerful enough to handle a wide variety of task management needs. With its ability to categorize tasks, set priorities, and mark tasks as completed, it serves as a practical solution for individuals looking to stay organized and productive. Future improvements, such as error handling and a graphical interface, could further enhance the application's usability and functionality, making it even more powerful for managing daily tasks.

## References

1. Python Software Foundation. (n.d.). Python documentation. Retrieved from <https://docs.python.org/>
2. W3Schools. (n.d.). Python tutorial. Retrieved from <https://www.w3schools.com/python/>