**Case Study: Simple To-Do List Management System**

**Problem Statement**

Time management is one of the most essential skills in both personal and professional life. In a fast-paced world, many individuals struggle to stay organized and track their daily responsibilities, leading to missed deadlines, low productivity, and stress. The **Simple To-Do List Management System** addresses this challenge by providing users with a basic, interactive task management tool that allows for efficient handling of day-to-day activities.

**Objectives**

The primary objective of this case study is to design and implement a Python-based console application that allows users to:

1. **Add new tasks** to a list.
2. **Remove tasks** from the list once completed.
3. **View all pending tasks** in an organized manner.
4. **Exit the application** gracefully.

These features are aimed at enhancing productivity, supporting task tracking, and promoting efficient time management.

**System Requirements**

To ensure a smooth and intuitive user experience, the system must adhere to the following requirements:

* **User-Friendly Interface**: A menu-driven command-line interface that presents options clearly and takes appropriate user inputs.
* **Dynamic Task Management**: Use of Python data structures (lists) to manage tasks dynamically.
* **Function-Based Design**: Use modular functions to implement each feature for clean, reusable, and maintainable code.
* **Input Validation**: Implement exception handling to manage invalid inputs without crashing the program.

**Functional Requirements**

The application must provide the following functionalities:

**1. Add Task**

* Users can input a task description.
* Duplicate tasks should not be allowed.
* The task is appended to the list and acknowledged.

**2. Remove Task**

* Users are shown a numbered list of current tasks.
* They can remove a task by entering its corresponding number.
* The system should handle out-of-range inputs or empty task lists gracefully.

**3. View Tasks**

* Display all pending tasks with a serial number for clarity.
* If there are no tasks, display an appropriate message.

**4. Exit**

* End the program with a thank-you or goodbye message.

**Implementation in Python**

python

CopyEdit

# Case Study: Simple To-Do List Management System

def display\_menu():

print("\nOptions:")

print("1. Add Task")

print("2. Remove Task")

print("3. View Tasks")

print("4. Exit")

def add\_task(task\_list):

task = input("Enter the task to add: ").strip()

if task in task\_list:

print("Task already exists. Try adding a new task.")

elif task == "":

print("Empty task cannot be added.")

else:

task\_list.append(task)

print(f"Task '{task}' added successfully.")

def remove\_task(task\_list):

if not task\_list:

print("No tasks to remove.")

return

print("\nCurrent Tasks:")

for i, task in enumerate(task\_list, 1):

print(f"{i}. {task}")

try:

index = int(input("Enter the task number to remove: "))

if 1 <= index <= len(task\_list):

removed\_task = task\_list.pop(index - 1)

print(f"Task '{removed\_task}' removed successfully.")

else:

print("Invalid task number.")

except ValueError:

print("Please enter a valid number.")

def view\_tasks(task\_list):

if not task\_list:

print("No pending tasks.")

else:

print("\nPending Tasks:")

for i, task in enumerate(task\_list, 1):

print(f"{i}. {task}")

def main():

tasks = []

while True:

display\_menu()

try:

choice = int(input("Enter your choice: "))

if choice == 1:

add\_task(tasks)

elif choice == 2:

remove\_task(tasks)

elif choice == 3:

view\_tasks(tasks)

elif choice == 4:

print("Thank you for using the To-Do List Management System. Goodbye!")

break

else:

print("Invalid choice. Please select from 1 to 4.")

except ValueError:

print("Please enter a numeric value.")

# Run the program

main()

**Sample Output**

mathematica

CopyEdit

Options:

1. Add Task

2. Remove Task

3. View Tasks

4. Exit

Enter your choice: 1

Enter the task to add: Submit assignment

Task 'Submit assignment' added successfully.

Options:

1. Add Task

2. Remove Task

3. View Tasks

4. Exit

Enter your choice: 3

Pending Tasks:

1. Submit assignment

Options:

1. Add Task

2. Remove Task

3. View Tasks

4. Exit

Enter your choice: 2

Current Tasks:

1. Submit assignment

Enter the task number to remove: 1

Task 'Submit assignment' removed successfully.

Options:

1. Add Task

2. Remove Task

3. View Tasks

4. Exit

Enter your choice: 4

Thank you for using the To-Do List Management System. Goodbye!

**Challenges & Considerations**

* **Avoiding Duplicates**: The system checks for existing tasks before adding a new one.
* **Invalid Input Handling**: Includes input validation for both task numbers and menu selections.
* **User Experience Enhancements** *(Future Scope)*:
  + Add task **priority levels** (High, Medium, Low).
  + Include **timestamps** for when a task was added.
  + Implement **task completion status** (e.g., Done/Pending).
  + Save tasks to a file or database for **persistent storage**.

**Conclusion**

The Simple To-Do List Management System provides an interactive, beginner-friendly application to organize daily tasks efficiently. With modular design, robust error handling, and clear logic, this system lays the foundation for more advanced task management solutions with features like reminders, deadlines, and data persistence.