UltraTodoPro – Technical Documentation

# 1. Overview

UltraTodoPro is a single‑window desktop application built with Tkinter that enables users to create, manage, and track personal tasks in a modern dark‑themed interface. This document covers the structural design, major components, and quality‑risk mitigation strategies, following the API documentation structure recommended by Document360.

# 2. Versioning & Release Info

• Current version: 1.0 (June 2025)

• Semantic versioning is followed (MAJOR.MINOR.PATCH).

# 3. Architecture Diagrams

## 3.1 Class Diagram

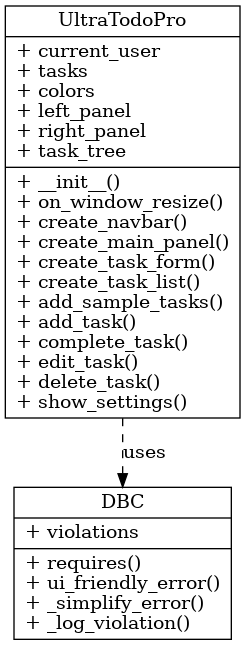


Figure 1 – UML class diagram showing relationships between UltraTodoPro (GUI), DBC (Design‑by‑Contract layer), and task storage.

## 3.2 Component Diagram

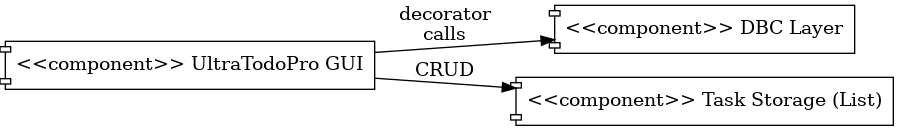


Figure 2 – High‑level component diagram illustrating GUI, Contract layer, and Task Storage interactions.

# 4. Class Descriptions

## 4.1 UltraTodoPro

**Role:** Main GUI class that orchestrates user interactions and delegates data operations to Task storage while enforcing contract checks via DBC decorators.  
**Key Attributes:** current\_user, tasks, colors, task\_tree.  
**Key Responsibilities:** render UI, handle CRUD actions, resize columns, invoke settings.

## 4.2 DBC (Design‑by‑Contract Layer)

**Role:** Reusable decorator set enforcing pre‑conditions and user‑friendly error prompts.  
**Key Attributes:** violations (list of contract breaches).  
**Key Responsibilities:** requires, ui\_friendly\_error, \_simplify\_error, \_log\_violation.

## 4.3 Task Storage (List)

**Role:** In‑memory list that holds tuples representing individual tasks.  
**Structure:** (id, title, description, status).

# 5. Risk Factor – Ease of Use & Mitigation Strategies

The primary quality risk selected for UltraTodoPro is Ease‑of‑Use. The following literature‑backed strategies are integrated into the development cycle:

|  |  |  |
| --- | --- | --- |
| Strategy | Implementation Details (Programming Context) | Key Reference |
| User‑Centered Design (UCD) | Conduct paper‑prototype walkthroughs with representative users each sprint, then adjust Tkinter layouts and widget labels accordingly. Track usability issues in a CSV checklist and close them before release. | ISO 9241‑210 |
| Heuristic Evaluation | Twice per sprint, two developers review every screen against Nielsen’s 10 heuristics, grading 1–4 severity; any 3/4 is logged as a blocking defect. | Nielsen 1994 |
| Iterative Usability Testing | Automate nightly builds that launch UltraTodoPro with demo data; record task‑completion time and fail CI if median time regresses >10 %. | Iterative Design Methodology |
| Accessibility (WCAG 2 Contrast) | Enforce 4.5 : 1 colour contrast in self.colors, ensure keyboard shortcuts for all actions, and add tooltips for icon‑only buttons. | WCAG 2.1 SC 1.4.3 |

# 6. Error Handling

UltraTodoPro surfaces validation errors (e.g., empty task title) via modal dialogs and logs contract violations in the DBC layer. Each breach is appended to the violations list for later audit.

# 7. Authentication

Not applicable – UltraTodoPro is a local desktop application.

# 8. Rate Limiting

None – operations are in‑process and unlimited.

# 9. Interaction Summary

Although UltraTodoPro is not a remote API, its public interactions can be viewed as callable operations:  
• Add Task – creates a new task entry.  
• Update Task – edits title or description.  
• Toggle Status – switches between Pending and Completed.  
• Delete Task – removes a task by ID.  
These interactions are available via the GUI and are enforced by DBC contracts.

# 10. References

- ISO 9241‑210:2019 – Human‑centred design for interactive systems.

- Nielsen, J. (1994). Heuristic Evaluation in Usability Inspection Methods.

- Iterative Design – Wikipedia article (May 2025).

- W3C WAI. Understanding SC 1.4.3: Contrast (Minimum).