To-Do List Application

Stephen

September 6, 2025

Contents

1 A Simple Design Document

2 Introduction

This document outlines the high-level design for a simple command-line interface (CLI) to-do list application. The application will serve as a foundational tool for a single user to manage, track, and filter their daily tasks. The design will follow a modular approach, separating core business logic from the user interface.

2.1 Core Features

The application will support the following key features:

Task Management:

1. Adding a new task with a description and a due date.
2. Marking an existing task as complete.
3. Deleting a task by its unique identifier.

Task Viewing:

1. Listing all tasks in a human-readable, formatted string.
2. Filtering tasks to show only those that are completed or incomplete.

3 Data Model

The application's state will be managed by a TodoManager struct, which contains a collection of Task structs. The data model is as follows:

3.1 Task

A Task represents a single to-do item.

1. ID (int): A unique identifier for the task.
2. Description (string): The description of the task.
3. Completed (bool): A boolean flag indicating whether the task is complete.
4. DueDate (time.Time): The due date of the task.

3.2 TodoManager

The TodoManager holds and manages all tasks.

1. tasks (map[int]*Task): A map that stores pointers to Task structs, with the task ID as the key.

This allows for efficient lookup and modification.

1. count (int): An integer to generate new, unique task IDs.

4 Architectural Considerations

The application will follow a clear separation of concerns:

4.0.1 Business Logic:

The TodoManager and Task structs handle all core logic, such as adding, completing, and filtering tasks. This logic is fully tested via the TDD process.

4.0.2 User Interface:

A separate main function (not yet implemented) will handle user input from the command line and display output by calling the business logic methods. This keeps the core logic clean and testable, independent of how the user interacts with it.