

PawCare: 3-Week Development Sprint Roadmap

Team Retrievers: Gaurav, Shivam, Ayush
Project: Kalvium Sprint #2 - PawCare Pet Discovery & Monitoring App
Technology Stack: Flutter + Firebase
Sprint Duration: 3 Weeks (21 Days)
Sprint Start Date: February 6, 2026

Sprint Overview

This roadmap divides the PawCare MVP development into three focused weeks with clear daily objectives, parallel workstreams, and defined responsibilities for each team member. The plan balances individual ownership with collaborative integration points to ensure smooth delivery.

Team Member Specializations

Member	Focus Area	Key Responsibilities
Gaurav	Authentication & Profiles	Firebase Auth setup, User/Caregiver profiles, Pet management, AuthProvider, Document uploads
Shivam	Discovery & Booking	Search/filters, Firestore queries, Booking system, CaregiverProvider, BookingProvider, Chat
Ayush	Monitoring & Reviews	Real-time activity feed, Photo uploads, StreamBuilder implementation, Reviews, Notifications

Table 1: Team Member Specializations

Week 1: Foundation & Core Infrastructure

Goal: Set up project foundation, implement authentication system, and create basic user profiles.

Daily Breakdown

Day 1 (Feb 6): Project Setup & Planning

All Team Members (Morning - 2 hours):

- Joint kickoff meeting to review project scope
- Set up GitHub repository with proper branch structure
- Create Flutter project with Firebase configuration
- Install required packages (firebase_core, firebase_auth, cloud_firestore, provider)
- Set up project folder structure (models, providers, repositories, screens, widgets)
- Define coding standards and naming conventions

Gaurav (Afternoon - 3 hours):

- Initialize Firebase project in console
- Add Firebase to Flutter (Android & iOS configuration)
- Enable Firebase Authentication (Email/Password, Google Sign-In)
- Create Firestore database in test mode
- Set up Firebase Storage buckets
- Test Firebase connection with simple read/write

Shivam (Afternoon - 3 hours):

- Design Firestore database schema (collections structure)
- Create data models (User, Caregiver, Pet, Booking)
- Write model classes with fromFirestore() and toMap() methods
- Document database relationships and indexes needed

Ayush (Afternoon - 3 hours):

- Set up app theme with design system colors
- Create constants file (colors, text styles, spacing)
- Design reusable widget components (CustomButton, CustomTextField)
- Create navigation structure (routes and route names)
- Build splash screen with app logo

Day 1 Deliverables:

- ✓ Flutter project with Firebase configured
- ✓ Database schema documented
- ✓ Basic project structure and navigation
- ✓ Splash screen implemented

Day 2 (Feb 7): Authentication System

Gaurav (Full Day - 6 hours):

- Create AuthRepository with sign-up, login, logout methods
- Implement AuthProvider using ChangeNotifier
- Add error handling for authentication failures
- Implement auth state persistence
- Write unit tests for AuthRepository

- Create Welcome screen with role selection (Owner/Caregiver)

Shivam (Full Day - 6 hours):

- Build Login screen UI with email/password fields
- Build Sign Up screen UI with validation
- Implement form validators (email format, password strength)
- Add "Forgot Password" functionality
- Create loading states and error messages
- Test authentication flows end-to-end

Ayush (Full Day - 6 hours):

- Implement Google Sign-In integration
- Create OTP verification screen UI
- Integrate phone authentication with Firebase
- Build authentication wrapper to handle route navigation
- Create custom error dialog widgets
- Add biometric authentication setup (future-ready)

Day 2 Deliverables:

- ✓ Complete authentication system (email, Google, phone OTP)
- ✓ AuthProvider with state management
- ✓ Login, Sign Up, Welcome screens functional
- ✓ Auth state persistence working

Integration Point: Team sync to test complete auth flow together (30 mins end of day)

Day 3 (Feb 8): User Profile Setup (Owner)

Gaurav (Full Day - 6 hours):

- Create UserRepository for Firestore user operations
- Build Owner Profile Setup screen (name, location, phone)
- Implement image picker for profile photo
- Upload profile photos to Firebase Storage
- Create UserProvider for profile state management
- Add location picker widget (manual address entry for MVP)

Shivam (Full Day - 6 hours):

- Create Pet model with all fields
- Build Pet Profile Form screen (name, breed, age, photo)
- Implement PetRepository for CRUD operations
- Create PetProvider for state management
- Build "My Pets" list screen showing all pets
- Add edit and delete pet functionality

Ayush (Full Day - 6 hours):

- Design and build Owner Home Dashboard UI
- Create search bar component

- Build quick action buttons (Find Walker, My Bookings, Profile)
- Add bottom navigation bar
- Create empty state placeholders
- Implement pull-to-refresh functionality

Day 3 Deliverables:

- ✓ Owner profile creation flow complete
- ✓ Pet profile management working
- ✓ Owner home dashboard UI ready
- ✓ Image upload to Firebase Storage functional

Day 4 (Feb 9): Caregiver Profile Setup

Gaurav (Full Day - 6 hours):

- Create CaregiverRepository for profile operations
- Build Caregiver Profile Setup screen (bio, experience, rate)
- Add service type selection (walking, daycare, training)
- Implement pet type specialization checkboxes
- Create document upload for ID verification (Aadhaar/PAN)
- Store verification documents in Firebase Storage

Shivam (Full Day - 6 hours):

- Build Availability Setup screen
- Create time slot selector for each day of week
- Implement service area selection (neighborhood picker)
- Add certifications upload section
- Create CaregiverProvider for state management
- Build caregiver verification pending screen

Ayush (Full Day - 6 hours):

- Design Caregiver Dashboard UI
- Create booking request cards
- Build earnings summary widget
- Add upcoming sessions calendar view
- Create caregiver stats display (total bookings, rating)
- Implement navigation between caregiver screens

Day 4 Deliverables:

- ✓ Complete caregiver onboarding flow
- ✓ Document verification upload working
- ✓ Availability and service area setup functional
- ✓ Caregiver dashboard UI ready

Integration Point: Full team review of onboarding flows (1 hour end of day)

Day 5 (Feb 10): Firestore Security & Repository Pattern

Gaurav (Full Day - 6 hours):

- Write Firestore security rules for users collection
- Add security rules for caregivers collection
- Test security rules with different user roles
- Create security rule for pet ownership validation
- Document security rules with comments
- Set up Firestore indexes required for queries

Shivam (Full Day - 6 hours):

- Refactor all repositories to follow consistent pattern
- Add error handling wrapper for all Firestore operations
- Implement retry logic for failed operations
- Create repository base class for common methods
- Add logging for debugging Firestore operations
- Write documentation for repository usage

Ayush (Full Day - 6 hours):

- Create NotificationService class
- Set up Firebase Cloud Messaging
- Implement local notification handling
- Add notification permissions request
- Create notification preferences screen
- Test notifications on both Android and iOS

Day 5 Deliverables:

- ✓ Firestore security rules implemented and tested
- ✓ Repository pattern standardized across codebase
- ✓ Push notifications infrastructure ready
- ✓ All Week 1 features integrated and tested

Week 1 Checkpoint: Team demo of authentication and profile management (1 hour)

Week 2: Discovery, Booking & Communication

Goal: Implement caregiver discovery, booking system, and basic communication features.

Daily Breakdown

Day 6 (Feb 11): Caregiver Discovery UI

Gaurav (Full Day - 6 hours):

- Create CaregiverCard widget with profile photo, rating, and details
- Build caregiver profile detail screen
- Add image carousel for caregiver photos
- Display certifications and verification badge
- Show availability calendar on profile

- Add "Contact" and "Book Now" buttons

Shivam (Full Day - 6 hours):

- Build Search Results screen with ListView
- Implement search bar with text filtering
- Create filter bottom sheet (location, rating, price)
- Add sort options (rating, price, distance)
- Implement pagination for caregiver list
- Add empty state when no results found

Ayush (Full Day - 6 hours):

- Design filter UI with chips and sliders
- Create rating filter (4+ stars, 3+ stars)
- Build price range slider
- Add pet type filter checkboxes
- Implement "Apply Filters" logic
- Create filter badge counter on search screen

Day 6 Deliverables:

- ✓ Caregiver discovery UI complete
- ✓ Search and filter functionality working
- ✓ Caregiver profile detail screen functional

Day 7 (Feb 12): Firestore Queries & Search Logic

Gaurav (Full Day - 6 hours):

- Update CaregiverProvider with fetch methods
- Implement Firestore query for verified caregivers
- Add location-based filtering using GeoPoint
- Create composite index for multiple filters
- Implement search by name functionality
- Add caching to reduce Firestore reads

Shivam (Full Day - 6 hours):

- Implement rating filter query
- Add price range filtering logic
- Create availability checking algorithm
- Build pet type specialization filter
- Add distance calculation for location filtering
- Test all filter combinations

Ayush (Full Day - 6 hours):

- Implement search debouncing for text input
- Add search history feature
- Create featured caregivers algorithm (high rating + verified)
- Build "Recently Viewed" caregivers list

- Add loading skeletons for better UX
- Test search performance with large dataset

Day 7 Deliverables:

- ✓ All search filters working with Firestore
- ✓ Location-based discovery functional
- ✓ Search performance optimized
- ✓ Firestore indexes created and deployed

Integration Point: Test complete discovery flow (30 mins)

Day 8 (Feb 13): Booking System - Part 1

Gaurav (Full Day - 6 hours):

- Create Booking model with all fields
- Build BookingRepository with CRUD methods
- Implement Firestore transaction for booking creation
- Add booking status state machine (pending, confirmed, etc.)
- Create BookingProvider for state management
- Write business logic for availability checking

Shivam (Full Day - 6 hours):

- Design Booking Form screen UI
- Add service type selector (walking, daycare, overnight)
- Create date picker for booking date
- Build time slot selector based on caregiver availability
- Add duration selector (1 hour, 2 hours, etc.)
- Implement special instructions text field

Ayush (Full Day - 6 hours):

- Build Booking Confirmation screen
- Display booking summary (date, time, pet, price)
- Create price calculation logic (hourly rate × duration)
- Add terms and conditions checkbox
- Implement "Confirm Booking" button logic
- Create booking success screen with confirmation details

Day 8 Deliverables:

- ✓ Booking model and repository complete
 - ✓ Booking form UI functional
 - ✓ Price calculation working
 - ✓ Booking creation flow end-to-end
-

Day 9 (Feb 14): Booking System - Part 2

Gaurav (Full Day - 6 hours):

- Build "My Bookings" screen for owners
- Create tabs for Upcoming, In-Progress, Completed, Cancelled
- Implement BookingCard widget with status badges
- Add booking detail screen
- Create cancel booking functionality with confirmation dialog
- Implement booking modification (reschedule)

Shivam (Full Day - 6 hours):

- Build "My Schedule" screen for caregivers
- Create calendar view with booking highlights
- Implement booking request notification
- Add accept/decline booking functionality
- Create booking detail screen for caregivers
- Build earnings calculation for each booking

Ayush (Full Day - 6 hours):

- Implement booking status update logic
- Create notification triggers for booking events
- Add push notification when booking is confirmed
- Build notification when booking is approaching
- Create reminder notification 1 hour before booking
- Test all booking notifications

Day 9 Deliverables:

- ✓ Complete booking management for owners
- ✓ Caregiver booking acceptance flow working
- ✓ Booking notifications implemented
- ✓ My Bookings and My Schedule screens functional

Integration Point: Full team testing of booking flow (1 hour)

Day 10 (Feb 15): In-App Chat - Part 1

Gaurav (Full Day - 6 hours):

- Create Chat and Message models
- Build ChatRepository with Firestore operations
- Implement chat creation between owner and caregiver
- Add message sending functionality
- Create ChatProvider for state management
- Set up real-time message listeners

Shivam (Full Day - 6 hours):

- Design Chat List screen showing all conversations
- Build chat preview cards with last message

- Display unread message count badges
- Add timestamp formatting (Just now, Yesterday, etc.)
- Implement pull-to-refresh for chat list
- Create empty state when no chats exist

Ayush (Full Day - 6 hours):

- Build Chat Screen UI with message bubbles
- Create message input field with send button
- Implement message grouping by date
- Add scroll-to-bottom button
- Create typing indicator (optional)
- Style sender vs receiver message bubbles differently

Day 10 Deliverables:

- ✓ Chat infrastructure with Firestore working
- ✓ Chat list screen functional
- ✓ Basic messaging UI complete
- ✓ Real-time message updates working

Day 11 (Feb 16): In-App Chat - Part 2 & Integration

Gaurav (Full Day - 6 hours):

- Implement read receipts for messages
- Add message timestamp display
- Create "Contact Caregiver" button on booking detail
- Auto-create chat when booking is confirmed
- Add Firestore security rules for chats
- Test chat permissions and access control

Shivam (Full Day - 6 hours):

- Implement push notifications for new messages
- Add notification when user receives message
- Create in-app notification banner
- Build "Mark all as read" functionality
- Add search in chat messages
- Test chat with multiple users

Ayush (Full Day - 6 hours):

- Add image sharing in chat (optional MVP feature)
- Implement message deletion
- Create report/block user functionality
- Add chat archived feature
- Build chat settings screen
- Polish chat UI with animations

Day 11 Deliverables:

- ✓ Complete in-app messaging system

- ✓ Chat notifications working
- ✓ Chat integrated with booking flow
- ✓ All Week 2 features tested and integrated

Week 2 Checkpoint: Team demo of discovery, booking, and chat (1 hour)

Week 3: Real-Time Monitoring, Reviews & Polish

Goal: Implement activity monitoring, reviews system, and polish the entire app for presentation.

Daily Breakdown

Day 12 (Feb 17): Activity Monitoring - Infrastructure

Gaurav (Full Day - 6 hours):

- Create Activity model with all fields
- Build ActivityRepository with Firestore subcollection
- Implement activity creation method
- Add photo upload to Firebase Storage
- Create ActivityProvider for state management
- Set up real-time listeners for activity updates

Shivam (Full Day - 6 hours):

- Build "Start Session" button on booking detail
- Create session state management (not started, active, completed)
- Implement session start logic (update booking status)
- Add session timer display
- Create "End Session" functionality
- Build session summary screen with total duration

Ayush (Full Day - 6 hours):

- Design Activity Feed UI for owners
- Create activity card widgets (photo, text, timestamp)
- Implement StreamBuilder for real-time updates
- Add photo viewer with zoom capability
- Create activity timeline visualization
- Build empty state for no activities yet

Day 12 Deliverables:

- ✓ Activity monitoring infrastructure ready
 - ✓ Session start/end functionality working
 - ✓ Real-time activity feed displaying updates
-

Day 13 (Feb 18): Activity Monitoring - Caregiver Features

Gaurav (Full Day - 6 hours):

- Build "Add Update" screen for caregivers
- Create photo picker with camera and gallery options
- Implement image compression before upload
- Add text message input field
- Create activity type selector (playing, resting, eating, walking)
- Build "Post Update" button with validation

Shivam (Full Day - 6 hours):

- Implement photo upload progress indicator
- Add multiple photo selection (up to 5 photos)
- Create photo preview before posting
- Build "Quick Updates" with predefined messages
- Add emoji selector for update messages
- Test activity creation with different media types

Ayush (Full Day - 6 hours):

- Implement push notifications for activity updates
- Send notification to owner when caregiver posts update
- Create rich notification with photo thumbnail
- Add notification action to view activity directly
- Build activity notification settings
- Test notifications on both platforms

Day 13 Deliverables:

- ✓ Caregiver activity update creation working
- ✓ Photo upload with compression functional
- ✓ Owner receives notifications for updates
- ✓ Complete activity monitoring flow tested

Integration Point: Test full session flow from start to end (30 mins)

Day 14 (Feb 19): Reviews & Ratings System

Gaurav (Full Day - 6 hours):

- Create Review model with all fields
- Build ReviewRepository with Firestore operations
- Implement review creation logic
- Add review submission validation (only after completed booking)
- Create ReviewProvider for state management
- Write Cloud Function to update caregiver average rating

Shivam (Full Day - 6 hours):

- Build Review Screen UI
- Create star rating selector (1-5 stars)

- Add text review input field
- Implement photo upload with review (optional)
- Create review submission button
- Build review success confirmation

Ayush (Full Day - 6 hours):

- Display reviews on caregiver profile
- Create review card widget with star rating
- Add reviewer name and date
- Implement review pagination (load more)
- Build rating breakdown chart (5-star: X, 4-star: Y, etc.)
- Add filter to show only verified reviews

Day 14 Deliverables:

- ✓ Complete reviews and ratings system
 - ✓ Review submission working for owners
 - ✓ Reviews displaying on caregiver profiles
 - ✓ Average rating calculation functional
-

Day 15 (Feb 20): Dashboard Widgets & Statistics

Gaurav (Full Day - 6 hours):

- Create owner dashboard widgets (total bookings, favorite caregivers)
- Build pet activity summary widget
- Add upcoming bookings widget on home screen
- Create quick action shortcuts
- Implement "Featured Caregivers" section
- Add recent activity feed preview

Shivam (Full Day - 6 hours):

- Build caregiver dashboard statistics
- Create earnings summary widget (today, week, month)
- Add booking stats (total, completed, cancelled)
- Display average rating prominently
- Create chart for earnings trend
- Build recent reviews widget

Ayush (Full Day - 6 hours):

- Implement analytics tracking for key events
- Add Firestore queries for statistics calculations
- Create data aggregation for dashboard widgets
- Build refresh mechanism for statistics
- Add loading states for dashboard data
- Test performance with statistics calculations

Day 15 Deliverables:

- ✓ Owner and caregiver dashboards fully functional

- ✓ Statistics and analytics displaying correctly
 - ✓ Dashboard widgets responsive and performant
-

Day 16 (Feb 21): Settings & Profile Management

Gaurav (Full Day - 6 hours):

- Build Settings screen with all options
- Create Edit Profile functionality
- Add change password feature
- Implement phone number update with verification
- Create account deletion with confirmation
- Build privacy settings screen

Shivam (Full Day - 6 hours):

- Create notification preferences screen
- Add toggle for push notifications
- Build email notification settings
- Create language selection (future-ready)
- Add app version and about screen
- Implement terms and conditions screen

Ayush (Full Day - 6 hours):

- Build help and support screen
- Create FAQ section
- Add contact support form
- Implement feedback submission
- Create app tutorial/onboarding (optional)
- Build logout confirmation dialog

Day 16 Deliverables:

- ✓ Complete settings and preferences
 - ✓ Profile editing functional
 - ✓ Help and support screens ready
-

Day 17 (Feb 22): UI/UX Polish & Bug Fixes

All Team Members (Morning - 2 hours):

- Team meeting to review all features
- Create bug list and prioritize fixes
- Divide bug fixes among team members
- Set polish priorities

Gaurav (Afternoon - 4 hours):

- Fix authentication-related bugs
- Polish profile screens UI
- Add loading states where missing
- Improve error messages

- Test edge cases in profile flows
- Optimize image loading and caching

Shivam (Afternoon - 4 hours):

- Fix booking and search bugs
- Polish booking form validation
- Improve filter UI responsiveness
- Add confirmation dialogs for critical actions
- Test booking edge cases
- Optimize Firestore query performance

Ayush (Afternoon - 4 hours):

- Fix activity monitoring bugs
- Polish real-time feed UI
- Improve notification reliability
- Add skeleton loaders for better UX
- Test notification delivery
- Optimize image uploads

Day 17 Deliverables:

- ✓ Major bugs fixed
- ✓ UI/UX improvements implemented
- ✓ All features tested for edge cases

Day 18 (Feb 23): Testing & Quality Assurance

Gaurav (Full Day - 6 hours):

- Conduct end-to-end testing for owner flow
- Test authentication on multiple devices
- Verify profile creation and editing
- Test pet management functionality
- Check Firebase Storage uploads
- Document any remaining issues

Shivam (Full Day - 6 hours):

- Conduct end-to-end testing for caregiver flow
- Test booking creation and acceptance
- Verify search and filters accuracy
- Test chat functionality thoroughly
- Check booking notifications
- Create test cases documentation

Ayush (Full Day - 6 hours):

- Test activity monitoring completely
- Verify real-time updates work reliably
- Test reviews and ratings system

- Check all notifications delivery
- Test app on different screen sizes
- Create known issues list

Day 18 Deliverables:

- ✓ Complete end-to-end testing done
- ✓ Test cases documented
- ✓ Critical bugs fixed
- ✓ App stable for demo

Integration Point: Full team testing session (2 hours)

Day 19 (Feb 24): Presentation Preparation - Part 1

Gaurav (Full Day - 6 hours):

- Create PPT structure and outline
- Design slides for problem statement and solution
- Add user personas slides
- Create technical architecture diagram slides
- Prepare authentication flow screenshots
- Write talking points for introduction

Shivam (Full Day - 6 hours):

- Create slides for core features
- Add booking system flow slides
- Design database structure slides
- Prepare search and discovery screenshots
- Create state management explanation slides
- Write feature demonstration script

Ayush (Full Day - 6 hours):

- Create slides for real-time monitoring
- Add activity feed demonstration slides
- Design future scope slides
- Prepare reviews system screenshots
- Create conclusion and learning outcomes slides
- Write demo script for presentation

Day 19 Deliverables:

- ✓ PPT structure complete
 - ✓ All slides drafted
 - ✓ Screenshots and diagrams ready
 - ✓ Demo script prepared
-

Day 20 (Feb 25): Presentation Preparation - Part 2

All Team Members (Morning - 2 hours):

- Review PPT together
- Refine slides based on team feedback
- Finalize slide order and transitions
- Ensure consistent design across slides

Gaurav (Afternoon - 4 hours):

- Practice demo of owner flow
- Prepare answers for authentication questions
- Record backup demo video for owner journey
- Polish PPT design and formatting
- Create handout document (summary)

Shivam (Afternoon - 4 hours):

- Practice demo of caregiver flow
- Prepare answers for database and booking questions
- Record backup demo video for caregiver journey
- Create technical FAQ document
- Prepare code snippets for viva

Ayush (Afternoon - 4 hours):

- Practice demo of monitoring features
- Prepare answers for real-time and notification questions
- Record backup demo video for activity monitoring
- Create demo device setup checklist
- Test presentation flow timing

Day 20 Deliverables:

- ✓ Complete PPT finalized
- ✓ Demo videos recorded
- ✓ Viva preparation complete
- ✓ Backup plans ready

Integration Point: Full team practice presentation (1 hour)

Day 21 (Feb 26): Final Rehearsal & Documentation

All Team Members (Morning - 3 hours):

- Full dress rehearsal of presentation
- Time the presentation (target: 15-20 mins)
- Practice Q&A session
- Refine transitions between team members
- Test demo on presentation device
- Check internet connectivity and Firebase access

Gaurav (Afternoon - 3 hours):

- Finalize project README on GitHub
- Create installation instructions
- Add architecture documentation
- Write contributor guidelines
- Prepare project submission files
- Create final APK build

Shivam (Afternoon - 3 hours):

- Update code documentation
- Add comments to complex logic
- Create API reference document
- Document Firebase setup steps
- Prepare database schema document
- Create future roadmap document

Ayush (Afternoon - 3 hours):

- Create user guide document
- Write troubleshooting guide
- Document known limitations
- Prepare video demo upload
- Create GitHub Pages documentation site
- Final quality check of all deliverables

Day 21 Deliverables:

- ✓ Presentation ready and rehearsed
- ✓ Complete documentation finished
- ✓ Project submitted
- ✓ Team ready for demo and viva

Daily Standup Structure

Every day at 9:30 AM (15 minutes):

1. **Yesterday's Progress:** Each member shares completed tasks
2. **Today's Plan:** Each member outlines daily goals
3. **Blockers:** Discuss any obstacles or dependencies
4. **Integration Needs:** Identify areas requiring team coordination

Every day at 6:00 PM (15 minutes):

1. **Day Review:** What got done, what's pending
 2. **Code Review:** Quick review of pushed code
 3. **Tomorrow's Prep:** What each member needs to start tomorrow
 4. **Help Needed:** Any support required from team members
-

Git Workflow & Collaboration

Branch Strategy

- **main:** Production-ready code only
- **develop:** Integration branch for all features
- **feature/auth-system** (Gaurav's branch)
- **feature/booking-system** (Shivam's branch)
- **feature/activity-monitoring** (Ayush's branch)

Daily Git Routine

Start of Day:

```
git checkout develop  
git pull origin develop  
git checkout -b feature/your-feature-name
```

End of Day:

```
git add .  
git commit -m "feat: descriptive commit message"  
git push origin feature/your-feature-name
```

Create Pull Request for team review

Integration Points

Critical merge days where team must coordinate:

- **Day 5 End:** Merge all Week 1 features to develop
- **Day 11 End:** Merge all Week 2 features to develop
- **Day 18 End:** Final merge to main for submission

Weekly Milestones & Success Criteria

Week 1 Success Criteria

Milestone	Status
Users can sign up and log in with email and Google	
Pet owners can create profile and add pets	
Caregivers can create profile with verification documents	
Firebase Auth, Firestore, Storage working correctly	
Navigation between screens smooth	
No critical bugs in authentication flow	

Table 2: Week 1 Success Criteria Checklist

Week 2 Success Criteria

Milestone	Status
Owners can search and filter caregivers	
Owners can view caregiver profiles with details	
Booking creation and confirmation working	
Caregivers can accept/decline bookings	
In-app chat between owner and caregiver functional	
All Firestore queries optimized with indexes	

Table 3: Week 2 Success Criteria Checklist

Week 3 Success Criteria

Milestone	Status
Caregivers can post activity updates during sessions	
Owners see real-time activity feed	
Reviews and ratings system complete	
Push notifications working reliably	
All screens polished and bug-free	
Presentation and documentation ready	

Table 4: Week 3 Success Criteria Checklist

Risk Management

Potential Risks & Mitigation

Risk	Impact	Mitigation Strategy
Firebase quota exceeded	High - App stops working	Monitor usage daily, implement caching, use pagination
Merge conflicts	Medium - Delays integration	Daily standups, clear code ownership, frequent small merges
Feature complexity underestimated	High - Miss deadlines	Buffer time on Day 17-18, prioritize MVP features
Team member unavailable	Medium - Work redistribution	Cross-training, documentation, flexible task reassignment
Real-time updates not working	High - Core feature broken	Test early (Day 12), have fallback polling mechanism
Presentation device issues	Medium - Demo fails	Backup video demos, test setup day before

Table 5: Risk Management Matrix

Communication Channels

Channel	Purpose	Response Time
WhatsApp Group	Quick questions, daily updates, blockers	Within 1 hour during work hours
GitHub Issues	Bug tracking, feature discussions	Check twice daily
Google Meet	Standups, pair programming, reviews	Scheduled meetings
Shared Google Doc	Design decisions, meeting notes	Async, no immediate response needed

Table 6: Communication Guidelines

Tools & Resources Needed

Development Tools

- **IDE:** Android Studio or VS Code with Flutter extensions
- **Version Control:** GitHub with repository access for all members
- **Testing:** Physical Android/iOS devices (emulators are insufficient for Firebase)
- **Design:** Figma or Adobe XD for UI mockups (optional)
- **Firebase Console:** Admin access for all team members

Documentation Tools

- **Presentation:** Google Slides or PowerPoint
- **Documentation:** Google Docs or Markdown files
- **Diagrams:** [Draw.io](https://draw.io) or Lucidchart for architecture diagrams
- **Screen Recording:** OBS Studio or QuickTime for demo videos

Learning Resources

- Flutter Documentation: <https://docs.flutter.dev>
- Firebase Docs: <https://firebase.flutter.dev>
- Provider Package: <https://pub.dev/packages/provider>
- Kalvium Learning Platform: Course materials and examples

Code Review Guidelines

Before Merging Any Pull Request:

1. **Code Quality:** Follows Dart style guide, proper naming conventions
2. **Functionality:** Feature works as expected, no breaking changes
3. **Testing:** Manually tested on at least one device

4. **Documentation:** Complex logic has comments, functions documented
5. **No Secrets:** No API keys or credentials in code
6. **Performance:** No obvious performance issues

Reviewer Checklist:

- ☐ Code builds without errors
 - ☐ Feature matches requirements
 - ☐ No obvious bugs or edge cases missed
 - ☐ Code is readable and maintainable
 - ☐ Proper error handling present
 - ☐ Approved to merge
-

Success Metrics

Technical Metrics

- **App Stability:** Zero crashes during demo and viva
- **Feature Completeness:** All MVP features functional
- **Code Quality:** Clean, documented, following best practices
- **Performance:** Smooth UI, fast loading times
- **Firebase Integration:** All services working reliably

Presentation Metrics

- **Clarity:** Able to explain technical concepts simply
- **Confidence:** Team members know their areas deeply
- **Demo:** Smooth execution without major hiccups
- **Q&A:** Prepared answers for expected questions
- **Timing:** Presentation within allocated time

Team Collaboration Metrics

- **Attendance:** All members present at standups
 - **Commits:** Regular commits from all members
 - **Communication:** Active participation in discussions
 - **Support:** Team members helping each other
 - **Integration:** Smooth merging of features
-

Emergency Contacts & Escalation

If Critical Blocker Occurs:

1. Post immediately in WhatsApp group with "BLOCKER:" prefix
2. Schedule emergency huddle within 30 minutes
3. Evaluate impact on timeline
4. Redistribute work if needed
5. Document decision and workaround

Technical Support:

- Kalvium Mentors: For conceptual doubts
 - Stack Overflow: For technical issues
 - Firebase Support: For platform-specific problems
 - Team Senior/Alumni: For architecture advice
-

Celebration & Retrospective

After Day 21 Presentation:

1. **Team Celebration:** Acknowledge everyone's hard work
 2. **Retrospective Meeting:** What went well, what could improve
 3. **Lessons Learned:** Document for future projects
 4. **Showcase Work:** Post on LinkedIn, add to portfolios
 5. **Thank Mentors:** Acknowledge guidance received
-

Conclusion

This 3-week sprint roadmap provides a clear path from project setup to final presentation. Success depends on:

- **Daily Discipline:** Consistent work, regular standups, timely commits
- **Clear Ownership:** Each member owns their domain but supports others
- **Communication:** Proactive updates, asking for help early
- **Quality Focus:** Not just completing features, but doing them well
- **Team Spirit:** Supporting each other through challenges

Team Retrievers - Let's Build Something Amazing! 🚀

Appendix: Quick Reference Cards

Gaurav's Quick Reference

Primary Focus: Authentication & Profile Management

Key Files:

- lib/providers/auth_provider.dart
- lib/repositories/auth_repository.dart
- lib/repositories/user_repository.dart
- lib/screens/auth/
- lib/screens/profile/

Firebase Services:

- Firebase Auth
- Firebase Storage (profile photos, documents)
- Firestore (users, caregivers collections)

Integration Points with Team:

- Day 4: Coordinate caregiver profile structure with Shivam

- Day 7: Provide AuthProvider to Shivam for booking auth checks
- Day 16: Integrate settings with Ayush's notification preferences

Shivam's Quick Reference

Primary Focus: Discovery, Booking & Communication

Key Files:

- lib/providers/caregiver_provider.dart
- lib/providers/booking_provider.dart
- lib/repositories/booking_repository.dart
- lib/repositories/chat_repository.dart
- lib/screens/search/
- lib/screens/booking/

Firebase Services:

- Firestore (complex queries, indexes)
- Bookings collection
- Chats collection

Integration Points with Team:

- Day 8: Get user authentication state from Gaurav's AuthProvider
- Day 10: Coordinate chat UI with Ayush's notification system
- Day 14: Provide booking completion data to Ayush for review triggers

Ayush's Quick Reference

Primary Focus: Real-Time Monitoring, Reviews & Notifications

Key Files:

- lib/providers/activity_provider.dart
- lib/providers/review_provider.dart
- lib/services/notification_service.dart
- lib/screens/activity/
- lib/screens/review/

Firebase Services:

- Firestore (real-time listeners, StreamBuilder)
- Firebase Storage (activity photos)
- Firebase Cloud Messaging (push notifications)

Integration Points with Team:

- Day 5: Set up FCM tokens in Gaurav's user profiles
 - Day 12: Get active booking state from Shivam's BookingProvider
 - Day 14: Trigger rating updates after reviews posted
-

Appendix: Testing Checklist

Manual Testing Checklist (Day 18)

Authentication Flow:

- ☐ Sign up with email creates user document
- ☐ Google Sign-In works on first attempt
- ☐ Phone OTP verification completes
- ☐ Password reset email received
- ☐ Logout clears session properly
- ☐ App remembers logged-in state after restart

Profile Management:

- ☐ Owner can create and edit profile
- ☐ Pet profiles save with photos
- ☐ Caregiver profile shows all fields
- ☐ Document upload completes successfully
- ☐ Profile photos display correctly

Discovery & Search:

- ☐ Search returns relevant caregivers
- ☐ All filters work individually
- ☐ Multiple filters work together
- ☐ Location filtering accurate
- ☐ Empty state shows when no results

Booking System:

- ☐ Booking form validates inputs
- ☐ Price calculates correctly
- ☐ Booking creates in Firestore
- ☐ Caregiver receives notification
- ☐ Booking appears in both users' lists
- ☐ Cancel booking works properly

Chat:

- ☐ Messages send in real-time
- ☐ Unread count updates
- ☐ Chat history loads correctly
- ☐ Notifications work for new messages

Activity Monitoring:

- ☐ Session starts and updates booking status
- ☐ Activity updates appear instantly for owner
- ☐ Photos upload and display
- ☐ Session end time calculates correctly
- ☐ Owner receives update notifications

Reviews:

- ☐ Review can only be left after completed booking
- ☐ Star rating saves correctly
- ☐ Review appears on caregiver profile
- ☐ Average rating updates

Ready to build? Let's do this, Team Retrievers! 🐾