PAWWWZ

Lost Pet Finder chip/app: Implementation 2 Assignment

CS 361: Software Engineering I

Group 16

Haya Ahmed
Ryan Alcorn
Marc Baiza
Shannon Farazi
Matthew Koenig

Site URL

As our project is a native app. As previously discussed with the grading TA, our Github repo access has been shared along with a video demonstrating usage.

- The URL to our GitHub repository is: https://github.com/shfarazi/CS361Project
- Figma interactive prototype: https://www.figma.com/proto/kegCUBjMUNHCcbZoeaMZRK/Pawwwz?node-id=0%3A2&viewport=-350%2C262%2C0.13081438839435577&scaling=scale-down
- You can watch a new video demo of this week's progress on the app here: https://1drv.ms/v/s!Al4u8GzfHIkjgTpp8wgHErS2fbX0
    - As a screen capture video, this video has no sound. However, the new functionality is demonstrated.

User Stories Completed

1. Edit Pet Status/Added help to find pet with map

Who worked on this task?
Shannon, Ryan, Marc

What do the relevant unit tests do?

1. We tested if a pet status could be changed between Missing, Found, and Lost, and that these changes were visible in the user interface and it was successful.
2. We tested that these changes persisted once the Pet Profile page was refreshed.
3. We tested that these changes occurred on the map page and vice versa. It was successful.
4. We tested whether the user would now be able to click on the map button and display the pet's location on the map. It was successful.
5. We tested that the map could generate directions to the pet based off of the pet's location through the chip and the user's (owner/app user/shelter) location through their phone's GPS. It was successful.

What problems, if any, did you encounter?
There were no problems implementing this user story and the Map page containing the pet's status didn't change from the original prototype. The user can change the pet's status on there and it will also change on the Pet Profile page as well.

How long did each task require?
Implementing this story took around 6 hours to complete.

What is the current status?
The current status of the user story is that it is completed.

What is left to be completed?
This user story is completed.

2. Delete Pet Account

Who worked on this task?
Shannon, Matt, Haya

What do the relevant unit tests do?

1. We tested that the owner of the pet could delete their previously created pet account by clicking the Delete Pet Profile button in the Settings page and it was successful.
2. We tested if the information displayed on the Pet Profile page would be erased after deletion and it was successful.
3. We tested whether the user could enter a new pet information after deleting the previous Pet Profile and it was successful.

What problems, if any, did you encounter?
There were no problems implementing this user story.

How long did each task require?
Implementing this story took around 10 hours to complete.

What is the current status?
The current status of the user story is that it is completed.

What is left to be completed?
This user story is completed.

3. Logout

Who worked on this task?
Shannon, Matt, Haya

What do the relevant unit tests do?

1. We tested that the user can log out of the app by clicking the Log Out button in the Settings page and it was successful.
2. We tested that after logging out, the user is presented with the Login screen again and it was successful.
3. We tested whether the user would be able to log back in again after logging out and it was successful.

What problems, if any, did you encounter?
There were no problems implementing this user story.

How long did each task require?
Implementing this story took around 5 hours to complete.

What is the current status?
The current status of the user story is that it is completed.

What is left to be completed?
This user story is completed.

Review of Spikes and UML Sequence Diagrams

We would reiterate that overall the UML Diagrams provided a modest guide to the coding process, but were not without fault. Again, the diagrams lacked the ability to guide the linking process and what order the screens should be presented, but the prototype we initially created was able to sufficiently supplement the diagram(s).

Edit Pet Status/Added help to find pet with map

Our spike for this story illustrated to us that something wasn't quite right for the user, but gave us a guide as to what needed to be improved. As a result, we created a means of utilizing the built-in navigation feature of their mobile device to improve efficiency in locating a pet, vs. displaying a static map image. Pinpointing the pet location on the map leads a user to a scroll where they can select whether the pet is noted as 'home', 'found' or 'missing'. The user is then notified through a warning message that the status has been updated.

Delete Pet

Our spike for this user story led reinforced our decision to delete a pet rather than the user itself. When we completed the process for the latter, it was clear this was neither a realistic use by a pet owner, nor what we would want to see from non-owners, generally. As the application depends on a critical mass of users, not all of which can be pet owners, we needed to account for that. In addition, owners may either have additional pets and/or anticipate another pet in the future. Simply deleting a pet rather than the user themselves is a better way to accomplish these dual goals, as it wipes the pet profile, but leaves the user profile intact.

Logout

As with the login process, we utilized Auth0. Implementing this on a native iOS/Android app would have otherwise taken many hours, but with Auth0 it took less time. The time spent learning the platform was a wise investment, and the UML Diagram for the login provided a framework from which we could "reverse" the process (although eliminating some unnecessary steps was required).

Description of Refactoring

Added Delete Button

In our original implementation we did not have a "delete button." We decided to add this feature to make it easier for customers to remove their pets information in case of an unfortunate event. Whether that be the pet passed away or the user decided to remove themselves from the app. This delete button will remove the pet profile from public view, and remove the information from the database.

Added help to find pet using integrated maps app

In our original implementation we did not have this feature, but decided to implement it to provide convenience for the user. This feature allows the application to use the built-in navigation application to show the customer the nearest route to their pet. This will allow the user to have the quickest path possible using the already developed technology of google/apple maps.

Additional Customer Questions

This week, we demonstrated our three user stories for our customer, Keenon: 1) Edit Pet Status on the Map screen; 2) Delete Pet Profile; and 3) Logging Out. Once again, we presented these stories using a screen share of emulator and utilizing the Figma application.

One of the main features/necessities of the Edit Pet Status story is the ability to mark the pet as either ‘found’, ‘missing’, or ‘home’. In the version we presented to Keenon, Figma limited our ability to use a scroll bar. Therefore, we utilized a color coded button system where blue = home, red = missing, and green = found. We discussed that this was merely a limitation of Figma, but would not affect the ultimate application’s feature or functionality. Keenon understood this limitation of Figma tool and did not have any problem with the actual implementation of a picker for pet status in the app.

We considered the possibility of altering the Delete Pet Profile to Delete Account, but decided it would be more realistic to retain our original concept, as it would reflect a user who no longer had a specific pet, but still had others. However, neither version required the creation of a ‘Delete’ button. Lastly, we presented our logout story. In many ways, it mirrors the log in story, but is more streamlined. It simply exits the user information, and returns to the landing page. Keenon agreed with our decisions and liked what we did.

Keenon thanked our group for its’ hard work during the term, even going so far as calling it a “super group”. He indicated that he was very pleased with how everything turned out, and that it more than met his vision for the idea. Most notably, he appreciated that our final product very closely mirrored the prototype we originally presented. As a group, we appreciated that he was consistent with his decisions and honest with his feedback.

Description of Integration Tests

This week we worked on implemented three additional user stories. These user stories were Edit Pet Status and coordinate this with the Pet Status on the Map screen, Delete Pet Profile and Log out. In terms of the entire app, all of the user stories had to be integrated with our previously working version of the app. In keeping consistent with our previous iteration of the app, we are still using Xamarin Framework to build out our cross-platform app and testing our build using the iOS emulator and an iPhone. As with our previous submission, since our device is being tested in iOS and uses iOS specific packages, it may require additional refactoring in order to run on an Android emulator and Android phone.

Our integration tests were deemed successful if they meet certain requirements. General requirements after including this week’s user stories were: 1) project would build successfully in the iOS emulator, 2) it would be displayed and functional when tested on an iPhone, and 3) it

would not unexpectedly crash or navigate to other screens without being prompted. In other words, the integration of this week's user stories did not lead to a global compromise of our project's build. All of our user stories passed these requirements. The user story specific requirements included in the integration of this week's User Stories did not cause unexpected behavior or bugs in their own implementation, nor in the relevant pages the user story would be acting on, nor in the already implemented user stories. These tests for each of the new user stories are described in greater detail below.

The user story of Edit Pet Status and coordinating this with the Pet Status on the Map Screen required the Pet Profile and Map screen to be integrated. The Map screen configures a map using the built-in navigation on an iPhone. If our app was fully functional, the Map would be integrated with the chip and provide the real-time location of the User's pet. Although we did not have the technology or resources to implement this functionality, the Map screen is linked with other parts of the app, namely the user and the pet profile. The Map page currently shows the user's location via accessing their GPS location through their phone. Additionally, the current Pet Status can be viewed on the Map and changed on the Map page through a scrollbar. The changes to Pet Status on the Map screen are reflected in the Pet Profile screen. Likewise, changes to Pet Status (via a scrollbar) on the Pet Profile Screen are reflected in the Map screen. Overall, given that changes to the user's location and the pet status are reflected in the Map page is an indication that our Map screen and Edit Pet Status functionality is integrated with the rest of our app.

The Log Out functionality had to be integrated with Login functionality provided by Auth0. The Log Out button was included into the Settings Screen. When the user clicks the Log Out button, they are logged out of the Pawwwz app and are once again presented with the Login/Sign Up page. If the user logs in again with an existing account, their user information and pet information remains the same as it was before logging out, indicating that there is no information loss or any other unexpected issue with the log out process.

The Delete Pet Profile functions was integrated with the Pet Profile. When the Delete Pet Profile button on the Setting screen is clicked, the Pet Profile information is set to null. The User Profile information remained the same, which indicates that this functionality was properly integrated and did not cause unexpected behavior such as deleting the user information as well. Once the Pet Profile is deleted, the user can then re-enter a new pet information. An extension of this functionality would be to allow a user to delete a specific pet if they had multiple pets, however the user story description for this week did not require that.

Overall, our system successfully integrated this week's user stories with the rest of the app and the previous week's user stories. The Map's screen was integrated into the app and the user is able to view their and their pet locations. Moreover, they can view and edit their pet's status. Changes to the pet's current status from the Map screen is reflected in the Pet Profile screen and vice versa. The log out functionality allows the user to log out without losing their information, and they are presented with the Login/Sign Up screen again. The user is able to

Login with any existing account if they desire. The user is also able to delete an existing pet profile without losing their own information. The changes from deleting the Pet Profile persist in subsequent sessions by the same user.

Implementation Plan for Next Week

After a thorough review process, the following three user stories were selected for next week:

1) Check The Map
2) View Other Pet Profiles
3) Update Search Proximity

Our choices for next week reflects a shift to more substantive stories previously outside the reach of this course.

The below chart represents our planned workflow to complete next week.

12/7 Saturday		
1 hour	Team meeting	Everyone
1 hour	Create dummy entries for database	Haya
12/8 Sunday		
9 hours	Coding	Marc, Ryan, Shannon
12/9 Monday		
4 hours	Coding	Shannon, Haya, Matt
12/10 Tuesday		
1 hour	Met up with Customer	Everyone
12/11 Wednesday		
2 hours	Write up refactoring, Review UML	Marc, Shannon, Matt
2 hours	Write up integration tests/User stories	Haya, Ryan
3 hours	Test Code and write up	Marc and Matt

12/12 Thursday		
1 hour	Finalize weekly report and create hypothetical future schedule	Marc and Matt

Customer

Our customer, Keenon, met with our entire team on Wednesday, December 4th. The group met up with him for about an hour and showed him the app through screen share. He was overall very impressed by the app and liked how it implemented features laid out in previous weeks through the prototypes and diagrams shown to him. He was very helpful and willing to answer any follow up questions, if needed. As was the case all throughout the term, he carefully considered our presentation, and provided helpful feedback. Would we have continued on for additional weeks, we believe he would continue to do so.

Team Member Contributions

- Haya: Description of Integration Tests, User Stories Completed
- Matt: Additional Customer Questions, Review of Spikes/UML
- Marc: Refactoring description, testing.
- Ryan: User Stories Completed
- Shannon: User Stories Completed, Integration Tests
- All: Meet with customer, Strategy and Planning, Editing, Coding, Testing