Design Specification

Title: Task flow

By:Paolo lauricella Done entirely by me

Course: COP4331

Platform:Java Swing GUI (Desktop)

Group: Paolo Lauricella, Besenja Joseph, Charlie Holder Functional

Requirements for TaskFlow

Task Management

1. **Add new task with title, description, due date, priority, tags, and optional reminder.**

2. **Edit existing task details with live validation. 3. Delete tasks and move them to Trash with Undo option.**

4. **Mark tasks as completed and toggle back to pending if reopened.**

5. **Support recurring tasks (daily, weekly, monthly).**

6. **Allow bulk operations delete multiple or mark several tasks as completed.**

7. **Implement task cloning duplication for quick creation of similar tasks.**

Organization & Filtering

Filter tasks by tagg, due date, priority, or completion status.

8. **Sort tasks by due date, priority, creation time, or alphabetical order.**

9. **Group tasks by project, tag, or week in calendar view.**
10. **Provide colorr coded labels or badges for priority levels and overdue tasks.**

Calendar & Scheduling

12. **Dual view List View and Calendar View (Day/Week/Month). 13. Allow dragand drop in Calendar View to change task due dates.**

14. **Display upcoming tasks in a “Today” or “This Week” summary section.**

15. Highlight overdue tasks visually in red orange within the calendar.

Reminders & Notifications

16. Optional pop up or sound reminder before due time.

17. Allow customizable reminder times (5 min, 30 min, 1 hr, 1 day before).

18. Support desktopp notifications even when app minimized. 19. Show missed reminders upon reopening the application.

User Interface & Accessibility

20. Responsive UI that adjusts layout on window resize.

21. Keyboard shortcuts for adding, deleting, or marking tasks complete.

22. Dark mode / light mode tooggle for accessibility. 23. Tooltip hints for icons and buttons.

24. Confirmation dialogs before destructive actions. Persistence & Data Management

25. Automatically save and load data locally in JSON or database.

26. Allow import/export to CSV, JSON, or XML.

27. Maintain history of task channges (versioning).

28. Store user preferences (theme, sort order, default reminder time).

Error Handling & System

29. Clear and descriptive error messages for invalid inputs. 30. Log errors and events for debugging and analytics.

31. Handle missing or corrupted save files gracefully with recovery options.

2. Main Use Cases

- **UC1: User Adds a Task**

The user createss a new task by entering a title, description, due date, and optional reminder or priority.

- **UC2: User Edits a Task**
  The user modifies details of an existing task, such as the title, due date, or reminder settings.

- **UC3: User Deletes or Restores a Task**
  The user deletes one or more tasks, which are moved to the Trash. Deleted tasks can be restored or permanently removed.

- **UC4: User Marks Task as Completed**
  The user marks a task as completeed. The system updates its status and refreshes the interface accordingly.

- **UC5: User Views Tasks in Calendar or List View**
  The user switches between List and Calendar views to visualize tasks by date or category.

- **UC6: User Filters and Sorts Tasks**
  The user filters tasks by priority, due date, or completion status and sorts them alphabetically or chronologically.

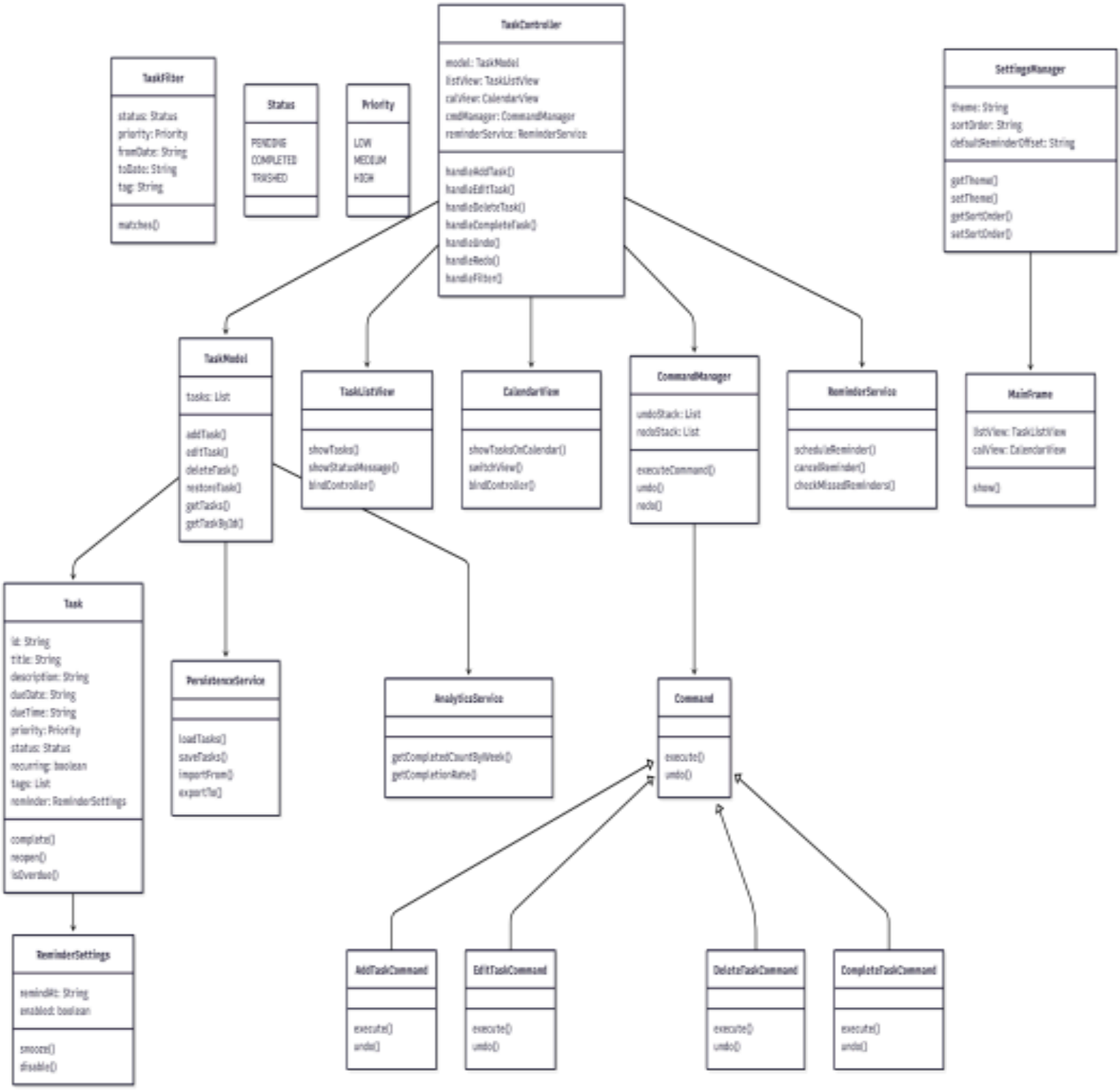
- **UC7: User Sets or Modifies a Reminder**
  The user customizes reminder notifications for upcoming tasks.

- **UC8: System Triggers Task Reminder**
  The system automatically sends alerts or notifications when a task is due.

- **UC9: User Imports or Exports Tasks**
  The user imports task data from external files or exports tasks to JSON or CSV for backup.

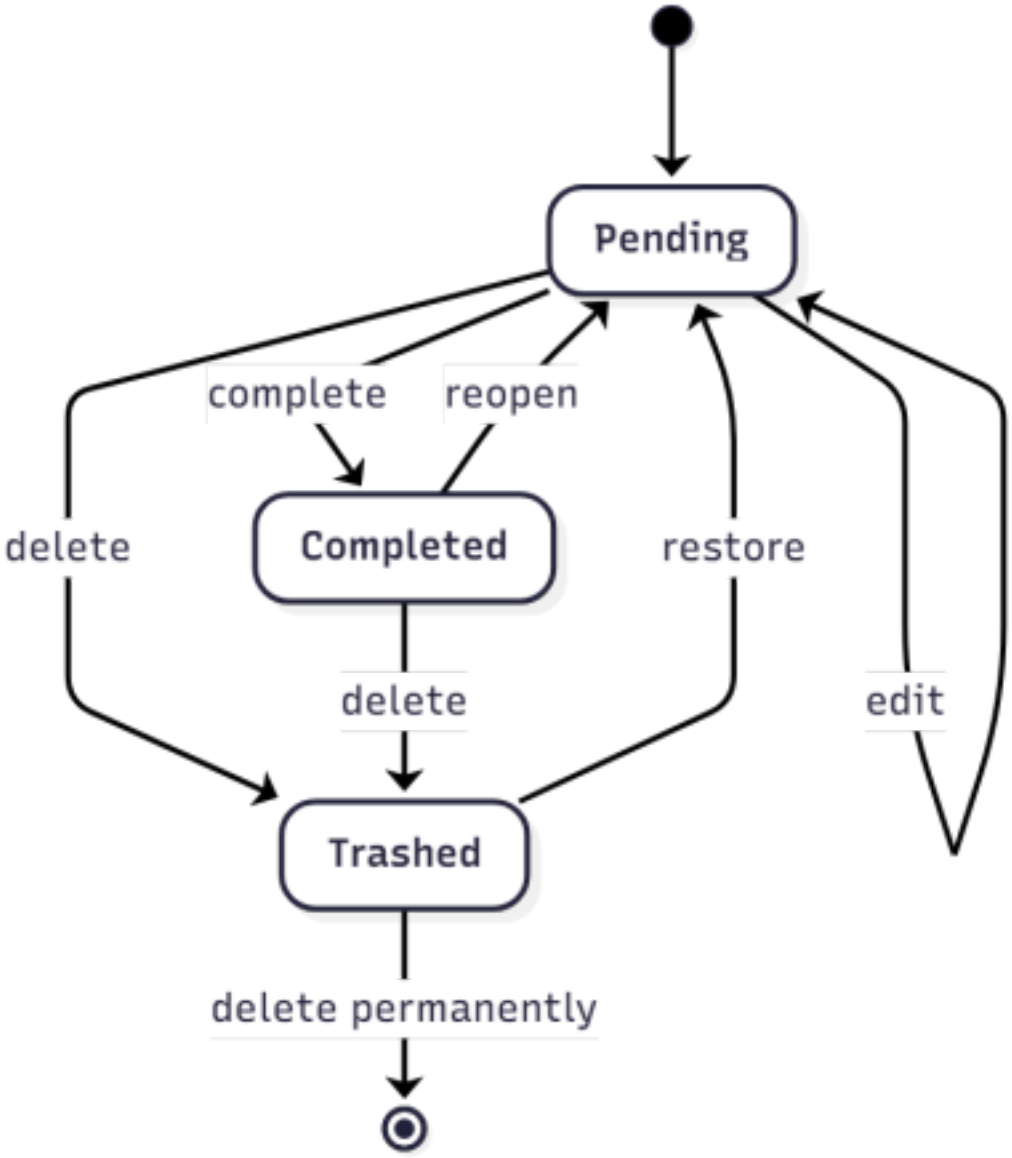
- **UC10: User Customizes Application Settings**
  The user changes preferences such as theme (dark/light), default remiinder time, and display options.
  Github link here
  https://github.com/Snowball2002/COP4331-Taskflow/tree/a6503955b204a5fd41c7ecf06ac72b690756fe4e **Also invited but public just in case**

3. UML Class Diagram TaskFlow (MVC + Services)

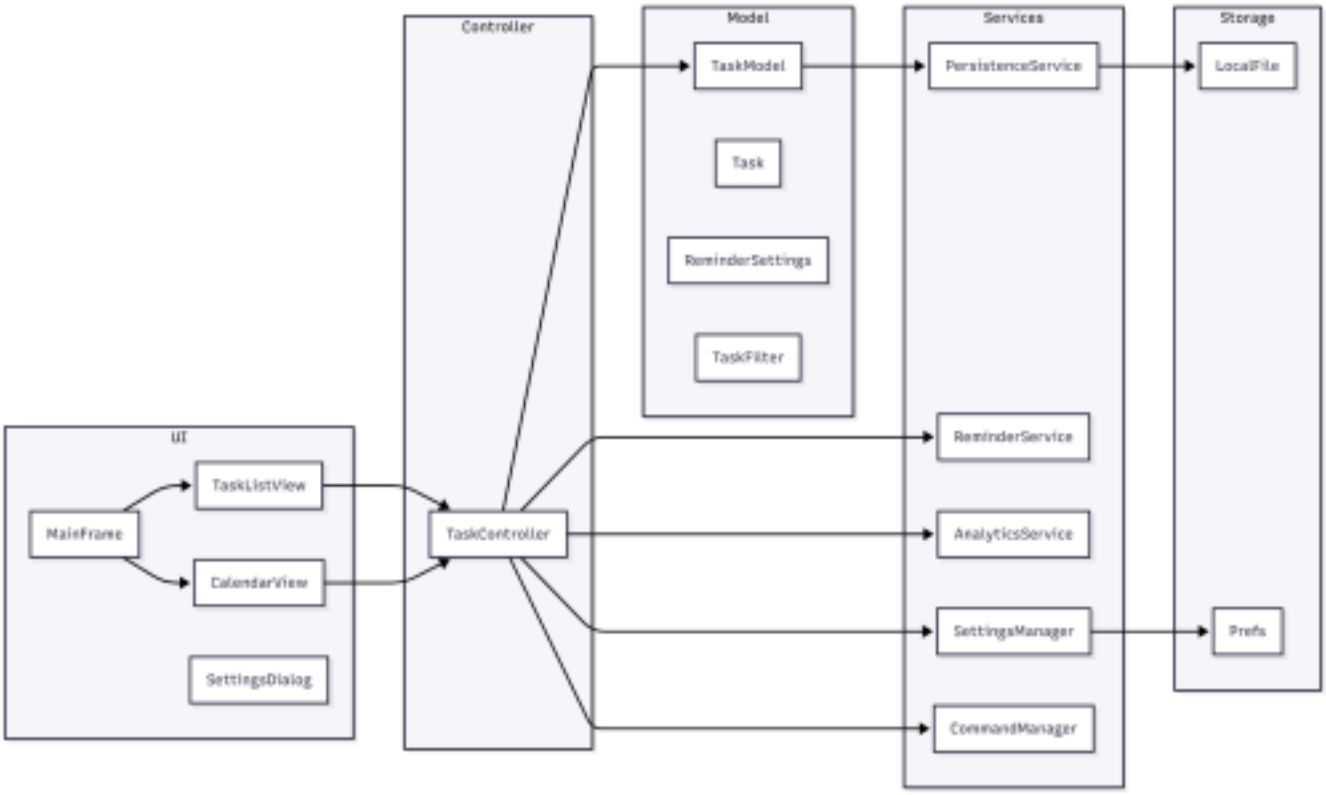


4. Sequence Diagrams Below

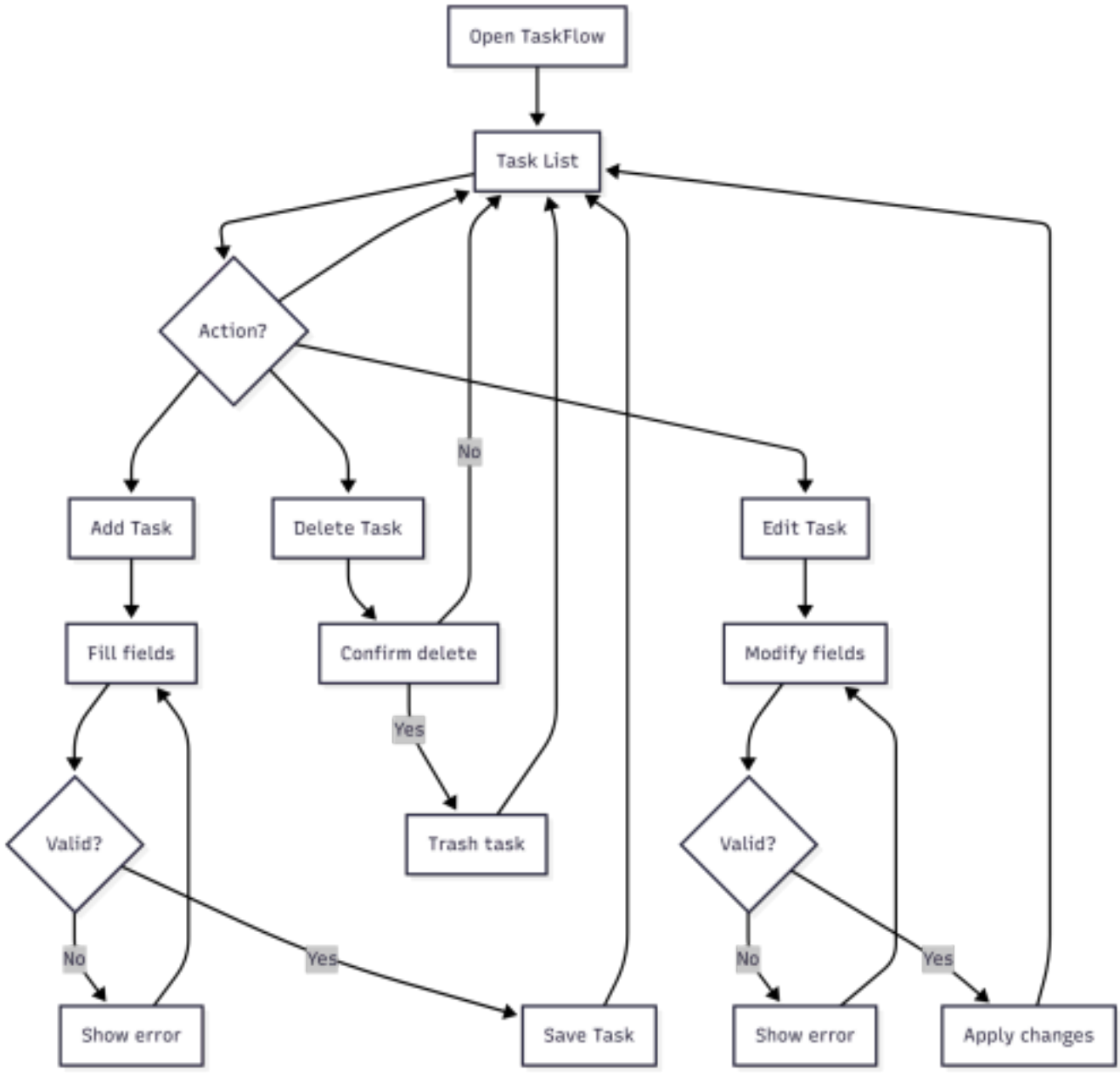
Add a Task
5. Sequence Diagram Mark Task as Completed Below
6. Sequence Diagram Delete Task + Undo (Command Pattern)
7.Sequence Diagram Reminder Trigger Missed Reminder handling
8. State Diagram Task lifecycle



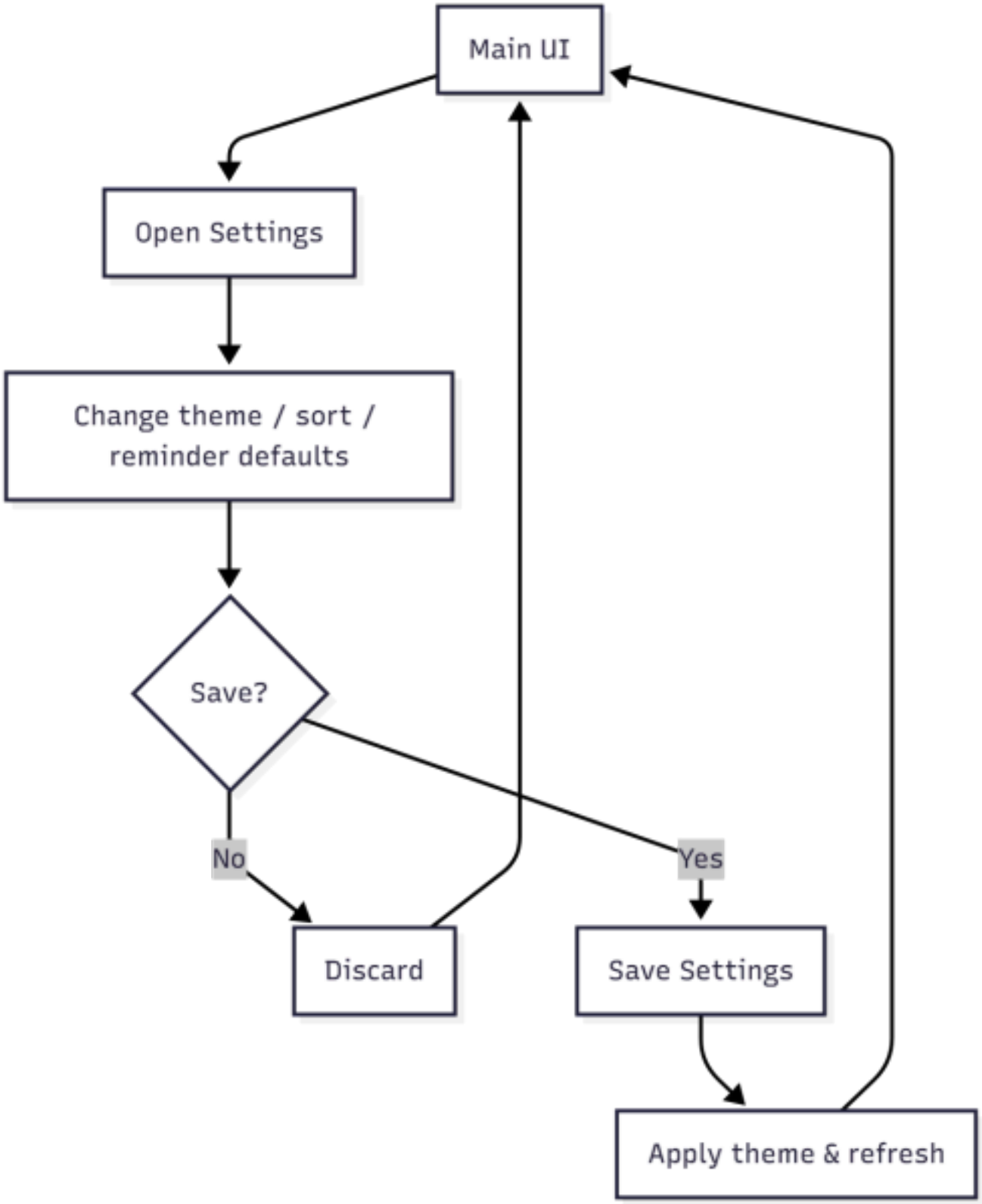
9. *System architecture* and Component Diagram
Task flow below



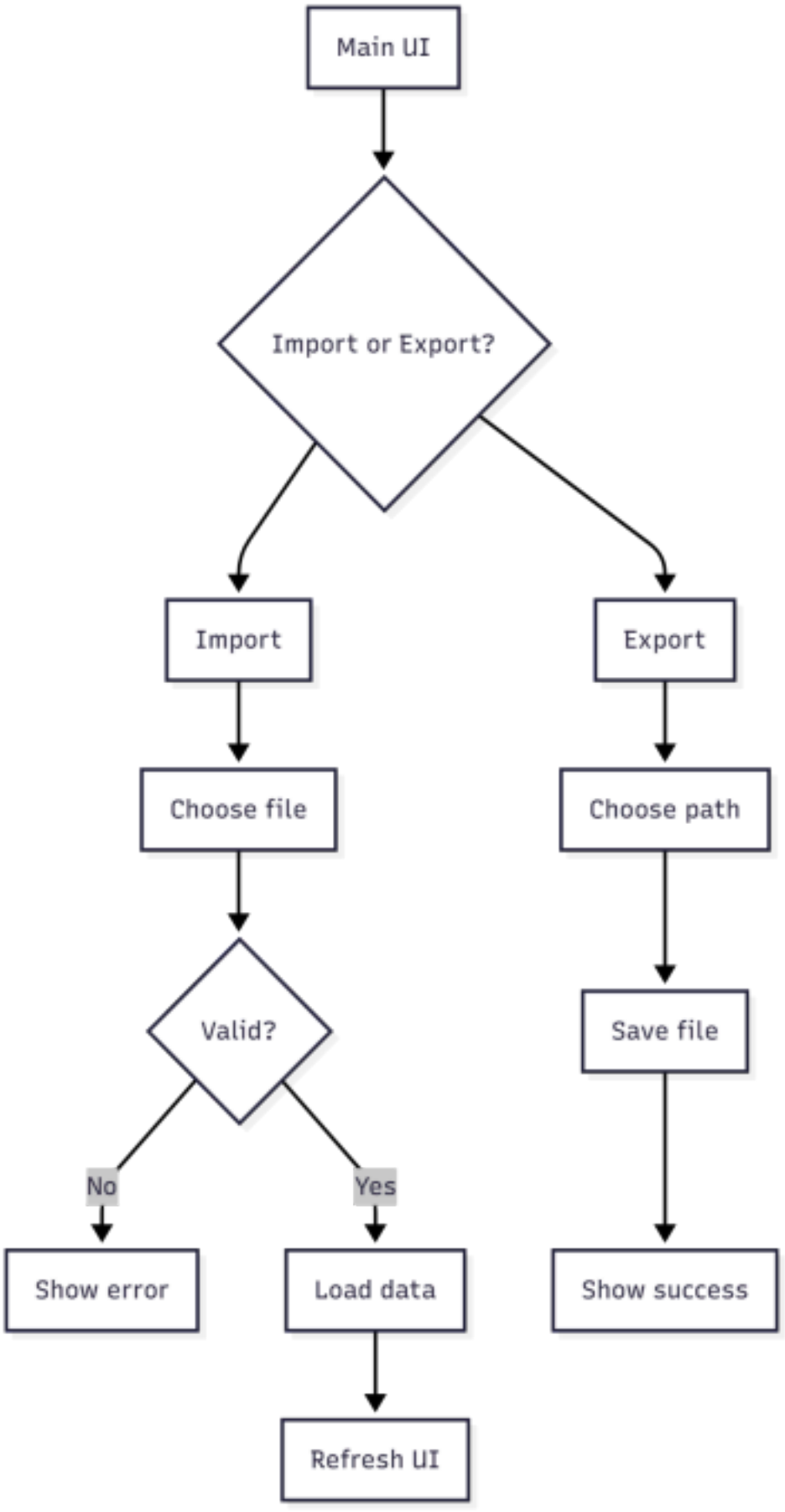
10. Activity Diagram Add/Edit/Delete Task (with recurring and bulk) Below



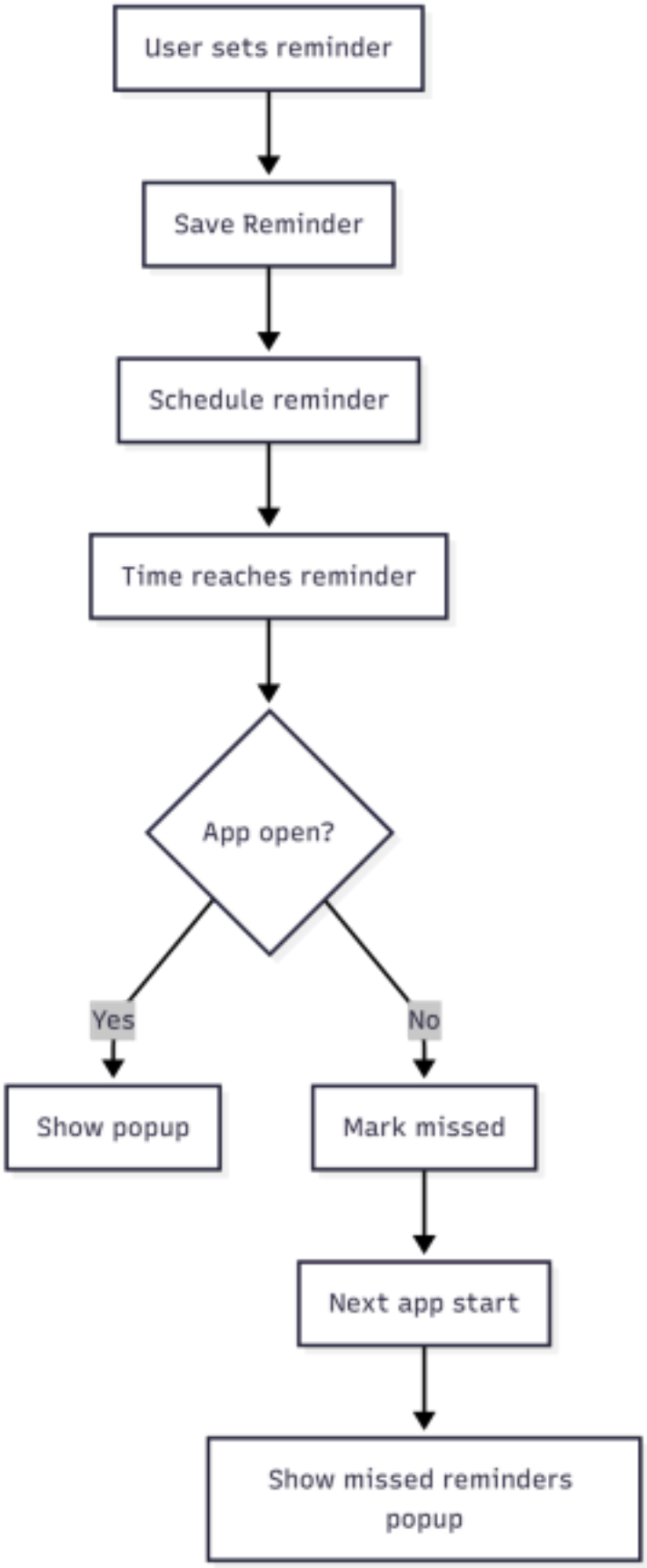
11. Activity Diagram Change Theme / Preferences



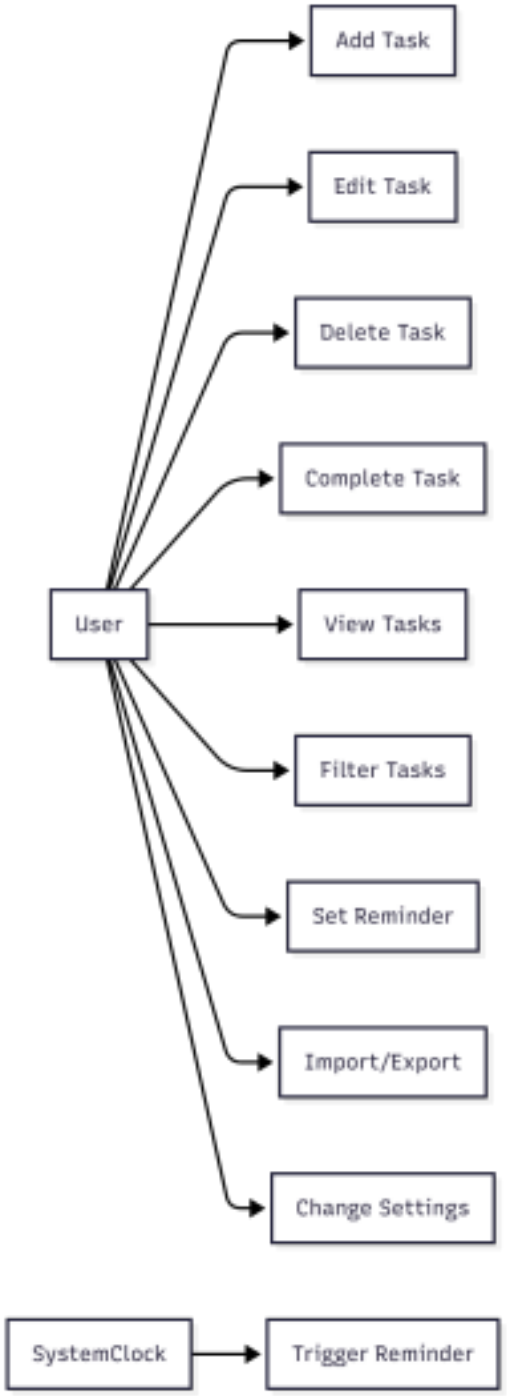
12. Activity Diagram Import/Export Data



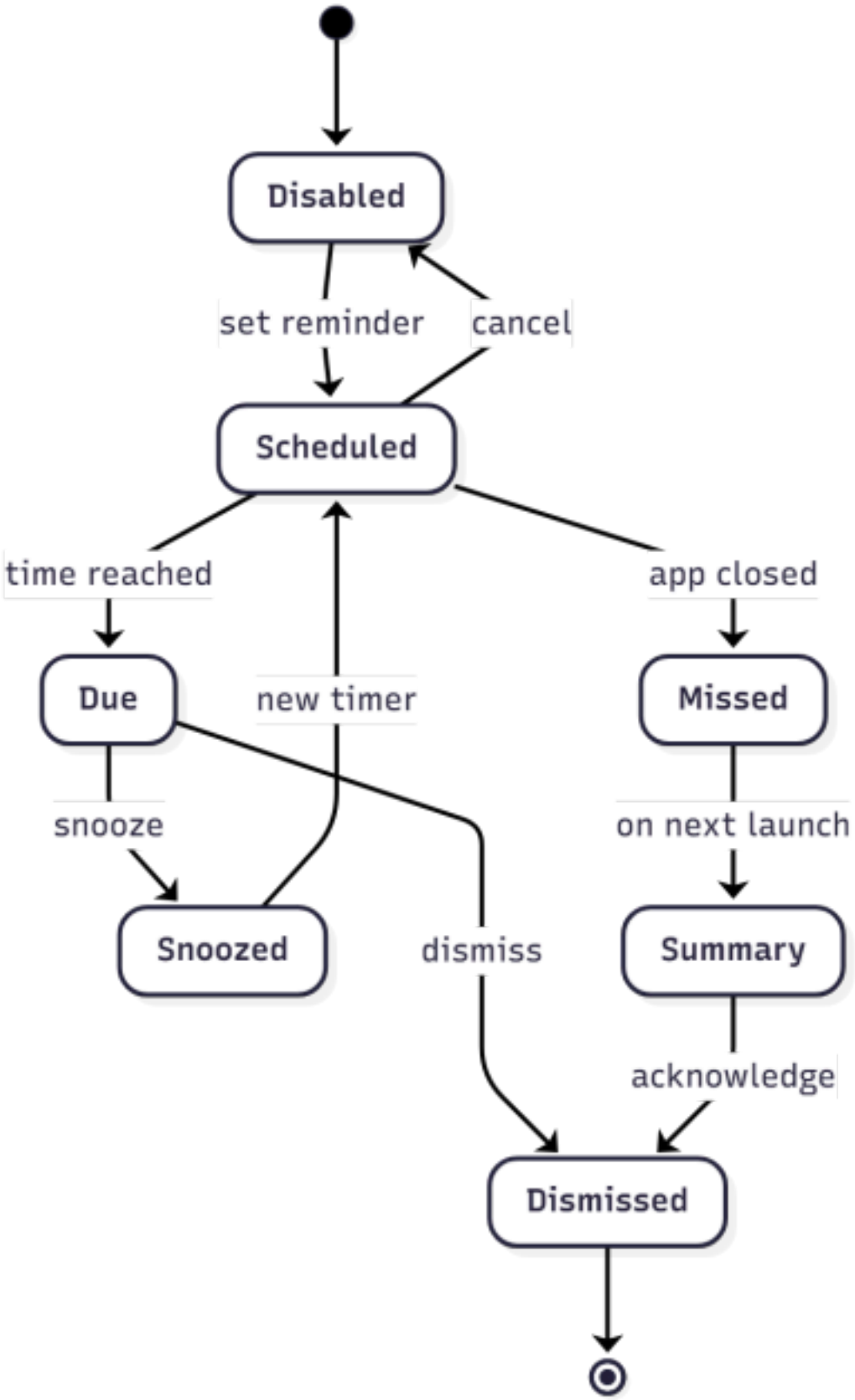
13. Activity Diagram Set Reminder and Trigger Alert Below



14. Use case diagram



15. State diagram


Disabled
set reminder
cancel
Scheduled
time reached
app closed
Due
new timer
Missed
snooze
on next launch
Snoozed
dismiss
Summary
acknowledge
Dismissed

Update code included in design spec