COLLEGE OF ENGINEERING

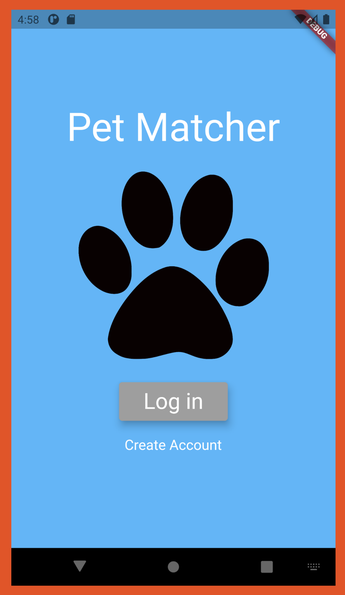
Electrical Engineering and Computer Science

PET MATCHER, CROSS-PLATFORM MOBILE APP

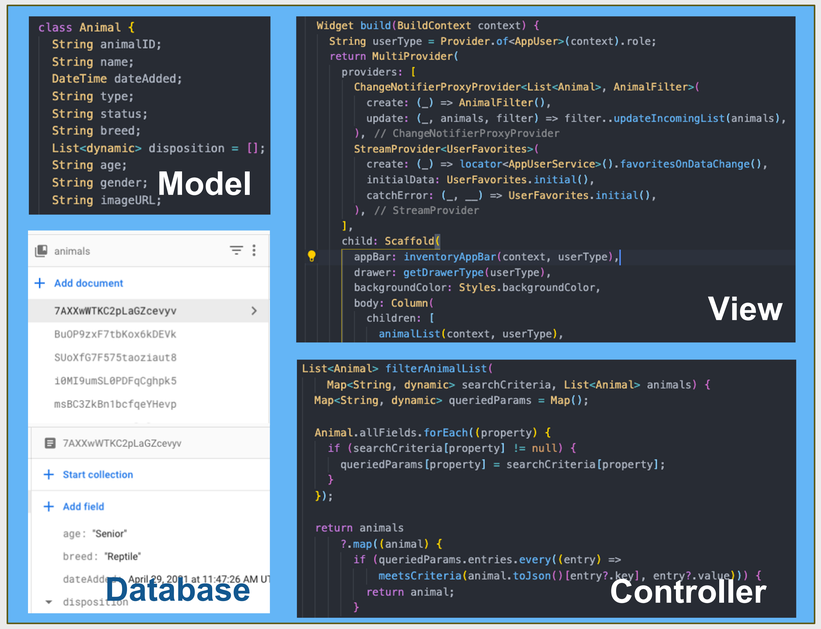
Duane Goodner (dmgoodner@gmail.com), Traci Goreham (traci.goreham@gmail.com),
Erica May (may.erica81@gmail.com)

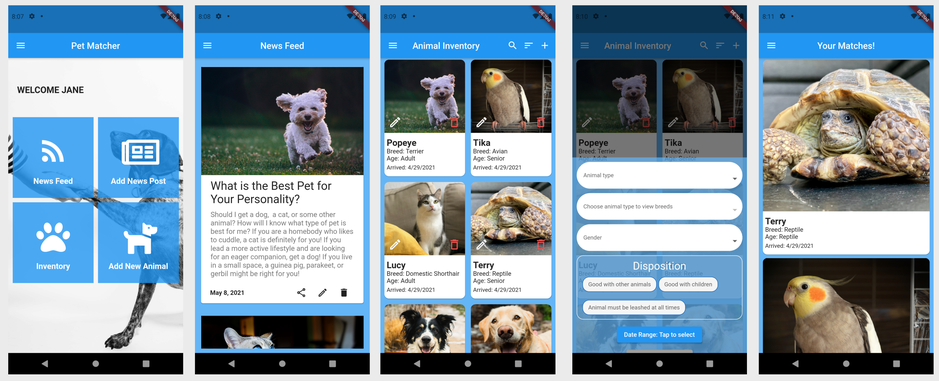
MOTIVATION

- Our primary objective was to create a cross-platform (iOS and Android) mobile application for connecting animal lovers in search of a new pet with animals in need of loving homes.
- We wanted to gain experience using the Flutter Software Development Kit and the Dart programming language to create a completely functional and visually attractive application a shelter could use for pet adoption and marketing objectives.
- Our goal was to integrate interface design and software architecture principles to produce an application with a high-quality user experience.



Reference: Paw print image obtained for free from https://wikiclipart.com/dog-paw-prints-clip-art_37264/





The above images provide a sample of screens in our application. Users can navigate to various sub-screens using tile icons in the homescreen. The same pages can be reached using menu items in a dropdown drawer that is available on the admin and user home screens and on sub-screens. (From left to right: Admin Home Screen, Admin News Feed, Admin Animal Inventory Screen, Inventory Search Form, User Animal Favorites Screen)

Reference: All animal images are personal photos or free images from https://unsplash.com.

IMPLEMENTATION / TECHNOLOGY

This application was written in the Dart programming language using the Flutter framework. Dart is a strongly typed, compiled, object-oriented language. Flutter can compile a single code-base to native iOS and Android code, allowing applications written with this framework to be cross-platform and fast.

Google Cloud Firestore was used for a non-structured database, image storage, and user authorization. Various Flutter packages are available for seamless integration with Firestore. We also relied heavily on data streams (rather than single-call queries) to easily keep the app state in sync with the database.

In a Flutter application, all UI components and many non-UI components are nodes (aka "widgets") in a tree data structure. Many of the details of passing state information from upper-level to lower-level widgets in our application were managed using the Provider package to minimize the amount of data passed through intermediate widgets. By using StreamProvider widgets that receive and store backend stream events, we were able to minimize the bandwidth needed to keep each UI view synchronized with backend data.

The application architecture primarily followed the "Model, View, Controller" paradigm. Many of the Service (aka Controller) and Model classes were managed using the Flutter GetIt package. This package allows easy control over the class type (e.g. Singleton vs. Factory) and provides clean access to specific class instances without requiring that business-logic be part of the widget tree.

APP FEATURES

- Supports accounts (administrative and public) and login
- App "remembers" the login state of device if user quits without logging out
- Has an overall landing page, as well as, a homescreen for each account type
- Admin user is able to create new animal profiles containing type of animal, breed, disposition, picture, availability, news item, and description
- Admin can edit and remove existing animal profiles
- Supports sorting by name, date added, animal type, status/availability, breed, age category, and gender
- Supports searching by type, breed, gender, disposition, and date added; selecting a particular type of animal causes breeds of only that animal type to be available in the breed dropdown
- Disposition descriptions are multi-select "chip" icons that include "Good with other animals", "Good with children", and "Animal must be leashed at all times"
- Administrators can post, edit, and remove news items to/from a news feed
- Public users can share news posts via text or email with others
- Public users can select (or de-select) their favorite animals when viewing inventory; favorite animals are shown on a user's matches page

Oregon State University