

## Planception: A Smart Task Manager for Multidisciplinary Minds

Terione R. - C. Martin

UAT

## Planception: A Smart Task Manager for Multidisciplinary Minds

### Overview

Planception is a task management app designed to help individuals with diverse commitments and neurodivergent thinking patterns (especially ADHD) organize their projects and daily priorities. It is engineered to balance creativity and structure, allowing users to sort tasks by urgency, domain, and cognitive load. The application lays the foundation for a future AI assistant that will optimize schedules, recognize productivity patterns, and provide smart task suggestions.

### Scope

Planception will allow users to create, organize, and prioritize tasks across multiple domains (e.g., school, creative work, business). The system will display tasks using filters (by category, due date, urgency) and highlight daily priorities.

#### What It Will Do:

- Add, update, and delete tasks
- Assign tasks to categories
- Set due dates and priority levels
- Filter/sort by urgency and domain
- Highlight “Today’s Tasks”
- Lay groundwork for AI integration (expandable task model, time analytics-ready)

#### What It Will Not Do (yet):

- Calendar or full scheduling integration (Google/Outlook)
- AI-based suggestions or time management
- Mobile app support (desktop-first design)

- User authentication/multiple profiles

### **Functional Requirements**

1. FR1: The system shall allow users to create tasks with a title, due date, priority, and category.
2. FR2: The system shall display a list of tasks sorted by due date by default.
3. FR3: The system shall allow filtering tasks by category and/or priority.
4. FR4: The system shall highlight tasks that are due today or overdue.
5. FR5: The system shall allow users to edit or delete existing tasks.
6. FR6: The system shall validate that task names are not empty and due dates are valid.
7. FR7: The system shall display a dedicated “Today’s View” for urgent tasks.
8. FR8: The system shall persist task data between sessions using a database.
9. FR9: The system shall support responsive layout for smaller windows.
10. FR10 (Stretch): The system shall support future integration of AI task suggestions via modular design.

### **Non-Functional Requirements**

1. Usability: The system should be usable without training, even for neurodivergent users.
2. Performance: The application should respond to user actions in under 1 second.
3. Maintainability: Code should follow clean design patterns (MVC), enabling future AI expansion.
4. Accessibility: The interface should use clear fonts, high contrast, and keyboard navigation.

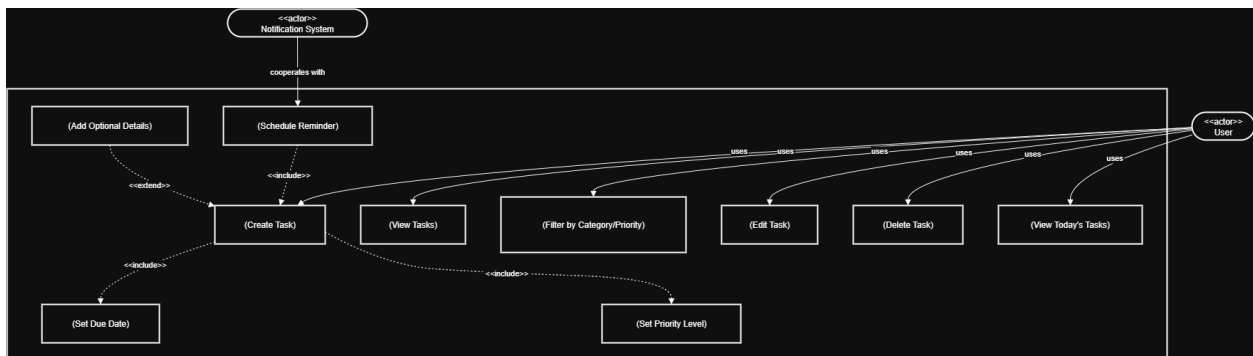
5. Portability: The system should work on all major browsers and desktop operating systems.

## Use Case Diagram

Actor: User

Use Cases:

- Create Task
- Edit/Delete Task
- View Tasks
- Filter Tasks
- Sort Tasks
- Highlight “Today’s Tasks”



## UML Class Diagram (need to make)

- Task
  - id, name, due\_date, priority, category, status
- TaskManager
  - add\_task(), update\_task(), delete\_task(), filter\_tasks(), get\_today\_tasks()
- UIController (optional)
  - handles interface rendering



## Glossary and Assumptions

### Glossary:

- Task: A user-defined activity that needs to be completed
- Priority: A flag indicating task importance (e.g., low, medium, high)
- Category: A domain such as "Writing," "School," or "Life"
- Today's View: A filtered screen that shows tasks relevant to the current day
- Planception: The application name, inspired by "plan" + "Inception"

### Assumptions:

- The user is a solo creator with many roles and overlapping projects.
- The user wants to avoid context-switching and overwhelm.

- Tasks do not need to be shared or synced between users (for MVP).
- The project will grow into a larger AI-based assistant.

### **Tech Stack**

- Language: Python
- Framework: Flask (simple web app)
- Database: SQLite (lightweight, no setup overhead)
- Front-end: HTML/CSS + Bootstrap for quick UI
- Testing: pytest

### **Conclusion**

Planception represents a comprehensive, user-focused solution designed to address the productivity challenges faced by multidisciplinary and neurodivergent individuals. We also see a better future by combining AI-driven task suggestions, customizable views, and cross-platform accessibility, which bridges the gap between task management and creative ideation. This Software Requirements Specification (SRS) outlines the application's scope, core features, quality attributes, and design through the use of UML and Use Case diagrams. The elements ensure a shared understanding among stakeholders and provide a clear roadmap for implementation, from this starting point to the potential future AI implementation. As development progresses, the structured foundation provided by this SRS will reduce misunderstandings, enhance collaboration, and help deliver a robust, intuitive, and scalable product that empowers users to organize their ideas and projects efficiently.

## References

- Ambler, S. W. (2005). The elements of UML 2.0 style. *Cambridge University Press*. [The Elements of UML 2.0 Style | PDF](#)
- Lucid Software. Unified Modeling Language (UML) Tutorial. *Lucidchart*. [Unified Modeling Language \(UML\) Tutorial | Lucidchart](#)
- OMG. (2017). OMG Unified Modeling Language (OMG UML), Version 2.5.1. *Object Management Group*. [About the Unified Modeling Language Specification Version 2.5.1](#)
- OpenAI. (2023). Best practices for prompt engineering with OpenAI API. OpenAI. [Best practices for prompt engineering with the OpenAI API](#)