Petfolio – Week 7 Progress

Navigation & Information Architecture

This Week's Goals (Week 7)

- Implement bottom navigation with 3 tabs (Pets, Care, Profile)
- Ensure clear active tab indication and consistent access to Profile
- Preserve onboarding/AppStartup flows while introducing new shell
- Create missing Care and Profile sections
- Keep tab state when switching and across app restarts

Key Numbers (Week 7)

Lines of Code (approx.)

- New/edited this week: ~500–700 LoC
- New files created: 7
- Features completed: 2 / 6 → 33%
- Requirements completed: 5 / 15 → 33%
- Burndown Rate: 33% for Sprint 2

✅ What Was Shipped

Unified Navigation Shell

- `MainScaffold` with Material 3 `NavigationBar`
- 3 tabs: Pets, Care, Profile
- `IndexedStack` preserves state per tab
- Integrated with `AppStartup → AuthWrapper` without regressions

✅ What Was Shipped (cont.)

New Sections

- `CareDashboardPage`
  - Renders existing `CarePlanDashboard`
  - Action chips: Today’s Tasks (placeholder), Add Care Plan (→ `/edit`), Manage Pets (switch tab)
  - FAB: Today’s Tasks (placeholder for Week 8)
- `ProfilePage`
  - Shows user name, email, avatar placeholder
  - Centralized Sign Out action

✅ What Was Shipped (cont.)

Consistent Profile Access

- Top-right profile icon (`UserAvatarAction`) on Pets & Care
- Tap switches to Profile tab via `navIndexProvider`

Tab State Persistence

- `SharedPreferences` stores last-selected tab
- App reopens on last tab used

🧱 Technical Highlights

- `lib/app/main_scaffold.dart` – app shell (NavigationBar + IndexedStack)
- `lib/app/navigation_provider.dart` – Riverpod notifier + persistence
- `lib/features/care_plans/.../care_dashboard_page.dart` – Care tab page
- `lib/features/auth/.../profile_page.dart` – Profile tab page
- `lib/app/widgets/user_avatar_action.dart` – top-right profile icon

🔎 Testing & Quality

- Manual validation on Web and Mobile form factors
- Lints clean across new files
- Added lightweight tests:
  - `NavIndexNotifier` unit test
  - `MainScaffold` widget test (destinations + basic tab switch)

🧪 Demo Path

1. Launch app → onboarding/sign-in as needed
2. Land on Dashboard within `MainScaffold`
3. Switch between Pets / Care / Profile tabs
4. Use top-right profile icon from Pets/Care to open Profile
5. In Care tab, use chips: Add Care Plan → `/edit`; Manage Pets → Pets tab
6. Close and reopen app → returns to last tab

Scope Decisions

- Deferred: `go_router` integration (kept MaterialApp routes to reduce risk)
- Deferred: Tab deep links (`/pets`, `/care`, `/profile`) after initial attempt introduced instability
- Placeholder: Today’s Tasks (to be implemented with real data in Week 8)

📈 Alignment to Sprint 2

Sprint 2 – Essential UX & Navigation

- US-004: Clear navigation between sections – ✅ Met
- US-005: Visible current section indicator – ✅ Met

Delivered for Week 7:

- Bottom navigation, consistent Profile access, section creation, and state retention

🚧 Known/Remaining Items

- Care: Today’s Tasks list and actions (Week 8 – real-time sync foundation)
- Optional: Subtle tab transition animations
- Optional: Wire Care tab badge to real “today’s tasks” count

📊 Success Metrics (Week 7)

- Navigation responsiveness between tabs: < 250 ms – ✅
- App startup to dashboard unchanged – ✅
- Visual consistency across tabs – ✅
- Usability: Clear, consistent access to Pets, Care, Profile – ✅

🏁 Outcome

Week 7 established a robust navigation foundation:

- Unified shell with persistent tab state
- Actionable Care and centralized Profile
- Stable integration with onboarding/auth

Ready to proceed to Week 8: Today’s Tasks list and real-time task updates.

Next Week (Week 8 Preview)

- Build “Today’s Tasks” list (read-only → interactive)
- Real-time task completion sync groundwork (listeners, models)
- Optional: Hook Care tab badge to today’s count

Sprint 2 Burndown (features)

- Sprint 2 planned features: 6
    - Week 6: Feature 1 – Onboarding Improvements – ✅ Completed
    - Week 7: Feature 2 – Bottom Navigation System – ✅ Completed
    - Remaining: Feature 3 (Real-time Sync), Feature 4 (Lost & Found), Feature 5 (UI Improvements), Feature 6 (Error Handling)