

Petfolio

Project Planning & Progress Overview

Cross-Platform Pet Care Application

Flutter • Firebase • Riverpod



Project Overview

The Problem

Pet owners face **coordination difficulties** when managing pet care across multiple caregivers:

- **No single source of truth** for pet information, diet, medical history, and care routines
- **Fragmented communication** between owners, family, and sitters
- **Complex handoffs** when transferring pet care responsibilities temporarily
- **Missed medications or feedings** due to lack of coordination
- **Emergency situations** without quick access to critical pet data



The Solution: Petfolio

Core Value Propositions

1. **Real-time Sync** - Shared care plans update instantly across all caregivers
2. **Role-Based Access Control** - Owner, co-caretaker, sitter (time-boxed), and public viewer roles
3. **Simple Handoffs** - QR code or link generation for instant sharing
4. **Single Source of Truth** - Diet, medications, routines, and behavior notes in one place
5. **Lost & Found Support** - One-tap "Mark as Lost" functionality with auto-generated posters

Vision: Keep owners, families, and sitters perfectly in sync with real-time schedules, reminders, and handoffs



Development Timeline

Two 5-Week Sprints

- Sprint 1 (Weeks 1-5): Owner MVP - Core pet care functionality
- Sprint 2 (Weeks 6-10): Essential UX & Multi-user Features

Total Duration: 10 weeks

Current Status: Sprint 2 Complete (100% of Sprint 2 features)



Sprint 1 Planning (Weeks 1-5)

Primary Goal

Ship core app for owners by end of Sprint 1

Must-Have Features

- User onboarding + Firebase Auth
- Create/edit Pet Profile (name, species, breed, age, weight, height, photo)
- Care Plan form (diet, feeding schedule, meds)
- Local notifications (feeding/med reminders)
- Read-only profile (QR/link share)
- Access token system with role-based permissions
- Sitter Dashboard for task management



Sprint 1 Progress: Week-by-Week

Week 1: Project Setup ✓

- ✓ Scaffold Flutter project
- ✓ Connect Firebase (Auth, Firestore, Storage)
- ✓ Verify local notifications package setup
- ✓ Basic app structure and routing

Week 2: Authentication System

-  User Registration & Login with Firebase Auth
-  Password Reset functionality
-  Session Management with automatic persistence
-  Protected Routes with AuthWrapper
-  User Domain Model with JSON serialization
-  Firestore Security Rules for user-specific data access
-  Migrated to Riverpod state management

Deliverables: ~1,200+ lines of code, 7 new files, complete authentication system

Week 3: Pet CRUD & Photo Upload

-  Pet Dashboard with avatars and summary
-  Create/Edit Pet form with validation
-  Photo picker with preview and progress states
-  Delete with confirmation dialog
-  PetsRepository (Firestore + Storage)
-  Real-time list updates on create/update/delete
-  Firestore & Storage security rules (owner-only access)

Deliverables: Complete pet management system with photo upload

Week 4: Care Plans & Local Notifications

-  Complete Care Plan CRUD operations
-  Feeding schedules with multiple times per day
-  Medication tracking with dosage and timing
-  Diet instructions and guidelines
-  Enhanced Pet Profiles (emergency contacts, medical history)
-  CareScheduler Service for notification scheduling
-  Permission management UI
-  Timezone-aware scheduling system
-  Automatic notification integration

Deliverables: ~3,000+ lines of code, 25+ new files, complete care planning system

Week 5: Sharing & Access Tokens

-  Shareable read-only profile (QR/link)
-  Public profile view accessible via QR/link
-  AccessToken system with role-based permissions
-  QR code generation and display
-  Sitter Dashboard for task management
-  Handoff creation and management UI
-  Permission checking system
-  Updated Firestore security rules for token-based access

Deliverables: Complete sharing and multi-user access system



Sprint 1 Achievements

MVP Complete ✓

Features Delivered:

- ✓ **Authentication:** Secure user management with Firebase
- ✓ **Pet Profiles:** Complete CRUD with photo upload
- ✓ **Care Plans:** Feeding schedules and medication tracking
- ✓ **Notifications:** Local reminder system with permissions
- ✓ **Sharing:** QR codes, access tokens, and public views
- ✓ **Sitter Dashboard:** Task management for shared access
- ✓ **Architecture:** Clean, scalable code structure

Metrics:



Sprint 2 Planning (Weeks 6-10)

Primary Goal

Essential UX improvements and multi-user collaboration features

Must-Have Features

- Basic onboarding improvements and welcome screen
- Bottom navigation system for better app structure
- Real-time sync for task completions between users
- Lost & Found mode with basic poster generation
- UI polish with better error handling and user feedback

Deferred to Future Sprints:

- Weight tracking system with charts
- Push notifications for task completions
- Advanced onboarding wizard
- Dark mode support
- Professional roles and offline support



Sprint 2 Progress: Week-by-Week

Week 6: Basic Onboarding ✓

- ✓ Welcome Screen (US-001) - App overview with value proposition cards
- ✓ Guided First Pet Creation (US-002) - 3-step flow with progress indicators
- ✓ Success Screen (US-003) - Celebratory completion with next steps guidance
- ✓ Smart State Management - Persistent onboarding state with fallback logic
- ✓ LocalStorageService - Persistent onboarding tracking using SharedPreferences
- ✓ AppStartup widget - Smart routing logic for new vs. returning users

Deliverables: ~1,500+ lines of code, 6 new files, complete onboarding experience

Week 7: Navigation Improvements

-  Bottom Navigation Bar (US-004, US-005) - 3-tab system: Pets, Care, Profile
-  Unified Navigation Shell - MainScaffold with Material 3 NavigationBar
-  State Persistence - Tab state saved and restored across app restarts
-  CareDashboardPage - Action chips and FAB for task management
-  ProfilePage - Centralized user profile and sign out
-  Consistent Profile Access - Top-right avatar icon on all tabs

Deliverables: ~500-700 lines of code, 7 new files, unified navigation system

Week 8: Real-time Task Completion Sync

-  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
-  User Display Names - Shows who completed tasks with batch fetching
-  Dashboard Statistics - Real-time completed/pending task counts

Deliverables: ~800-1,000 lines of code, 2 new files, 8 modified files, real-time sync system

Week 9: Lost & Found Mode

-  LostReport Domain Model - Complete data model with all required fields
-  LostReportRepository - Firestore operations for lost reports
-  LostFoundNotifier - Riverpod state management for lost/found operations
-  Poster Generation Service - Widget-to-image conversion with Firebase Storage upload
-  Poster Widget - Professional poster design with pet information
-  UI Integration - Mark as Lost/Found buttons, poster page, dashboard indicators
-  Firebase Security Rules - Firestore and Storage rules deployed
-  Comprehensive Test Suite - 240+ tests passing

Deliverables: ~1,500-2,000 lines of code, 8 new files, 6 modified files, complete lost & found system

Week 10: UI Polish & Error Handling

-  Basic UI Improvements (US-012, US-013)
 - Success/error feedback animations
 - Improved loading indicators
 - Better button states and interactions
-  Error Handling Improvements (US-014, US-015)
 - User-friendly error messages
 - Basic retry mechanisms
 - Clear guidance for common issues

Deliverables: Complete UI polish and error handling system

 Sprint 2 Progress Summary

Features Completed: 6 of 6 (100%)

-  **Feature 1:** Basic Onboarding Improvements (Week 6)
-  **Feature 2:** Bottom Navigation System (Week 7)
-  **Feature 3:** Real-time Task Completion Sync (Week 8)
-  **Feature 4:** Lost & Found Mode (Week 9)
-  **Feature 5:** Basic UI Improvements (Week 10)
-  **Feature 6:** Error Handling Improvements (Week 10)

User Stories Completed: 15 of 15 (100%)

-  US-001 through US-015 completed



Overall Project Metrics

Code Statistics

- Total Lines of Code: ~21,200+ lines
- Dart Source Files: 122 files
- Test Files: 43 files
- Automated Tests: 457 tests (100% pass rate)

Feature Breakdown

- 15 major features implemented
- 6 domain models
- 10+ use cases
- 2 sprints completed (10 weeks)



Technical Architecture

Technology Stack

- **Frontend:** Flutter 3.x (iOS, Android, Web, Desktop)
- **State Management:** Riverpod
- **Backend:** Firebase (Auth, Firestore, Storage)
- **Notifications:** flutter_local_notifications
- **Architecture:** Clean Architecture with Feature Modules

High-Level Structure

```
lib/
  └── app/          # App shell, routing, theme
  └── core/         # Shared widgets
  └── features/
    └── auth/
    └── pets/
    └── care_plans/
    └── sharing/
    └── lost_found/
    └── onboarding/
  └── services/      # Cross-cutting services
```



Design Patterns Applied

1. Factory Method Pattern

- Factory constructors for domain models (JSON/Firebase deserialization)
- Usage: Throughout domain models for object creation

2. Observer Pattern

- Streams and reactive state notify observers of changes
- Usage: Real-time data sync, state management with Riverpod

3. Strategy Pattern

- Interchangeable algorithms (e.g., Clock abstraction for testing)
- Usage: Testability and flexibility

4. Facade Pattern

- Simplified interfaces to complex subsystems
- Usage: Services like AuthService, QRCodeService, CareScheduler



SOLID Principles

Single Responsibility Principle (SRP)

- Repositories: Only data access
- Use Cases: Single business operation
- Services: Focused functionality

Open/Closed Principle (OCP)

- Abstract repository interfaces allow extension without modification

Liskov Substitution Principle (LSP)

- Repository implementations are fully substitutable

Interface Segregation Principle (ISP)

- Focused interfaces (AccessTokenRepository, TaskCompletionRepository)

Dependency Inversion Principle (DIP)

- High-level modules depend on abstractions
- Riverpod providers inject dependencies



Clean Architecture Layers

Domain Layer

- Entities: Pet, CarePlan, AccessToken, LostReport
- Repository Interfaces: Abstract contracts
- Pure business logic, no dependencies

Data Layer

- Repository Implementations: Firestore-specific code
- DTOs: Data transfer objects
- External dependencies isolated here

Presentation Layer

- Pages: UI screens



Security & Access Control

Firestore Security Rules

- Owner-only access to pets and care plans
- Token-based authorization for sitters
- Time-boxed access with automatic expiration
- Read-only public views for lost pet posters

Authentication Flow

- Firebase Auth for user identity
- AuthWrapper guards protected routes
- Role-based permission checking via PermissionService



Quality Assurance

Testing Strategy

- **Unit Tests:** Business logic, use cases, repositories
- **Widget Tests:** UI components
- **Integration Tests:** End-to-end flows

Test Coverage

- **457 tests** across all layers
- **100% pass rate**
- Tests for critical paths: authentication, pet CRUD, care plans, sharing

Code Quality

- **Consistent architecture** across features



Project Achievements

Completed Features

- User authentication & onboarding
- Pet profile management with photos
- Comprehensive care plans (diet, meds, schedules)
- Timezone-aware local notifications
- QR code & link sharing system
- Sitter dashboard with task completion
- Real-time task sync
- Lost & Found mode with poster generation
- Role-based access control
- Public profile views
- Bottom navigation system
- Guided onboarding flow



Future Enhancements

Planned Features

- **Messaging System:** Owner-sitter chat
- **Push Notifications:** FCM integration
- **Weight Tracking:** Health monitoring with charts
- **Offline Support:** Local caching with Hive
- **Dark Mode:** Complete theme implementation
- **Professional Roles:** Vets, trainers, groomers



Key Takeaways

Problem Solved

Petfolio provides a **unified platform** for pet care coordination, eliminating communication gaps and ensuring consistent care.

Technical Excellence

- **Clean Architecture** with clear separation of concerns
- **SOLID principles** applied consistently
- **Design patterns** (Repository, Use Case, Provider) for maintainability
- **Comprehensive testing** for reliability

Impact

- **Real-world applicability:** Solves actual coordination problems



Current Project Status

Sprint 1: ✓ 100% COMPLETE

- All MVP features delivered
- Core pet care functionality operational
- Sharing and access control system functional

Sprint 2: ✓ 100% COMPLETE

- 6 of 6 features complete
- 15 of 15 user stories complete
- All Sprint 2 goals achieved

Overall Progress

- 10 weeks of development completed

Milestones Reached

Sprint 1 Milestones

-  Week 2: Authentication system complete
-  Week 3: Pet CRUD with photo upload complete
-  Week 4: Care plans and notifications complete
-  Week 5: Sharing and access tokens complete
-  Sprint 1 MVP Complete

Sprint 2 Milestones

-  Week 6: Onboarding improvements complete
-  Week 7: Navigation system complete
-  Week 8: Real-time sync complete
-  Week 9: Lost & Found mode complete



Burndown Summary

Sprint 1

- **Planned:** 5 features
- **Completed:** 5 features
- **Burndown Rate:** 100%

Sprint 2

- **Planned:** 6 features
- **Completed:** 6 features
- **Burndown Rate:** 100%

Overall

- **Total Planned:** 11 features



Project Highlights

What Went Well

- **✓ Complete Authentication System** - Successfully implemented Firebase Auth
- **✓ Comprehensive Pet Management** - Built full CRUD system with photo upload
- **✓ Robust Care Plan System** - Developed scheduling with local notifications
- **✓ Clean Architecture** - Proper separation of concerns using Riverpod
- **✓ Production-Ready MVP** - Delivered all core features functional and polished
- **✓ Real-time Collaboration** - Multi-user sync working seamlessly
- **✓ Emergency Features** - Lost & Found mode with poster generation



Lessons Learned

Individual Level Insights

- **Focus on MVP First** - Defer enhancements to later sprints for better time management
- **Cross-Platform Testing** - Allocate more time for testing across different platforms
- **Error Handling** - Implement comprehensive error handling and user feedback systems
- **Time Management** - Plan better to avoid last-minute feature additions
- **Documentation** - Create detailed technical documentation for future development



Next Steps

Immediate (Post-Sprint 2)

- Conduct final testing and quality assurance
- User acceptance testing
- Performance optimization

Short-term (Post-Sprint 2)

- User testing and feedback collection
- Performance optimization
- Bug fixes and refinements

Long-term (Future Sprints)

- Weight tracking system



Success Criteria Status

Sprint 1: ✓ MET

- ✓ One owner can create a pet, add a care plan, get reminders, share QR profile
- ✓ All core MVP features functional and tested

Sprint 2: ✓ MET

- ✓ Improved user experience with onboarding and navigation
- ✓ Real-time task completion sync working
- ✓ Lost & Found mode functional
- ✓ UI polish and error handling complete

Long-term: ON TRACK

- Foundation established for future professional roles
- Scalable architecture ready for expansion



Project Vision

Current State

Petfolio provides a **unified platform** for pet care coordination with:

- Real-time synchronization
- Role-based access control
- Simple handoff mechanisms
- Emergency response features

Future Vision

- Professional ecosystem for vets, trainers, and groomers
- Health monitoring with weight tracking and analytics
- Offline support for reliable access
- Advanced notifications for critical events



Final Statistics

Development Metrics

- **Total Development Time:** 10 weeks
- **Sprint 1 Duration:** 5 weeks (100% complete)
- **Sprint 2 Duration:** 5 weeks (100% complete)
- **Total Features:** 15 major features
- **Code Quality:** Production-ready with comprehensive testing

Code Metrics

- **Lines of Code:** ~21,200+
- **Source Files:** 122 Dart files
- **Test Files:** 43 test files
- **Test Coverage:** 457 automated tests (100% pass rate)

🏁 Conclusion

Project Status: **COMPLETE** ✓

Petfolio has successfully delivered:

- ✓ Complete MVP with all core features
- ✓ Real-time collaboration for multi-user scenarios
- ✓ Emergency features for lost pet management
- ✓ Enhanced UX with onboarding, navigation, and UI polish
- ✓ Production-ready codebase with comprehensive testing
- ✓ Scalable architecture for future growth
- ✓ All Sprint 1 and Sprint 2 goals achieved

Ready for: User testing, deployment, and future enhancements



Thank you!