

Here's a comprehensive status report for handoff:

FuelBal Multi-Aircraft System - Development Status Report

Date: December 31, 2025

Phase: Multi-Aircraft Foundation + Fuel Options UI

Branch: Main

PROJECT OVERVIEW

FuelBal is a native iOS fuel tracking app for N215C (Piper Cherokee 6) with plans to support multiple aircraft configurations. The app uses a unique fuel burn methodology that infers flight mode (balanced vs endurance) based on user behavior at Swap #2.

Aircraft Config: 4-tank system (L TIP: 17 gal, L MAIN: 25 gal, R MAIN: 25 gal, R TIP: 17 gal)

Bundle ID: com.dossindustries.fuelbal

Platform: iOS 17+, SwiftUI

COMPLETED FEATURES

Data Architecture

-  Aircraft model (tail number, manufacturer, model, ICAO, fuel type, tank array)
-  FuelTank model (position enum, capacity)
-  TankPosition enum (lTip, lMain, rMain, rTip with rawValue labels)
-  FuelType enum (AVGAS 6.0 lb/gal, JET-A 6.8 lb/gal)
-  N215C preset hardcoded in Aircraft.n215c static property
-  All models Codable and Identifiable

Color System

-  Centralized Colors.swift with semantic names
-  Colors: appBackground, cardBackground, primaryText, secondaryText, accentText, fuelActive, fuelLow, buttonPrimary/Secondary/Disabled, tankLeft/tankRight
-  All views migrated to use color system (no hardcoded hex colors)

UI Screens - BUILT

1. **AircraftSelectionView** 

 - Shows N215C card with tail number, manufacturer/model, ICAO, total capacity
 - Tappable card triggers fullScreenCover with FuelOptionsView
 - Clean black background with cyan accents

2. **FuelOptionsView** 

 - **Two Modes Based on Saved State:**
 - **No Saved State:** TOP OFF, TABS, QUANTITY OVERRIDE
 - **Saved State:** RESUME FLIGHT, ADD FUEL (primary), QUANTITY OVERRIDE (tertiary)
 - Aircraft header (tail, manufacturer, model, ICAO, capacity)
 - Custom splash panel with 4 tank inputs + live total calculation
 - Add Fuel panel with current/add/new display per tank
 - Smooth animations on panel expand/collapse

3. **FlightView**  (V1 Prototype - Fully Functional)

 - Header with shutdown (X) and undo (↔) buttons
 - Phase indicator (MAINS/TIPS/BALANCED/ENDURANCE/EXHAUSTED)
 - 4-tank gauges with fill bars, colors (active=yellow, low=red, normal=cyan)
 - Center "BURNING" indicator showing current + next tank
 - Last reading display
 - Swap targets (climbout 7.0, dual targets for Swap #2, warnings for limits)
 - Totalizer input with validation
 - LOG SWAP button
 - Swap history (last 4 swaps)
 - **NEW:** Shutdown prompt sheet with totalizer input and validation

4. **StartView**  (V1 - Working but Deprecated)

 - N215C title + "Fuel Tracker"
 - Resume box (if saved state exists)
 - TOP OFF / TABS preset buttons
 - QUANTITY OVERRIDE button + panel
 - Add Fuel panel (for resuming flights)

NAVIGATION FLOW

Current Working Flow:

1. App launches → **ContentView**
2. **ContentView** checks `fuel.isFlying`

3. If `false` → `StartView` (V1 working screen)
4. If `true` → `FlightView`
5. User taps TOP OFF/TABS → `fuel.startFlight(.topoff/.tabs)` → sets `isFlying = true` → navigates to `FlightView`
6. User logs swaps, flies, then taps X → `ShutdownPromptView` sheet appears
7. User enters totalizer reading → `fuel.shutdown(reading: X)` → calculates final burn → sets `isFlying = false` → returns to `StartView`

New Flow (Partially Built, NOT WIRED UP):

1. App launches → `ContentView` → `AircraftSelectionView` (NOT YET DEFAULT)
2. User taps N215C → `FuelOptionsView` appears via `fullScreenCover`
3. User selects option → [NOT WIRED] Should start flight but doesn't yet
4. [TODO] Wire `FuelOptionsView` buttons to actually call `fuel.startFlight()`

CRITICAL SHUTDOWN FLOW

New Feature: Accurate Fuel Bookmarking at Shutdown

Previously, fuel levels were only accurate to the last swap. Now:

1. User taps X in `FlightView`
2. `ShutdownPromptView` sheet appears
3. User enters current totalizer reading (validated \geq last reading)
4. `FuelState.shutdown(reading:)` calculates fuel burned since last swap
5. Updates `tankBurned[currentTank]` with final burn amount
6. Creates special shutdown entry in `swapLog` (marked with "(SHUTDOWN)")
7. Saves accurate fuel state to `UserDefaults`
8. Sets `isFlying = false` → returns to `StartView`

This enables accurate "ADD FUEL" mode when resuming.

FILE STRUCTURE

```

/fuelbal/
  ├── fuelbalApp.swift          // App entry point
  ├── ContentView.swift         // Root view, switches
  between StartView/FlightView
  └── AircraftSelectionView.swift // NEW - Shows aircraft
  list (only N215C for now)

```

```

└── FuelOptionsView.swift           // NEW - Fuel selection for
aircraft (2 modes)
└── FlightView.swift              // V1 - Main fuel tracker UI
(fully functional)
└── StartView.swift               // V1 - Deprecated start
screen (still works)
└── FuelState.swift               // V1 - Core fuel logic +
persistence
└── Colors.swift                 // NEW - Centralized color
system
└── TestView.swift                // Debug view (can delete)
  └── /Models/
    └── Aircraft.swift           // NEW - Aircraft data model

```

WHAT'S NOT WIRED UP YET

FuelOptionsView → FlightView Connection

-  TOP OFF button prints "TOP OFF tapped" but doesn't start flight
-  TABS button prints "TABS tapped" but doesn't start flight
-  QUANTITY OVERRIDE "Start Flight" button calls `startCustomFlight()` but doesn't actually start
-  RESUME FLIGHT button calls `resumeFlight()` but doesn't navigate to FlightView
-  ADD FUEL "Start Flight" button calls `addFuelAndStart()` but doesn't start

Why: FuelOptionsView doesn't have access to the FuelState instance yet. It only has an Aircraft object.

Fix Needed: Pass `@ObservedObject var fuel: FuelState` to FuelOptionsView, then call `fuel.startFlight()` in button actions.

ContentView Default Screen

-  App still launches to `StartView` (V1) instead of `AircraftSelectionView` (V2)
- **Current:** ContentView hardcoded to show `StartView`
- **Needed:** Change to `AircraftSelectionView` once FuelOptionsView is wired up

Per-Aircraft Fuel State

- ~~✗~~ `hasSavedState` currently checks GLOBAL FuelState (all aircraft share one state)
- **Needed:** Separate UserDefaults keys per aircraft (`fuelState_N215C`, `fuelState_N123AB`, etc.)
- **Needed:** `savedFuelLevels` should load from aircraft-specific storage

Missing UI Features

- ~~✗~~ QUANTITY OVERRIDE "Reset Warning" alert (when `showResetWarning = true`)
- ~~✗~~ Totalizer update prompt after ADD FUEL (tell user "Added 7.0 gal, update totalizer to XX.X")
- ~~✗~~ Pre-flight targetRMN confirmation (prompt before starting engine)

KNOWN ISSUES

1. **Optional Binding Bug:** Had to hardcode `Aircraft.n215c` in `AircraftSelectionView` sheet instead of using `selectedAircraft` optional binding (black screen issue)
2. **Namespace Conflicts:**
 - `PresetButton` defined in both `FuelOptionsView` and `StartView` → Renamed to `FuelPresetButton` in `FuelOptionsView`
 - `TankInput` defined in both → Renamed to `FuelTankInput` in `FuelOptionsView`
 - `AddFuelPanel` defined in both → Renamed to `AddFuelOptionsPanel` in `FuelOptionsView`
3. **Color System Inconsistencies:**
 - Some modifiers require `Color.` prefix
(`.background(Color.cardBackground)`)
 - Others work with shorthand (`.foregroundColor(.primaryText)`)
 - Rule: Use `Color.` prefix when passing to functions/modifiers
4. **StartView Border Visibility:** Preset buttons had black borders on black background (fixed with white borders at 0.15 opacity)

FUEL BURN METHODOLOGY (V1 - Working)

Phase System:

- MAINS phase: L MAIN → R MAIN (balanced) or extended burns (endurance)
- TIPS phase: L TIP → R TIP (always balanced, fixed 8.5 gal burns)

Mode Detection (Swap #2 Only):

- Climbout (Swap #1): Always 7.0 gallons
- Swap #2: Dual targets shown (balanced + endurance)
 - If reading \leq midpoint → BALANCED mode locked
 - If reading $>$ midpoint → ENDURANCE mode locked
- Swap #3+: Single target based on locked mode

Burn Calculations:

- Balanced: 12.5 gal per main tank
- Endurance: 18.0 gal per main tank
- Tips: 8.5 gal per tip tank
- Safety reserve: 0.9 gal (not usable)
- Exhausted threshold: 2.0 gal (red, faded)

NEXT STEPS (Priority Order)

Immediate (Wire Up V2 Flow)

1. Pass `FuelState` to `FuelOptionsView` via `AircraftSelectionView`
2. Update button actions in `FuelOptionsView` to call `fuel.startFlight()`
3. Test full flow: Aircraft Selection → Fuel Options → Flight → Shutdown → Resume
4. Change `ContentView` default to `AircraftSelectionView`

Per-Aircraft Persistence

5. Implement aircraft-specific `UserDefaults` keys
6. Update `hasSavedState` to check per-aircraft
7. Update `savedFuelLevels` to load per-aircraft
8. Add "IN FLIGHT" badge to aircraft cards when `isFlying = true`

Missing Prompts/Alerts

9. Add "Reset Warning" alert for QUANTITY OVERRIDE when saved state exists
10. Add totalizer update prompt after ADD FUEL ("Added X.X gal, update to Y.Y")
11. Add pre-flight targetRMN confirmation (user-friendly wording TBD)

UI Polish

12. Add barber pole animation to active tank gauge
13. Add `observedGPH` tracking (editable in-flight parameter)
14. Add "Time Until Next Swap" HUD metric (requires `observedGPH`)
15. Add fuel weight display (gallons \times `weightPerGallon`)

Future Features (Post-Launch)

16. Aircraft Editor (create/edit custom aircraft)
17. Multi-aircraft list (not just N215C)
18. Multi-engine support (dual totalizers, dual burn tracking)
19. Lock Screen widget (HUD essentials)
20. 3/5 tank configurations (aux/ferry tanks)

CODE QUALITY NOTES

Good:

- Clean separation of concerns (models, views, state)
- Centralized color system
- ObservableObject pattern for state management
- UserDefaults persistence working

Needs Improvement:

- V1 (StartView/FlightView) and V2 (AircraftSelectionView/FuelOptionsView) coexist - confusing
- Hardcoded N215C tank capacities in multiple places (should reference `Aircraft.n215c`)
- No error handling on UserDefaults decode failures
- `FuelState` uses string keys for tanks ("ITip") instead of `TankPosition` enum
- Global shared `FuelState` instead of per-aircraft instances

Architecture Decision Needed:

- Keep V1 flow as fallback? Or fully migrate to V2?
- How to handle aircraft switching mid-flight? (Should be prevented)
- Should `FuelState` be per-aircraft or global with aircraft ID tracking?

TESTING CHECKLIST

Tested & Working:

-  Aircraft Selection screen renders
-  Tapping N215C shows Fuel Options
-  Custom Splash panel expands/collapses
-  Tank inputs update live totals
-  Shutdown prompt validates totalizer input
-  Shutdown calculates final fuel burn correctly
-  Saved state persists across app restarts
-  V1 flow (StartView → FlightView) fully functional

-  Swap logging, undo, fuel exhaustion logic

Not Yet Tested:

-  ADD FUEL mode (panel renders but doesn't start flight)
-  RESUME FLIGHT (button exists but doesn't navigate)
-  Per-aircraft state isolation
-  Multi-aircraft list (only N215C exists)
-  FuelOptionsView → FlightView navigation

DEPENDENCIES

- iOS 17.0+
- SwiftUI
- Combine (for ObservableObject)
- UserDefaults (persistence)

No External Libraries

GIT STATUS

Last Commits:

1. "Add shutdown prompt with totalizer reading and rename Custom Splash to Quantity Override"
2. "Add Fuel Options screen with TOP OFF and TABS presets"
3. "Update StartView to use centralized color system"
4. "Add multi-aircraft foundation and color system"

Branch: main

Remote: [github.com/\[user\]/n215c-fuel](https://github.com/[user]/n215c-fuel)

QUESTIONS FOR HANDOFF

1. Should we fully migrate to V2 architecture or keep V1 as fallback?
2. FuelState refactor: Per-aircraft instances or single global with aircraft tracking?
3. Priority: Wire up V2 navigation first, or implement per-aircraft persistence?
4. Should QUANTITY OVERRIDE be accessible when saved state exists? (Currently shows with warning)

End of Status Report