PAWWWZ

Lost Pet Finder chip/app: Implementation 1 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/file/kegCUBjMUNHCcbZoeaMZRK/Pawwwz?node-id=0%3A1
    - Please wait for all pictures to load, then click the play arrow in the upper-right corner to enter the interactive prototype.
- You can watch a video demo of the current state of the app here: https://1drv.ms/v/s!Al4u8GzfHIkjgRV5rtIOAmBleIVt
    - As a screen capture video, this video has no sound. However, all current functionality is demonstrated.

User Stories Completed

1. User Login

Who worked on this task?
Shannon, Ryan, Marc, Matt, Haya through coding and testing

What do the relevant unit tests do?

1. We tested if login information that the system already knows would be remembered and the result was successful.
2. We tested if login information that the system didn't know would create new login information and it was successful.
3. We tested that if the app is already running that the user wouldn't have to enter login information again and it was successful.

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

How long did each task require?
Implementing this story took around 4 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. Creating user and pet profile

Who worked on this task?
Shannon, Ryan, Marc, Matt, Haya through coding and testing

What do the relevant unit tests do?

1. We tested whether the user would be able to click on the user profile/pet profile icon at the bottom of the app and it was successful.
2. We tested whether the user would be able to enter information into the fields on the pet profile/user profile and it was successful.
3. We tested whether the app would remember information the user entered into the fields on these pages and it was successful.
4. We tested whether the app would remember information the user entered into the fields on these pages after the app is closed and it was successful.

What problems, if any, did you encounter?
There were many UI changes from the prototype versions to the actual app. We felt that these changes made more sense in the actual app so there was discussion about how the actual pages would look on the app. This includes adding a button to save the information entered by the user and where these buttons are located on the menu.

How long did each task require?
Implementing this story took around 20 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. Mark pet status as found

Who worked on this task?
Shannon, Ryan, Marc, Matt, Haya through coding and testing

What do the relevant unit tests do?

1. We tested whether the user would be able to click on the user pet profile icon at the bottom of the app and it was successful.
2. We tested whether the user would be able to enter information for the pet on the pet profile page and it was successful.
3. We tested whether the user would be able to change the pet's status to found and it was successful.
4. We tested whether the user would be able to change the pet's status to other statuses and it was successful.

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

How long did each task require?
Implementing this story took around 4 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

Overall the UML Diagrams provided a modest guide to the coding process, but were not infallible. 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.

<u>User Login</u>
We utilized Auth0 to streamline the login and logout process, which was very successful. Implementing this on a native iOS/Android app would have otherwise taken many hours, but with Auth0 it took less time. This sort of adaptation served as a great learning tool for the remainder of the coding of the app. Rather than force a square peg into a round

hole, we learned that there is sometimes a better way to do things and changing course is often the best option.

Create User & Pet Profile

The learning curve to implement the prototype as actual screens is significant, but we eventually overcame it. Again, the prototype served as just a valuable tool as the UML Diagram owing, mainly, to the visualization aspect. The UML Diagram serves as a nice checklist of tasks the app should accomplish, but it does not always result in full implementation. In theory, a UML Diagram would consist of every task the previously created prototype noted, but the limited nature of the UML sometimes leaves pockets of unaddressed issues.

Mark A Pet Found

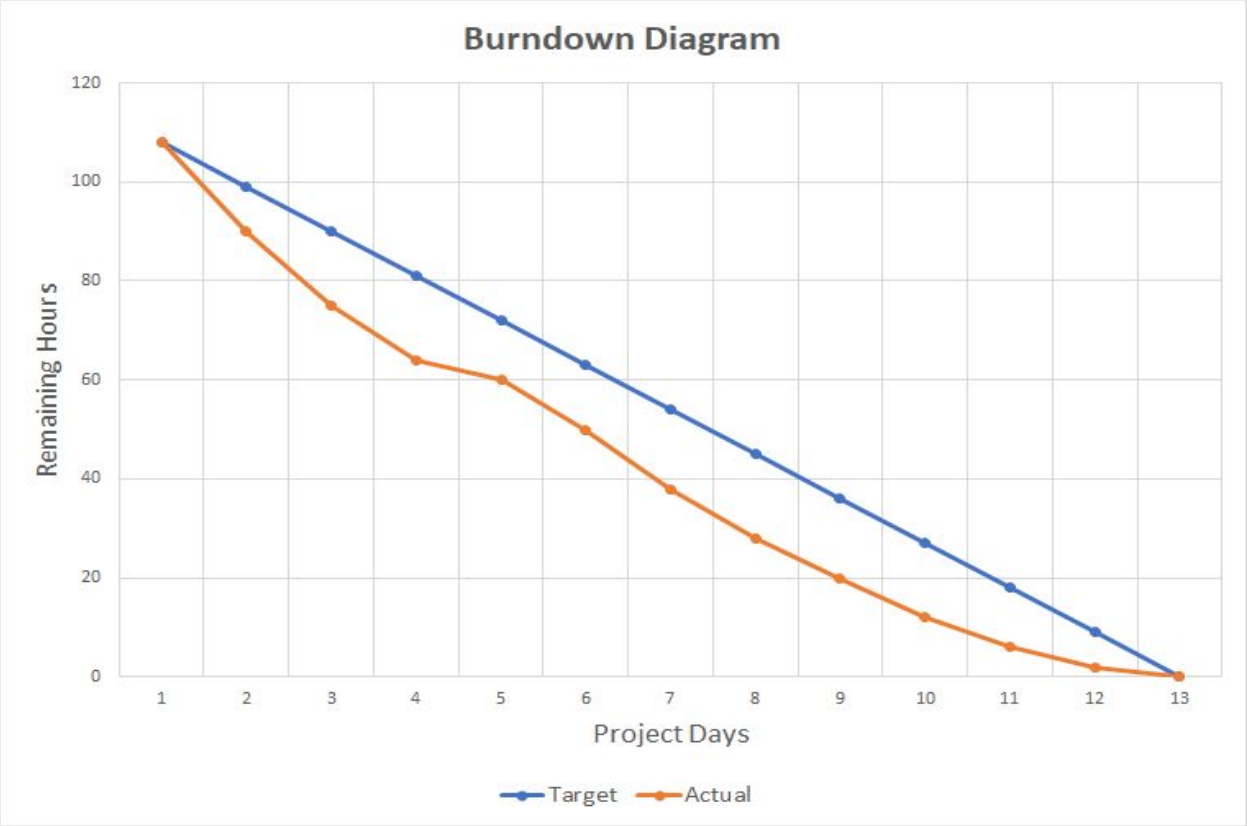
Once we overcame the hurdle of implementing our designs in XAML, the main challenge was developing our API and database, and integrating it with our app. We drew on our experiences from other courses in the program more so than the diagram(s) to accomplish this and other user stories. We decided not to use the combination of API and SQL, but rather we utilized the power of Azure Cosmos DB. Because Cosmos DB is accessible through an HTTP web service interface, we were able to skip building our own API as a middleman. We believe much of the code (or at the process) will provide a guide to our other related user stories, such as Navigate To A Lost Pet and View Local Lost Pets.

Other Diagrams

We utilized a number of diagrams throughout this class and our software development life cycle. We routinely used sequence diagrams, especially in developing our user stories in HW5. They focus mainly on the sequence of events/interactions amongst people and things. We also used a UML Class Diagram, but found that it was, ultimately, of limited benefit. Namely, because it gave a solid construction of the classes, but didn't have the breadth we truly needed using a Xamarin app.

The one diagram that would have been beneficial is an Activity Diagram, which specifically tracks user behavior. We believe it would have been helpful in providing a guide to our app's usability, particularly with respect to user/customer input.

Burndown Diagram



Description of Refactoring

Removing Buttons from Homepage

In our original implementation/mock we had 6 buttons. This included “Report Missing”, “Pet Profile”, “Map”, “User Profile”, “Scan”, and “Settings.” We ended up reducing the total buttons on the screen to 3, and moving some down to the horizontal buttons on the bottom of the screen. So our “Map”, “Home”, “Profile”, and “Pet Profile” buttons are on the tab bar with the others on the main homepage.

Added Save Button to Pet Profile creation

We originally overlooked a “save” button to our pet profile creation so we added the functionality in so a user could save their information after entering.

Removed some back-buttons for functionality

We found that most of the pages did not need “back buttons” as our original mock-up had. As the 4 horizontal buttons were always accessible at the bottom to take you back to the homepage. We removed some of these buttons to reduce clutter, and increase functionality.

Pet profile wording updated for usability/alignment

On our pet profile creation page we updated the wording of some of our sections so everything was properly aligned. This was for presentability reasons, and to have a cleaner looking experience/application.

Added a “share info” switch to allow users to see information publicly

Initially, each category in our pet profile page had a switch. This switch served to make the information public/private. We found that this was redundant, and grouped some of the basic information together that is always public such as the user’s picture and name. Some of the information such as the user's last name, email and phone number will be optional to be toggled to be made public/private at any time. The rest of the user’s info will remain private indefinitely. This reduced clutter as we did not have a switch for every single category. As far as code goes we were able to easily go from a multi-switch system to a one-switch system.

Additional Customer Questions

This week we implemented our log-in process, creating the user and pet’s profiles, and marking the pet as found. On 6pm Saturday we got together with our customer Keenon to discuss our implementations. We asked him to look over what we had and give us feedback so that we could fix it before the due date. He said that we met his expectations, and made the application very similar to our prototype. He liked this because it was exactly what he was expecting, and we did not compromise on usability to create the proper functionality.

We asked Keenon about the changes we made which included button changes when it came to the home page. In our mock our homepage consisted of 6 main buttons with four other buttons at the bottom. We found that the 4 buttons at the bottom were the same as some of our 6 “main buttons” on the screen so we reduced the final screen to 3 buttons. Keenon confirmed that this was a good change, and reduced clutter. We also went over alignment and formatting and he liked the changes we made to the UI.

Overall, Keenon was impressed with our implementation, and liked the functionality. He said that it is exactly how he expected a user should interact with his app. He was impressed that we were able to take some initiative, and improve the application on our own. Overall he like the application, and did not make any request to change what we have created.

Description of Integration Tests

This week we implemented three basic, but necessary tasks in our application. These tasks included the ability to login to the app, the ability to create a pet profile/user profile, and the ability to mark a pet's status as found. Along with implementing those features our primary focus was to make sure we added the proper functionality within the app to make it a smooth experience for users. The app was designed in Visual Studio through the Xamarin framework which made it possible to create a familiar experience for the user, through native iOS and Android apps. We are currently building both iOS and Android apps, but the app is primarily tested in iOS right now. Some of the functionality might not work on other platforms or will require additional improvements/refactoring to get it functioning at the same level on those platforms.

A big aspect of testing how everything was implemented was first laid out in our prototypes that were created in an earlier week. Overall, the app contains the pages laid out in those prototypes. There were differences on some pages to account for more usability of the app. Customers expect an app to function a certain way so it was important to make changes from the prototypes that fit what the user expects like having important buttons at the bottom of the screen like the "Home" button. There were also features that we didn't think of when creating those prototypes like saving information entered by the user through a button so they won't have to re-enter in their pet/user information.

To describe how everything is integrated would require running through the features of the app presented through Xamarin. The app appears on the emulated iOS screen for the user to click on and we tested that the user would be able to click on this icon. The icon contains the image[1] that is featured on the first page of this document and the icon contains the name of the app. After clicking on the icon, the user is taken to the login page where they can click on a button to enter information to login. The user will be required to enter valid login information and their account will be remembered if the login information already existed through the database. This step only happens if the app wasn't open before. Otherwise, the user only has to click on the login button and they are taken through the normal process of the app.

There is a popup on login that says, "'Pawwwz' Wants to Use 'auth0.com' to Sign In: This allows the app and website to share information about you". This message is somewhat unclear, but is being displayed because we are using a free tier of Auth0 and do not have our own domain name registered. We tested that all of these features work and the user is able to proceed after entering their login information.

After finishing the login, the user is taken to the pet alert page that will allow the user to potentially click on the button to send out an alert that the pet is missing. At this time, that button is not implemented so users would not be able to send that information. There is also a button on that page called "Home Page" that allows the user to go to the rest of the app with the menu.

There was a change from the prototypes where this page in the app had more buttons listed vertically. To allow the user to have an experience similar to other iOS apps some of those buttons were put down at the bottom of the app. This allows the user to click on buttons at the bottom like the user profile/pet profile page and give them an experience that they are more used to with other apps on their phone.

We tested that once the user reached the menu page that they would be able to click on those buttons and go to the pet profile or user profile page along with the map. At this time, the map button does not function as intended because that feature was not apart of this week's stories. The Xamarin framework allowed this functionality to happen easily and the user is able to get to pages like the user profile and pet profile once they press those buttons. The user can then enter the information for the pet or user profile. It was tested that the information entered will be saved after the user clicks on another page in the app. It was also tested that the information will be saved after the app is closed. There is also an option integrated to change the pet's status to found or to another status which was another user story from this week. The user does have to click on a button to save the information they entered on these pages. We tested that the button works and that the app will save information after the user when the user exits the page either by clicking on another page or exiting the app. The user can click on a button at the bottom of the screen to go back to the home page at any time to see other pages of the app.

There are other features of the app that we tested that don't work right now. These are primarily features that weren't part of the user stories that we chose this week. One of these features is the scan button. Clicking on that button will take the user to an appropriate page where they can scan the chip of the pet, but does not have that capability yet. This would have been a very tough feature to implement because scanning the chip requires different technology. There is also a button on the pet profile page that says "Pet History". We also tested this button by clicking on it and it doesn't take the user to the appropriate page to see the pet history at this time. The map feature was included at the bottom of the app as one of the buttons the user can click on no matter what page they are on after getting past the login process and pet alert page. We tested whether this button would work right now as the user intended and it currently does not display the appropriate map.

Overall, the system is very well integrated for these three stories. The user is able to login and have their login information remembered. They can then go through the pet alert page to find the home page. From there they can click on the other buttons to take them to other pages like the pet profile or user profile page. They can enter information about the pet or user and then return to the home page with use of the buttons at the bottom of the app. The user is also able to exit the app and have their information remembered including information that they used to login along with information entered in fields for the user and pet profile page.

Implementation Plan for Week 7

After a thorough review process, the following three user stories were selected for HW7:

1) Edit Pet Status
2) Deactivate Pet Account
3) Logout

Our choices for HW7 reflect reasonably achievable user stories given the time limitations and the advice provided by the instructor regarding unavailable technologies.

The below chart represents our planned workflow to complete HW7.

11/26 Tuesday		
6 hours	Coding	Shannon
11/27 Wednesday		
1 hour	Team meeting	Everyone
1 hour	Draft HW7 Document	Matt and Marc
11/29 Thursday		
10 hours	Coding	Shannon and Haya
11/30 Saturday		
12 hours	Coding	Ryan, Shannon
1 hour	Meet with customer	Shannon, Ryan, Marc
12/1 Sunday		
2 hours	Write up refactoring	Marc
2 hours	Write up integration tests	Ryan
3 hours	Test Code and write up	Marc and Matt
12/2 Monday		
8 hours	Coding	Shannon and Haya
1 hour	Meet with customer	Shannon, Matt, Haya
2 hours	Coding and Write up integration tests	Ryan, Shannon

12/3 Tuesday		
1 hour	Finalize HW7 & create hypothetical future schedule	Marc and Matt

Customer Meeting

Our customer, Keenon, met with our team on Saturday, November 30th. 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. The main changes/upgrades that he wanted were for features not apart of this week's stories, but they might be implemented next week as part of the process. The group plans on meeting up with him again next week to discuss those stories.

Team Member Contributions

- Haya: Database, User Stories Completed
- Matt: Review of Spikes and UML Diagram, Burndown Diagram
- Marc: Refactoring, Additional Customer Questions
- Ryan: User Stories Completed, Integration Testing
- Shannon: User Stories Completed, Integration Testing, Refactoring
- All: Meet with customer, Strategy and Planning, Editing, Coding, Testing

References:

1) Media assets from https://pixabay.com .