Cognitive Wellness & Memory Aid App – Light Planning Spec

General Description of the Project

The Cognitive Wellness & Memory Aid App is a digital reminder and task management system designed for elderly individuals and users with mild cognitive decline. The app allows users or caregivers to create reminders for daily routines, medication schedules, appointments, and important personal events. Notifications are delivered as text or audio prompts, and users can mark tasks as completed, skipped, or deferred.
The app is intended to be highly accessible and simple to use, with a clean interface optimized for mobile devices. Optional features include caregiver integration, visual cues (icons/images), and basic logging of task completion for monitoring.
The app will be implemented primarily as a web app using Flask, with HTML/CSS/JS frontend. The initial version could also run via CLI for prototype testing, with reminders appearing in the console. Additionally, a future API endpoint could allow remote creation or updating of reminders from other apps or devices.
The main backend mechanisms include Python datetime for scheduling, APScheduler or similar for background notifications, and gTTS or pyttsx3 for optional text-to-speech audio prompts. Task data will be stored locally in JSON or CSV format for Version 1.

Task Vignettes (User Activity Flow)

1. Creating a Reminder

Alice wants to set a daily reminder to take her 10 am medication. She opens the app and clicks the “New Reminder” button. A dialog appears with:
Task name text field

Time picker
Repeat frequency dropdown (once, daily, weekly)
Optional notes or image upload
Alice enters the details and clicks “Save.” The reminder is stored locally, scheduled in the background, and will trigger notifications at the correct time.
Technical details:
User input stored as JSON dictionary: {task_name, time, frequency, notes, image_path}
Scheduled tasks handled via APScheduler
Optional TTS audio prompt generated at notification time

2. Receiving & Responding to a Reminder

At 10 am, Alice receives a notification: “It’s time to take your morning medication.” An icon of a pill is displayed, and a short audio prompt plays. Alice taps the notification and marks the task as completed. The task is logged in the local file/database.
Technical details:
Notification triggers a TTS audio message (gTTS/pyttsx3)
User interaction updates task status (completed, skipped, or deferred)
Logs stored locally for optional caregiver review

3. Reviewing Task History

Alice wants to see how well she’s keeping up with her medication. She opens the “Task History” page, which shows a table of completed and missed tasks over the past week, and a simple bar chart of adherence.
Technical details:
Task history extracted from local JSON or CSV logs
Simple plotting via matplotlib or Plotly for graphs
Optional: caregiver can view the same dashboard via shared API endpoint

4. Optional Caregiver Task Creation
Bob, Alice’s son, wants to add a new reminder remotely. He uses an optional API to send a task creation request:
```
{ "task_name": "Call doctor", "time": "14:00", "frequency": "once" }
```

The app validates the input and schedules the reminder on Alice’s device.
Technical details:
Flask API endpoint receives JSON payload
Input validation ensures time format and required fields
APScheduler schedules the task immediately

Technical Flow
Overall Data Flow:
User Input → task name, time, frequency, optional image/audio
Backend Scheduler → APScheduler sets up reminders based on input
Notification Engine → TTS/audio or text notification triggers at scheduled time
User Response → updates status of task (completed/skipped/deferred)
Task Logging → local storage (JSON/CSV)
Visualization Module → optional charts/dashboard for history review

Data Structures:
Task: dictionary with keys {task_name, time, frequency, status, notes, image_path}
TaskLog: list of task dictionaries with timestamps for status updates
Blocks:
Input Module → forms or CLI input
Scheduler → manages background task execution
Notification Module → TTS/audio + visual notification
Logging Module → stores completed/skipped/deferred status
Visualization Module → displays historical adherence graphs
User Interaction Blocks:
Input: Create/Update task
Output: Notification and audio prompt
Input: Mark completion
Output: History dashboard

(Somewhat) Unknowns for Version 1:
How to handle multiple simultaneous notifications
Notification reliability across devices/browsers

Best layout for task history and adherence graph
Optional caregiver remote API integration

Version 1 Progress Report A

Methods:

Conducted 3 informal interviews with elderly users and caregivers.
Reviewed existing reminder/task management apps for accessibility features.

Key Findings:
Elderly users sometimes forget medications or daily routines, especially when schedules change.
Notifications must be clear, with options for audio and text prompts.
Users prefer minimal, simple interfaces to avoid confusion.
Caregivers want an easy way to track users' completed tasks.

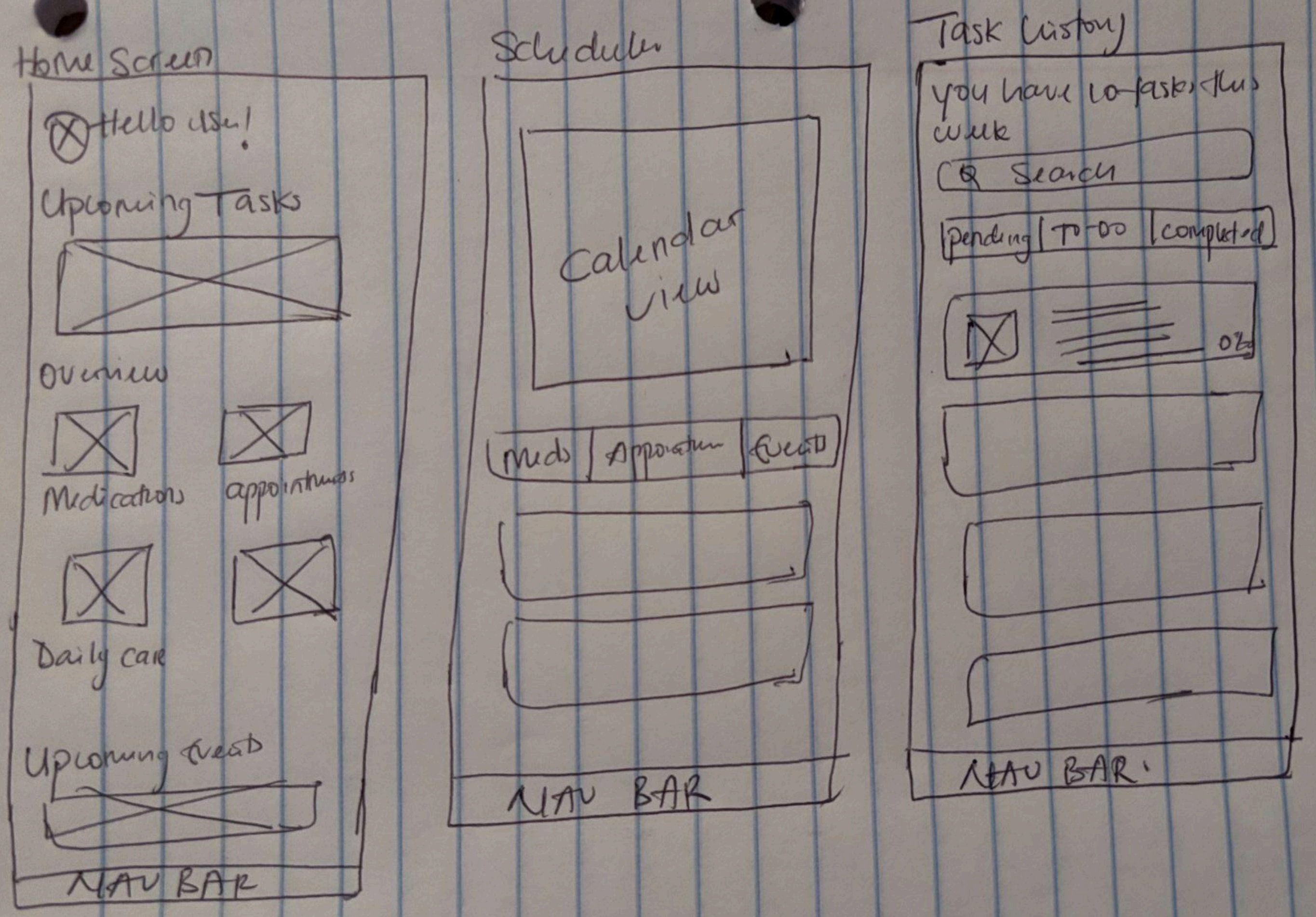
Design Implications:
Include both audio and text notifications.
Simple task management interface with clear buttons for completed, skipped, or deferred.
Allow caregiver monitoring or shared task management.

Personas

Persona 1 – Grace, 78 (Elderly User)
Goals: Remember daily routines and medication on time.
Frustrations: Easily forgets tasks; small or cluttered interfaces are confusing.
Design Need: Clear, minimal UI with audio prompts and simple task management.

Persona 2 – Samuel, 55 (Caregiver)
Goals: Help his mother manage medications and appointments without constant reminders.
Frustrations: Hard to track completed tasks; notifications sometimes ignored.
Design Need: Dashboard to view task status, ability to add or adjust reminders quickly.

Persona 3 – Linda, 70 (Elderly User with mild cognitive decline)
Goals: Maintain independence while remembering personal events and routines.
Frustrations: Easily overwhelmed by multiple notifications; hard to navigate menus.
Design Need: Minimal notifications, optional audio cues, simple "mark done" interaction.


Home Screen
Hello user!
Upcoming Tasks
Overview
Medications
appointments
Daily care
Upcoming events
NAV BAR
Schedule
Calendar view
Meds
Appointm
Events
NAV BAR
Task (history)
you have 10 tasks this week
Search
pending
TO-DO
completed
0%
NAV BAR.