

Petfolio – Week 8 Progress

Real-Time Task Completion Sync

This Week's Goals (Week 8)

- Implement real-time synchronization between pet owners and sitters for care tasks
- Add Stream-based methods to TaskCompletionRepository for real-time listeners
- Create providers that merge CareTask data with TaskCompletion status
- Update UI components to display completion status, timestamps, and user info in real-time
- Ensure both owner and sitter views show synchronized task completion status

Key Numbers (Week 8)

Lines of Code (approx.)

- New/edited this week: ~800–1,000 LoC
- New files created: 2
- Files modified: 8
- Features completed: 1 / 6 → Feature 3 (Real-Time Task Sync)
- Requirements completed: 3 / 15 → 20% (US-006, US-007, US-008)
- Burndown Rate: 50% for Sprint 2 (3 of 6 features)

What Was Shipped

Real-Time Task Synchronization

- **Stream-based Repository Methods** - Three new Stream methods in TaskCompletionRepository
- **Real-time Firestore Listeners** - `.snapshots()` with proper error handling
- **Completion Status Merging** - `CareTaskWithCompletion` class merges tasks with completion data
- **Real-time UI Updates** - StreamProvider-based updates across all views

What Was Shipped (cont.)

Enhanced Care Task Provider

- **careTaskWithCompletionProvider** - StreamProvider that merges tasks with completion status
- **careTaskStatsWithCompletionProvider** - Real-time statistics with completion counts
- **allPetsWithPlanCompletionProvider** - Completion-aware pet statistics for dashboard
- **Backward Compatibility** - Existing providers still work for non-real-time use cases

What Was Shipped (cont.)

UI Components & Display

- **CareTaskCard Updates** - Shows completion status, timestamps, and user names
- **TaskCompletionStatus Widget** - Reusable component for completion display
- **Dashboard Statistics** - Real-time completed/pending task counts
- **User Display Names** - Shows who completed tasks (with email fallback)

✓ What Was Shipped (cont.)

User Display Name Integration

- **User Provider** - `userDisplayNamesProvider` for batch fetching user names
- **AuthService Extensions** - `getUserById()` and `getUsersByIds()` methods
- **Batch Fetching** - Handles Firestore 10-item limit for multiple users
- **UI Integration** - All task cards show user names for completed tasks

Technical Highlights

Repository Layer

- `lib/features/sharing/domain/task_completion_repository.dart` - Stream method interfaces
- `lib/features/sharing/data/task_completion_repository_impl.dart` - Firestore stream implementations
- Uses `.snapshots(includeMetadataChanges: false)` for optimal performance
- Proper Timestamp handling and error management

Technical Highlights (cont.)

Provider Layer

- `lib/features/care_plans/application/care_task_provider.dart` - Completion-aware providers
- `lib/features/care_plans/application/pet_with_plan_provider.dart` - Completion-aware pet stats
- `lib/features/auth/application/user_provider.dart` - User display name providers
- StreamProvider pattern for real-time updates

Technical Highlights (cont.)

Presentation Layer

- `lib/features/care_plans/presentation/widgets/care_task_card.dart` - Completion status display
- `lib/features/care_plans/presentation/widgets/care_plan_dashboard.dart` - Completion statistics
- `lib/features/care_plans/presentation/pages/care_plan_view_page.dart` - User name integration
- `lib/features/sharing/presentation/widgets/task_completion_status.dart` - Reusable status widget

Technical Implementation

Stream-Based Real-Time Architecture

```
// Repository layer
TaskCompletionRepository {
    Stream<List<TaskCompletion>> watchTaskCompletionsForPet(String petId);
    Stream<List<TaskCompletion>> watchTaskCompletionsForTask(String careTaskId);
    Stream<TaskCompletion?> watchLatestCompletionForTask(String careTaskId);
}

// Provider layer
careTaskWithCompletionProvider {
    - Merges CareTask with TaskCompletion status
    - Real-time updates via Firestore stream
    - Handles multiple completions (selects latest)
}
```

Technical Implementation (cont.)

Completion-Aware Statistics

```
// Enhanced statistics with completion data
CareTaskStats {
  - total, overdue, dueSoon, today
  - completed, pending (NEW)
  - completionSummary getter
}

// Real-time statistics provider
careTaskStatsWithCompletionProvider {
  - StreamProvider with completion-aware stats
  - Updates in real-time as tasks are completed
  - Calculates completed/pending counts
}
```

Demo Path







1. **Owner Setup:** Log in as owner, create pet with care plan, generate AccessToken
2. **Sitter Access:** Log in as sitter, access pet using AccessToken
3. **Real-Time Sync:** Sitter marks task complete → Owner sees update within 3 seconds
4. **Completion Status:** Verify completion shows checkmark, timestamp, and user name
5. **Dashboard:** Check dashboard shows completed/pending statistics updating in real-time
6. **Persistence:** Close and reopen app, verify completion status persists



Current Status vs Plan



Completed (100% of Week 8 goals)

- **Stream Methods:**  All 3 Stream methods implemented
- **CareTaskWithCompletion:**  Complete with all fields and methods
- **Real-time Providers:**  Completion-aware providers implemented
- **UI Components:**  All components updated for completion status
- **User Display Names:**  Fully integrated with batch fetching
- **Dashboard Statistics:**  Completion statistics with real-time updates

Known/Resolved Issues

Resolved







- **Completion Statistics:** Added to dashboard with real-time updates
- **User Display Names:** Created provider and integrated into UI
- **Provider Complexity:** Simplified completion-aware provider implementation
- **Backward Compatibility:** Existing providers continue to work

Testing & Quality

- Manual validation with two devices/browser tabs
- Lints clean across all modified files
- Real-time sync verified (< 3 seconds update time)
- User display names verified with batch fetching
- Completion statistics verified for accuracy
- Persistence verified across app restarts

Requirements Met

Week 8 Milestones (100% Complete)





-  **Real-time Firestore Listeners** - Stream-based, not Future-based
-  **UI Updates Immediately** - Updates within 3 seconds of task completion
-  **Completion Timestamps** - Displayed with relative time formatting
-  **User Attribution** - Shows who completed tasks with display names
-  **Conflict Resolution** - Last-write-wins with server timestamps
-  **Multi-user Support** - Works for both owner and sitter

User Stories Completed

ID	User Story	Status
US-006	As a pet owner, I want to see when my sitter completes care tasks so that I know my pet is being cared for	 COMPLETED
US-007	As a pet sitter, I want to mark tasks as complete and have the owner see it so that they have peace of mind	 COMPLETED
US-008	As a pet owner, I want to see which tasks are completed and which are pending so that I can track care	 COMPLETED





Key Benefits

Real-Time Collaboration

-  **Instant Updates** - Task completions sync in real-time (< 3 seconds)
-  **User Attribution** - Clear visibility of who completed tasks
-  **Peace of Mind** - Owners can see care progress in real-time
-  **Accurate Tracking** - Completion statistics provide clear overview

Key Benefits (cont.)

Technical Excellence

-  **Stream-Based Architecture** - Efficient real-time updates via Firestore streams
-  **Scalable Design** - Handles multiple pets, tasks, and users
-  **Performance Optimized** - Batch fetching for user names, efficient queries
-  **Backward Compatible** - Existing code continues to work



Success Metrics (Week 8)

Performance Metrics

- Sync Latency: < 3 seconds (target met)
- UI Response: < 250 ms (target met)
- Update Accuracy: 100% consistency (verified)
- Real-time Performance: No lag with < 50 tasks (verified)



Project Statistics

Week 8 Additions

- **Total Files Created:** 2 new files
- **Files Modified:** 8 files
- **Core Services:** 1 new service (user provider)
- **UI Components:** 4 components updated
- **Lines of Code:** ~800-1,000 lines (excluding comments)
- **Features Implemented:** 1 major feature (Real-Time Task Sync)








Project Statistics (cont.)

Overall Sprint 2 Progress




- **Features Completed: 3 of 6 (50%)**
 - Feature 1: Onboarding Improvements (Week 6)
 - Feature 2: Bottom Navigation System (Week 7)
 - Feature 3: Real-Time Task Completion Sync (Week 8)
- **User Stories Completed: 8 of 15 (53%)**
- **Requirements Completed: 8 of 15 (53%)**

Scope Decisions

Implemented




-  Real-time Stream-based synchronization
-  Completion statistics in dashboard
-  User display names for completed tasks
-  Completion-aware statistics providers
-  Batch fetching for user names (handles Firestore limits)

Deferred

-  Push notifications for task completions (future sprint)
-  Advanced conflict resolution UI
-  Task completion history/audit trail

Alignment to Sprint 2

Sprint 2 – Multi-User Collaboration Foundation





- US-006: Real-time task completion visibility –  Met
- US-007: Sitter task completion with owner visibility – 
Met
- US-008: Completion status tracking –  Met

Delivered for Week 8:





- Real-time synchronization, completion statistics, user attribution, and dashboard integration

Known/Remaining Items

Completed This Week






-  Real-time task completion sync
-  Completion statistics in dashboard
-  User display names integration
-  Completion-aware providers

Future Enhancements

-  Push notifications for task completions (Week 9+)
-  Task completion history/audit trail
-  Advanced conflict resolution UI
-  Performance optimizations for large task lists

Outcome

Week 8 successfully implemented real-time task synchronization:

-  Stream-based real-time updates via Firestore
-  Completion status visible across all views
-  User attribution with display names
-  Dashboard statistics with real-time updates
-  Production-ready implementation

Ready to proceed to Week 9: Lost & Found mode implementation.










Next Week (Week 9 Preview)

Feature 4: Lost & Found Mode

- Add "Mark as Lost" toggle and visual status indicators
- Generate shareable lost pet posters (image + text)
- Enable one-click "Found" recovery flow
- Integrate with profile view and dashboard cards

Sprint 2 Burndown (features)






Progress: 3 of 6 Features Complete (50%)


-  **Week 6:** Feature 1 – Onboarding Improvements – 
Completed
-  **Week 7:** Feature 2 – Bottom Navigation System – 
Completed
-  **Week 8:** Feature 3 – Real-Time Task Completion Sync – 
Completed
-  **Week 9:** Feature 4 – Lost & Found Mode – Planned
-  **Week 10:** Feature 5 – UI Improvements – Planned
-  **Week 10:** Feature 6 – Error Handling – Planned



Real-Time Sync Milestone Reached!

Petfolio now provides real-time collaboration:

-  **Instant Updates** - Task completions sync in real-time
-  **User Attribution** - Clear visibility of who completed tasks
-  **Completion Tracking** - Accurate statistics and status display
-  **Multi-user Support** - Works seamlessly for owners and sitters
-  **Production Ready** - Robust implementation with error handling

The real-time task synchronization feature is complete and ready for user testing! 

Achievement Summary

Week 8 Complete: Successfully implemented real-time task completion synchronization that enables seamless collaboration between pet owners and sitters. The implementation includes stream-based real-time updates, completion statistics, user attribution, and dashboard integration, providing a production-ready multi-user collaboration foundation.

Real-Time Sync Status:  **COMPLETE** - All Week 8 requirements met

Ready for Week 9: Lost & Found mode implementation to complete essential multi-user features.