Sprint 2 Submission - PetAlert (Team Sunrise Inc)

GitHub Repository: https://github.com/SunriseInc-Capstone/PetAlert

Team Information and Introduction

Team Name

Sunrise Inc

Project Name

PetAlert

Team Members

Ojaswi Subedi, Tanvi Biswal, Kimberley Juarez, Anjali Fernando, Bibek Pandey

Overview

Problem Context

Pet owners face stress and delays when a pet goes missing. Information about pets and contacts is often scattered, causing slow response times.

Goal

PetAlert provides a local-first mobile app that stores pet info and generates alerts (social post, email, SMS, phone script, flyer) with one tap.

Requirements

Functional Requirements (User Stories)

- R1 Pet Profile: Create/update pet profiles with photo and details (local storage).
- R2 Emergency Contacts: Add/edit vet, shelter, police contacts with one-tap call/email/SMS.

- R3 Missing Pet Alert: Generate social media post, email/SMS draft, and phone script instantly.
- R4 Regular Update Prompts: Local notifications reminding owners to update info/photos.
- R5 Photo Gallery: Upload/view multiple pet photos locally.
- R6 Recommendations: Static safety tips and search strategies.
- R7 Toxic Foods Infographics: Quick visual guides for dangerous foods.

Non-Functional Requirements

- NF1 Security: All data stored locally; optional OS-level lock (FaceID/Passcode).
- NF2 Reliability: Core features work offline (profiles, contacts, alerts).
- NF3 Usability: ≤3 taps to reach any function; emergency-first button on home.
- NF4 Accessibility: Support large text, high contrast, screen reader labels.
- NF5 Cross-Platform: Runs consistently on iOS 14+ and Android 10+.
- NF6 Accuracy: Generated alerts match stored details exactly.

Design

The app uses a tab-based navigation system:

- Home: Big Missing Pet Alert button
- Pets: List of pets, access to detail screen
- Pet Detail: Shows info + generate alert option
- Contacts: Emergency contacts with quick actions
- Toxic Foods: Static infographic cards

Reflection Report

Accomplishments

In this sprint, we dedicated our time to research what platforms we should use to code our app. We had different platforms in mind, such as Kivy, BeeWare, React Native, and Expo, but

ultimately, we went with Flutter as it met almost every requirement we had in mind: Python for a backend, easy for beginners, good tools. We decided to compromise with having to learn Dart for our frontend with Flutter, but it was much better than learning two new languages instead of just one, which we would have to do with the other platform options.

What Went Well

Team collaboration and organization was strong. Each member participated in refining documentation, attending team meetings, and doing their assigned tasks. The workflow was smooth and everyone was cooperative while reviewing and editing.

What Could Be Improved

Although time management has been better compared to the previous sprints, there's still room for improvement. Starting at least a week before the end of a sprint or even at the start of a new sprint is the ideal and goal for our team's improvement in this area.

Plans for Sprint 3

In Sprint 3, we plan to evenly divide the tasks among team members, complete any unfinished features from Sprint 2, and develop additional new features.

Rubric Reflection

What went well in the presentation?

Every team member got to talk about their research and completed tasks in this sprint, and the team had a small demo ready for the app.

What can be improved in the presentation?

There was not enough substantial material to talk about to the professor. Showing off what the team has accomplished, including the smaller details, would be a good way to have more to talk about.

Other Notes

...