# DayBoard User Manual

## Introduction

DayBoard is a personal productivity and finance dashboard for macOS and iOS. It combines your calendar, subscription bills, commute, and internship pay into one clean, black‑and‑white interface reminiscent of premium apps like Uber or OpenAI. The app runs locally on your device and communicates only with your own Supabase database and external APIs through a secure Go backend.

## Key Features

1. **Next Meeting** – Connect your Google Calendar and view your next event for today. The event includes a one‑click “Join” button that opens Meet, Zoom, or Teams. Local notifications alert you ten minutes before each meeting.
2. **Subscriptions** – Link your bank through Plaid to automatically detect recurring charges. DayBoard reminds you before each subscription renews. You can also add subscriptions manually or import from a CSV file.
3. **Commute Estimate** – DayBoard calculates the distance and duration between your home and office using Google’s Distance Matrix API, then estimates a ride‑share fare based on a simple cost model stored in the database. You can override the estimate or adjust a surge slider.
4. **Pay Outlook** – Enter your hourly or stipend pay, state, pay frequency, and hours worked. DayBoard calculates after‑tax take‑home pay using federal and state tax tables seeded in your database. When combined with commute and food costs, the app shows your net savings for today and for the entire internship term.
5. **Today’s Burn** – When you have an in‑office day, DayBoard adds commute and lunch costs to “Today’s burn,” helping you understand how daily expenses impact your take‑home pay.

## How It Works

• **Client** – The SwiftUI client calls the Go backend using JSON REST endpoints. The client stores no sensitive information; all secrets stay on the server. Local notifications are scheduled by the client. • **Backend** – The Go server authenticates with Google Calendar, Plaid, and the Distance Matrix API, and stores encrypted OAuth tokens in Supabase. It normalizes data and computes taxes and commute estimates. • **Subscriptions** – The backend polls Plaid transactions (or processes CSV uploads) and detects recurring merchants using deterministic rules: the same merchant and amount recurring within a cadence window. Detected subscriptions are saved to the database with a next due date. • **Tax Estimator** – The server reads federal and state tax brackets from tables seeded by SQL migrations and applies them to your annualized income. FICA (Social Security and Medicare) is included. Results are returned per paycheck and per term. • **Commute Estimator** – The server calls the Distance Matrix API to obtain distance and duration, then multiplies them by city‑specific cost factors (base fare, per mile, per minute) stored in the database. Default values target Indianapolis but can be updated.

## Data Privacy

DayBoard is designed to be local‑first and privacy‑conscious. The backend runs in your own environment and stores tokens encrypted at rest. No personal data is sent to third‑party servers except the providers you authorize (Google and Plaid). You can delete your data at any time by calling /profile with an empty body.

## Sample Flow

1. You open DayBoard; the menu bar shows “Standup at 10:00 – Join.” Clicking **Join** opens the Meet link in your browser.
2. Before lunch the app reminds you that Spotify renews tomorrow for $9.99. You can snooze or cancel directly from the Subscriptions tab.
3. If today is an in‑office day, DayBoard shows “Commute: Office (3.2 mi, 14 min) · est $12–$15” and adds $12 for lunch to Today’s burn.
4. In the Finances tab you see “Estimated pay this week $682 · Net term: $8,540.” After subtracting rent and subscriptions, you know exactly what you’ll save.

## Future Improvements

• Reconcile actual paycheck deposits via Plaid (v2).  
• Support Microsoft Outlook/Calendar.  
• Add multiple city cost models and real ride‑share quotes.  
• Provide push notifications via APNs.  
• Sync settings through iCloud to multiple devices.  
• Visualize spending and time with charts.

## Testing & Troubleshooting

• Always test locally with sandbox API keys before going live.  
• Use curl or Postman to call backend endpoints and verify JSON responses.  
• If you receive “Invalid credentials” errors, ensure your Google and Plaid credentials and redirect URIs match your environment.  
• Watch the Go server logs in your terminal; most issues will appear there.  
• Make sure your local machine has internet access; the app needs to reach Google, Plaid, and Supabase.

## FAANG Engineering Strength

DayBoard demonstrates secure API integration with Google Calendar, Plaid, and Google’s Distance Matrix, handling OAuth flows and encrypted token storage. It showcases backend concurrency in Go, relational data modeling, and deterministic tax and cost computations. On the front end, it implements a minimalist, polished SwiftUI interface reminiscent of premium apps built by top tech companies. Together, these skills illustrate strong systems design, user‑centric thinking, and an ability to ship real, useful software.