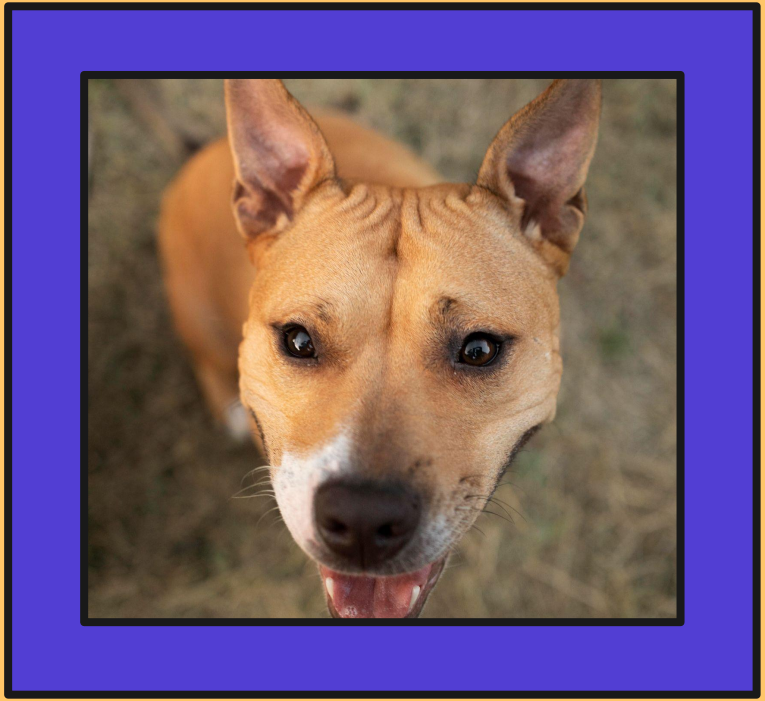
PetPal

Nathaniel Brown, Noah Acosta, Lucas Noack, Tera Parish, Rachel Bang, Aidan Duong & Ritvik Chilakamarthy

“Bumble for pet adoption”

Our mobile app shows users pets that need adopting in their area, and the user can swipe (left or right) to match with their potential future pet and the shelters or owners housing them.



Our motivation

For too long, pet adoption has been a convoluted and abstract process. By offering a location-based matching service, users are able to both give away or adopt a beloved pet near them. We hope that our app will prevent any further animals from being euthanized in shelters (as a result of never being adopted), abandoned by owners who no longer want them, and so much more.

What sets us apart?

Inclusivity

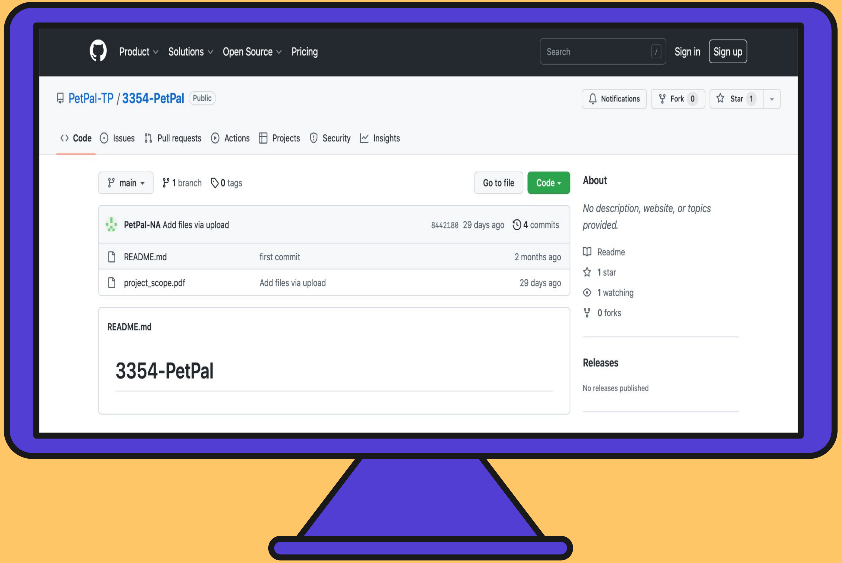
Our adoption platform is open to pets in any and all forms, not just cats & dogs.

Personal

Private pet owners, not just shelters, are welcome to give away their pets after undergoing a background check. We understand that giving away a loved one to a shelter can be heartbreaking, so we seek to be the bridge between two loving homes.

Our Github Repository

https://github.com/PetPal-TP/3354-PetPal



PetPal-NA Add files via upload		8442180 29 days ago 4 commits
README.md	first commit	2 months ago
project_scope.pdf	Add files via upload	29 days ago

README.md

3354-PetPal

About

No description, website, or topics provided.

Readme

1 star

1 watching

0 forks

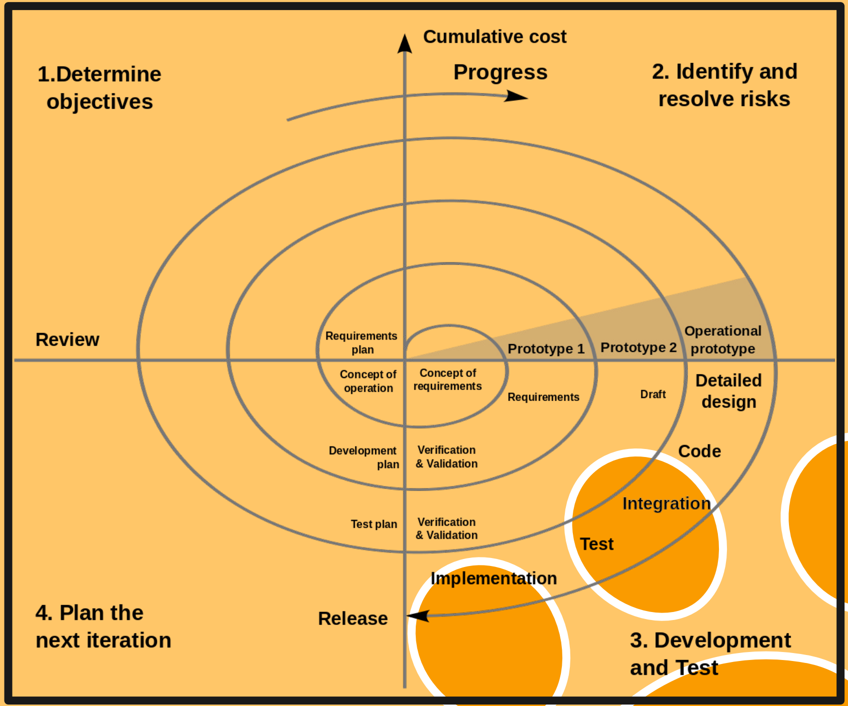
Releases

No releases published

Software Process Model

➔ Spiral Model

- At the beginning stages of our project, the specific requirements are not yet known; allows us to understand, react, and adapt to changing requirements and risks.
- Most closely reflects how development progresses in the real world; as the group keeps iterating through multiple cycles, our project will progressively become more and more complete, building upon previous iterations.



Functional Software Requirements

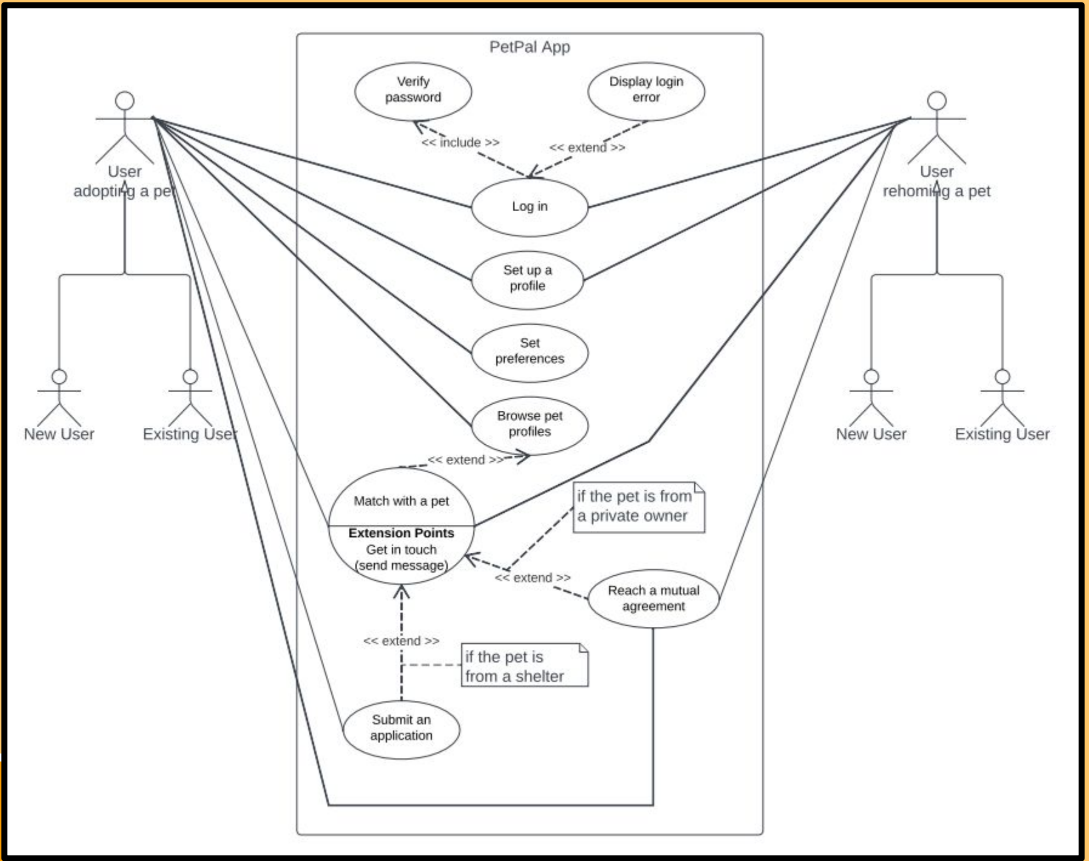
A user shall be able to:

- Create a profile for themselves or their pets, containing relevant information about them
- Message any other user they ‘match’ with
- Select requirements in order to specify what qualities they wish to look for in their ideal match
- Browse other users which match their requirement, and view these other users’ profiles
- ‘Swipe left’ to reject a candidate, or ‘swipe right’ to accept a candidate
- View other users who ‘swiped right’ on them, view their profile, and swipe back in return

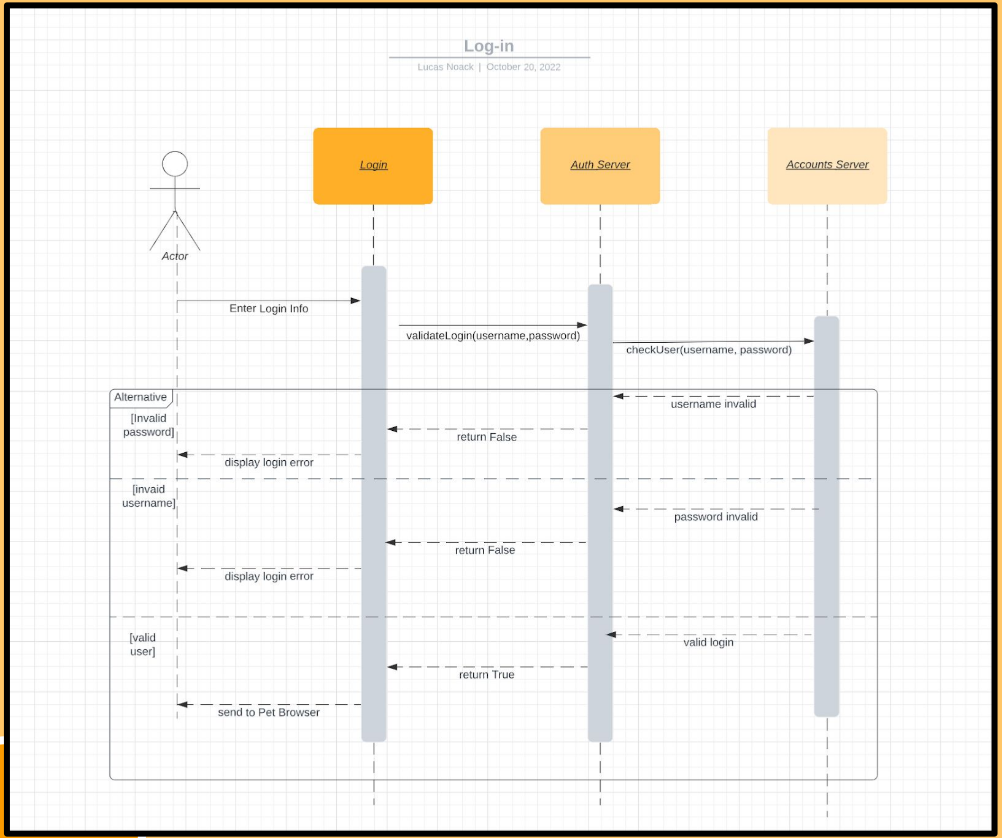
Non-functional Software Requirements

Usability	The ‘swiping’ system shall be explained after the user creates their profile such that users are not confused on how the app’s primary system, the matching system, works
Space	The application shall not take up more than 1GB of space
Dependability	The software development team shall continuously monitor the application to fix all bugs or errors as soon as possible.
Security	The application will store passwords in a secure location to avoid access by hostile entities
Environmental	The app must take weather into account when determining adoption dates and types. It should keep an eye out for where a pet will be rehomed (eg. it should warn that a Husky should not live in a hot area)
Operational	The application must be able to perform all actions requested by the user
Development	The software must be programmed in a high-level programming language and implement a database management system such as SQL
Accounting	The application must include a security fail-safe to protect user’s account
Safety **Security**	If we allow users under 13 to be on the app (if a family wants to allow their child to choose a pet to adopt), the app must abide by COPPA

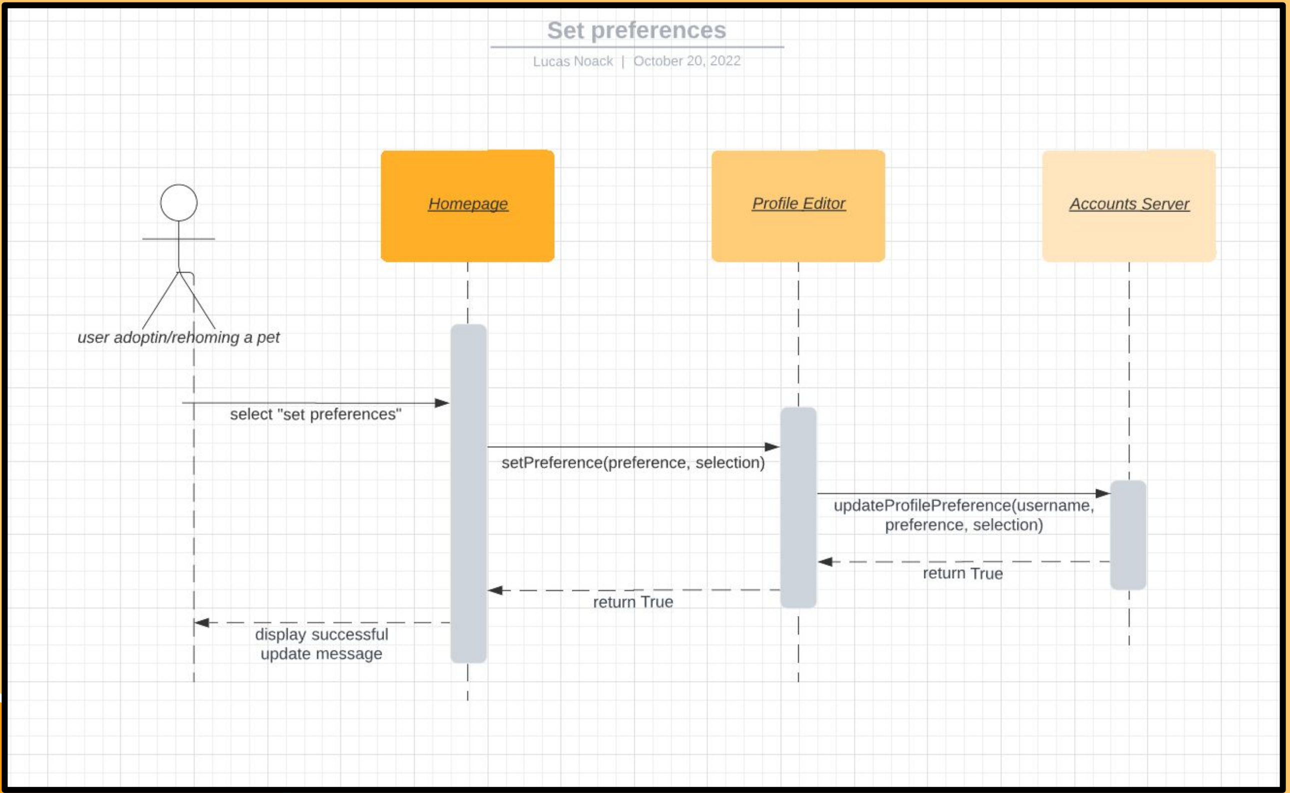
Use Case Diagram



Sequence Diagram 1



Sequence Diagram 2

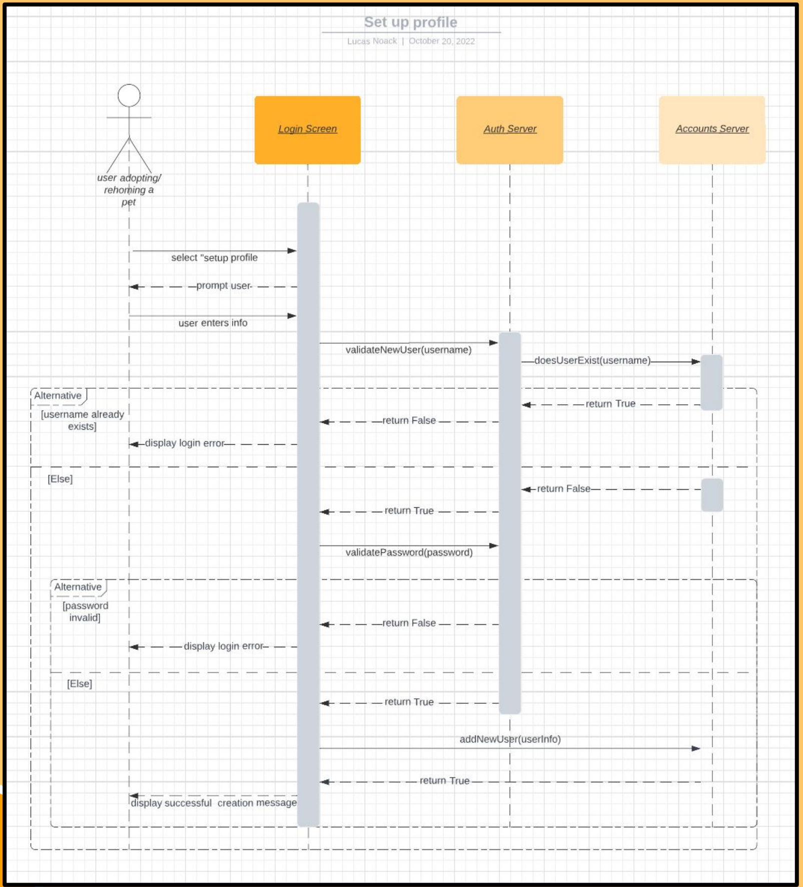


Set preferences

Lucas Noack | October 20, 2022

user adoptin/rehoming a pet
Homepage
Profile Editor
Accounts Server
select "set preferences"
setPreference(preference, selection)
updateProfilePreference(username, preference, selection)
return True
return True
display successful update message

Sequence Diagram 3



Set up profile

Lucas Noack | October 20, 2022

user adopting/ rehoming a pet

Login Screen

Auth Server

Accounts Server

select "setup profile

prompt user-

user enters info

validateNewUser(username)

doesUserExist(username)

Alternative

[username already exists]

return True

return False

display login error

[Else]

return False

return True

validatePassword(password)

Alternative

[password invalid]

return False

display login error

[Else]

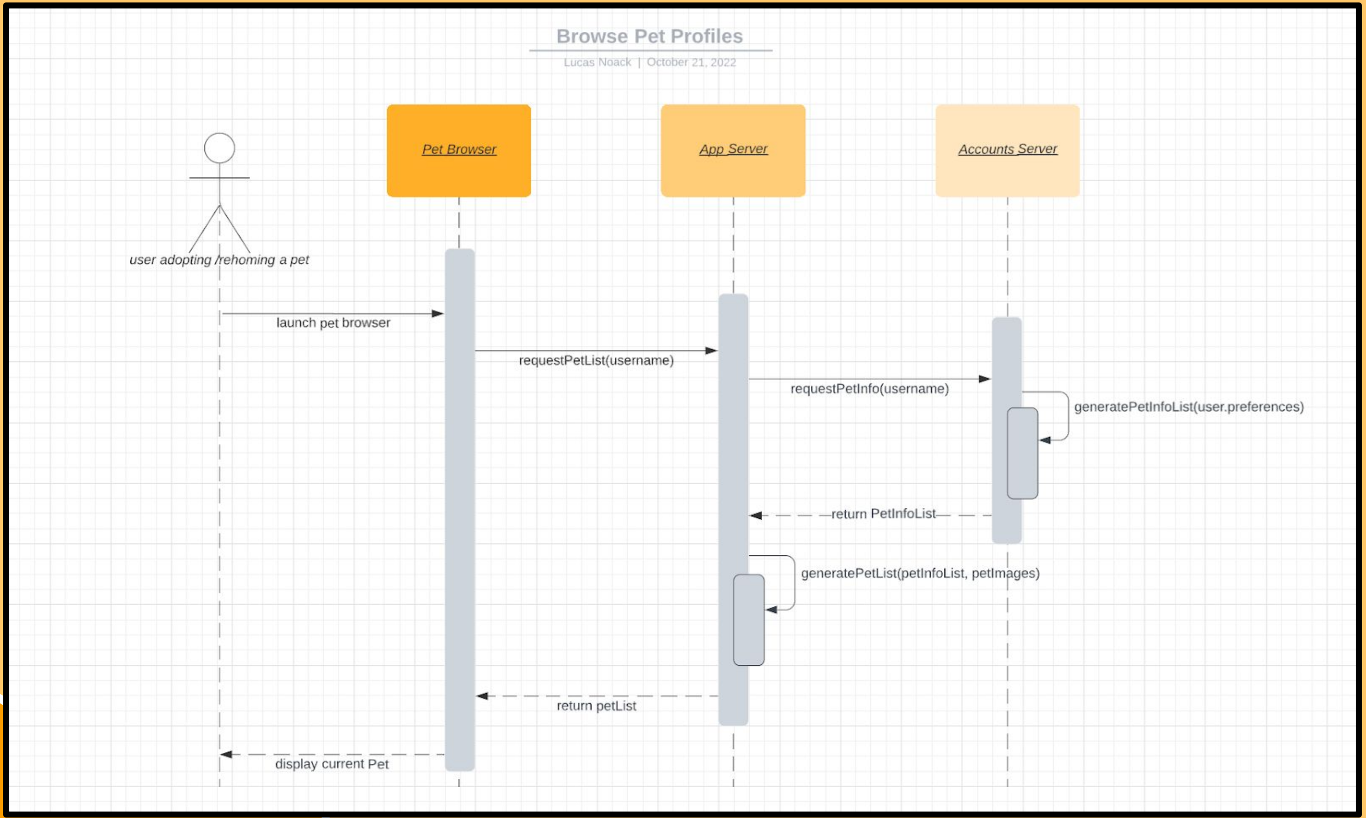
return True

addNewUser(userInfo)

return True

display successful creation message

Sequence Diagram 4



Browse Pet Profiles

Lucas Noack | October 21, 2022

Pet Browser

App Server

Accounts Server

user adopting /rehoming a pet

launch pet browser

requestPetList(username)

requestPetInfo(username)

generatePetInfoList(user.preferences)

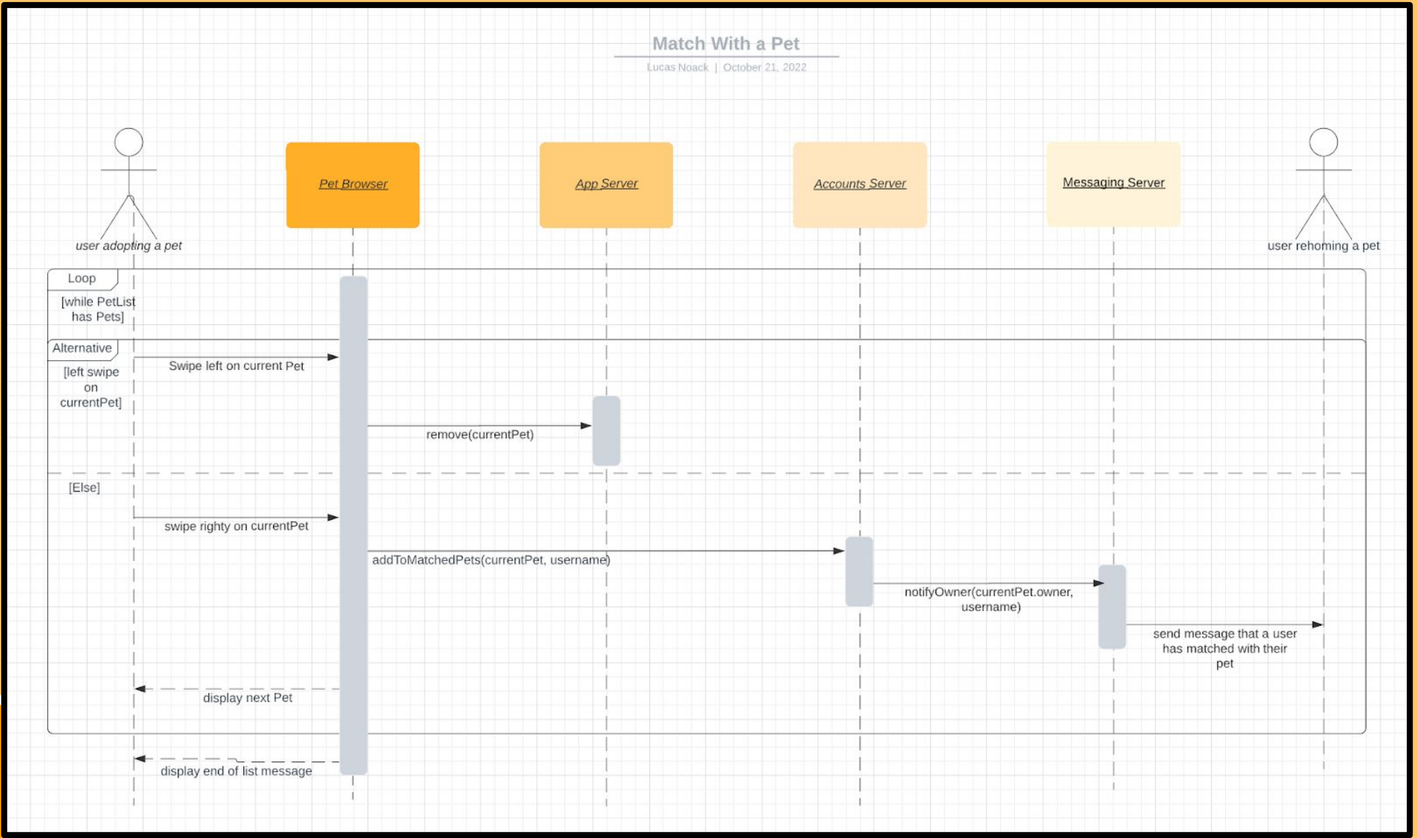
return PetInfoList

generatePetList(petInfoList, petImages)

return petList

display current Pet

Sequence Diagram 5



Match With a Pet

Lucas Noack | October 21, 2022

user adopting a pet

Pet Browser

App Server

Accounts Server

Messaging Server

user rehoming a pet

Loop

[while PetList has Pets]

Alternative

[left swipe on currentPet]

Swipe left on current Pet

remove(currentPet)

[Else]

swipe righty on currentPet

addToMatchedPets(currentPet, username)

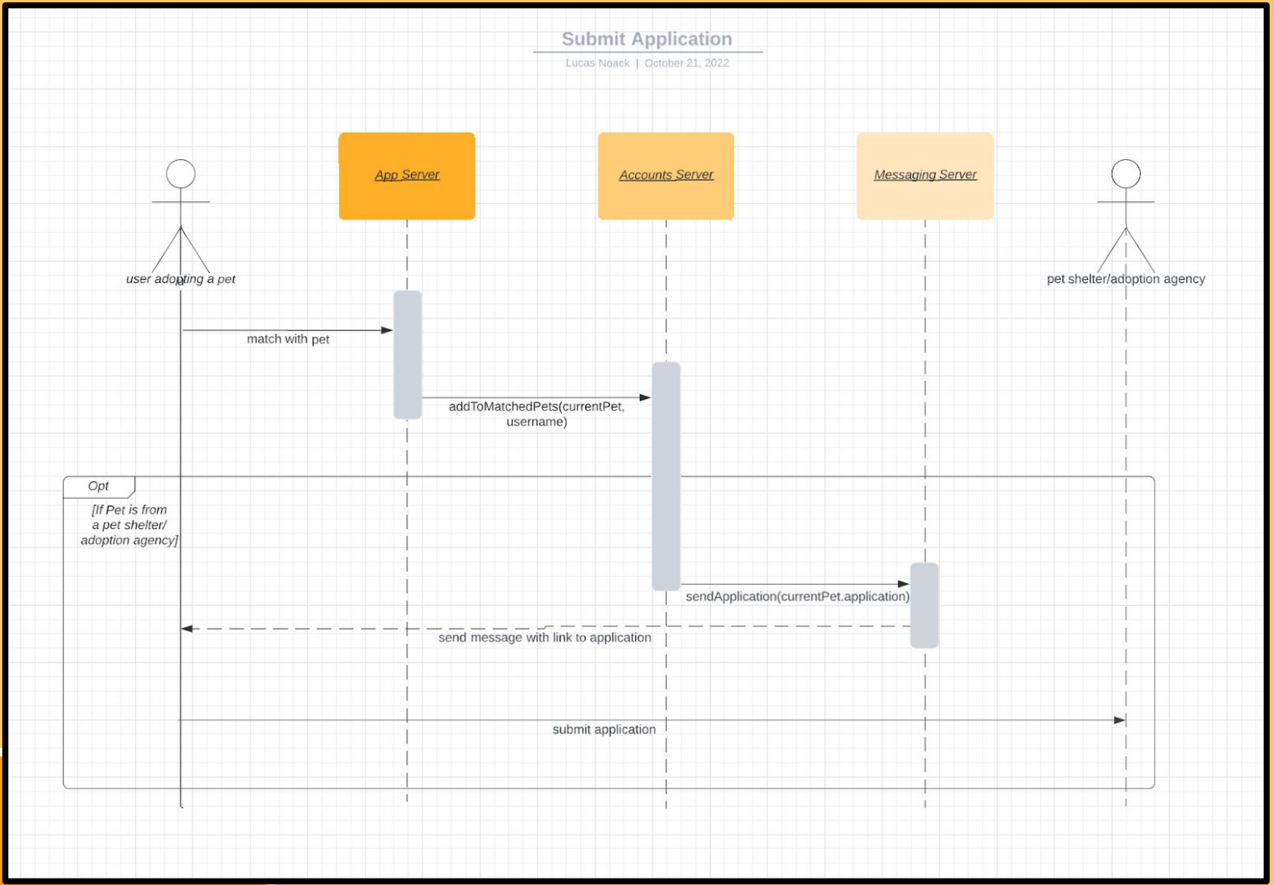
notifyOwner(currentPet.owner, username)

send message that a user has matched with their pet

display next Pet

display end of list message

Sequence Diagram 6

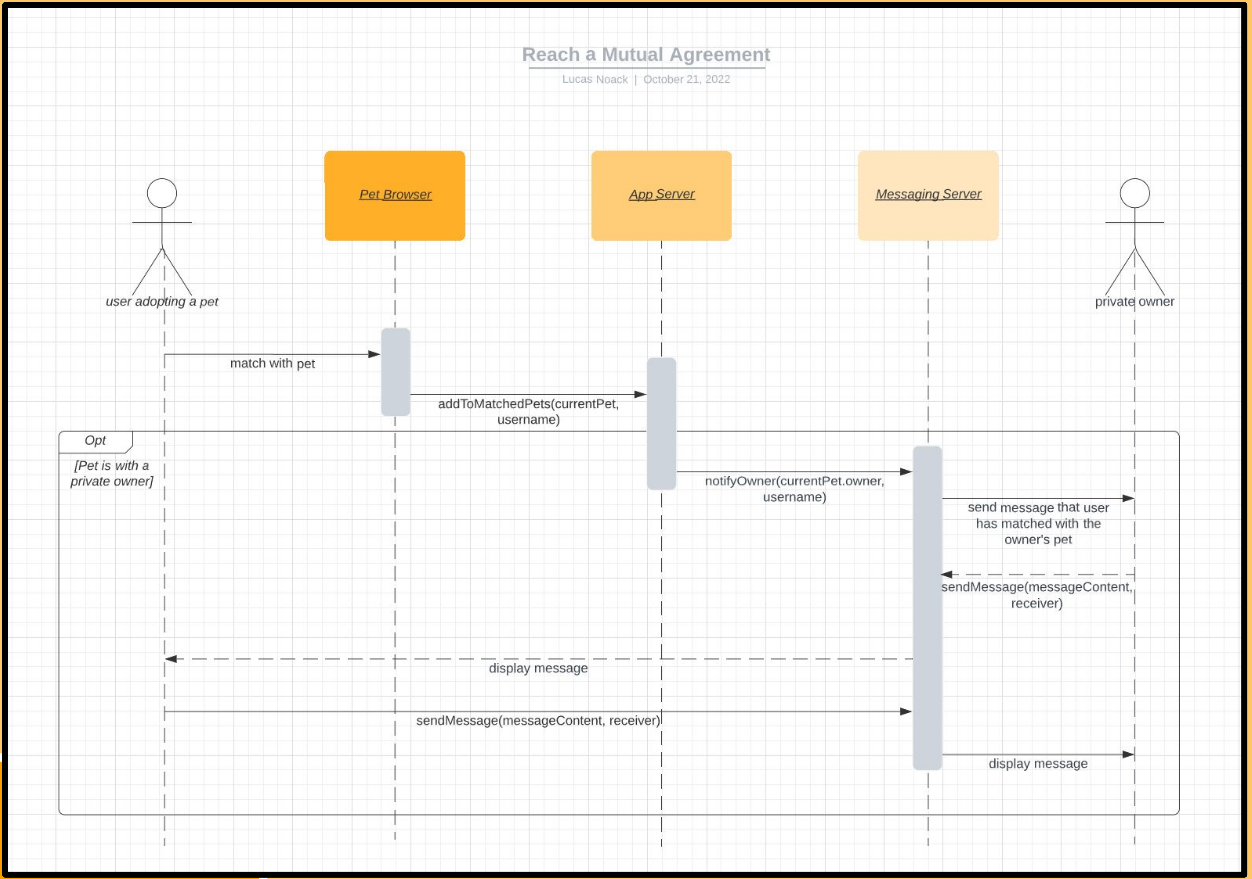


Submit Application

Lucas Noack | October 21, 2022

user adopting a pet
App Server
Accounts Server
Messaging Server
pet shelter/adoption agency
match with pet
addToMatchedPets(currentPet, username)
Opt
[If Pet is from a pet shelter/ adoption agency]
sendApplication(currentPet.application)
send message with link to application
submit application

Sequence Diagram 7



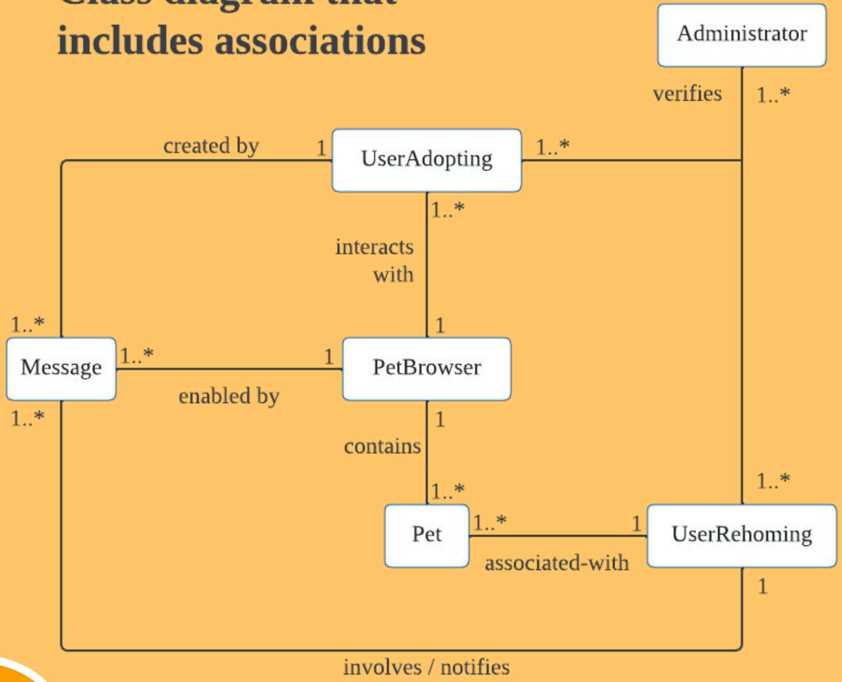
Reach a Mutual Agreement

Lucas Noack | October 21, 2022

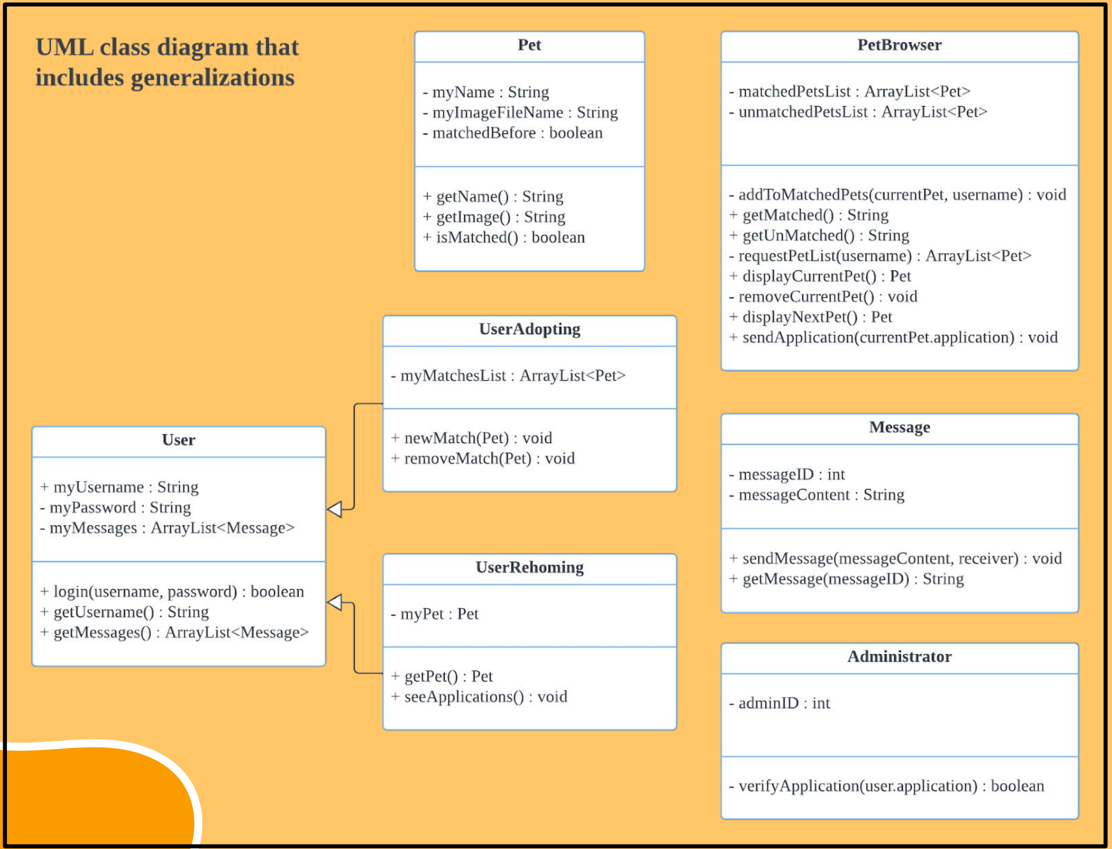
Pet Browser
App Server
Messaging Server
user adopting a pet
private owner
match with pet
addToMatchedPets(currentPet, username)
Opt
[Pet is with a private owner]
notifyOwner(currentPet.owner, username)
send message that user has matched with the owner's pet
sendMessage(messageContent, receiver)
display message
sendMessage(messageContent, receiver)
display message

Class Diagram 1

Class diagram that includes associations



Class Diagram 2



UML class diagram that includes generalizations

Pet

- myName : String
- myImageFileName : String
- matchedBefore : boolean

+ getName() : String
+ getImage() : String
+ isMatched() : boolean

PetBrowser

- matchedPetsList : ArrayList<Pet>
- unmatchedPetsList : ArrayList<Pet>

- addToMatchedPets(currentPet, username) : void
+ getMatched() : String
+ getUnMatched() : String
- requestPetList(username) : ArrayList<Pet>
+ displayCurrentPet() : Pet
- removeCurrentPet() : void
+ displayNextPet() : Pet
+ sendApplication(currentPet.application) : void

UserAdopting

- myMatchesList : ArrayList<Pet>

+ newMatch(Pet) : void
+ removeMatch(Pet) : void

User

+ myUsername : String
- myPassword : String
- myMessages : ArrayList<Message>

+ login(username, password) : boolean
+ getUsername() : String
+ getMessages() : ArrayList<Message>

Message

- messageID : int
- messageContent : String

+ sendMessage(messageContent, receiver) : void
+ getMessage(messageID) : String

UserRehoming

- myPet : Pet

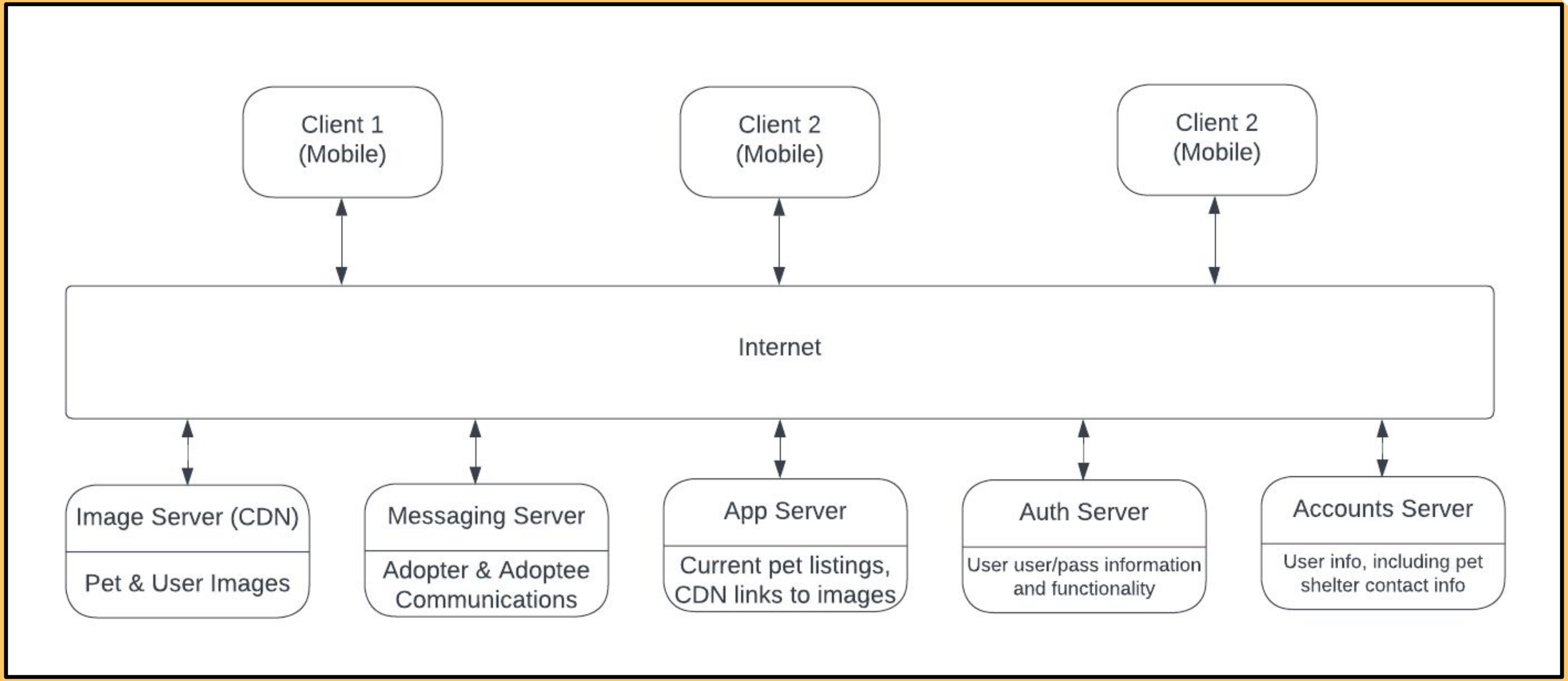
+ getPet() : Pet
+ seeApplications() : void

Administrator

- adminID : int

- verifyApplication(user.application) : boolean

Client-server Architecture

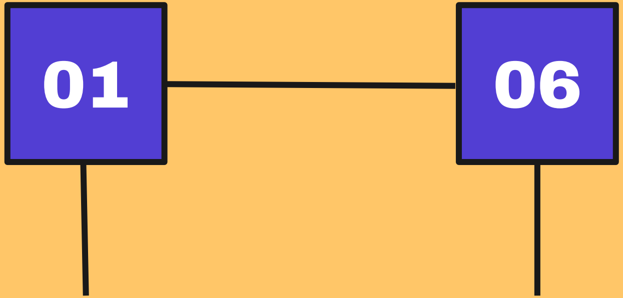


Client 1 (Mobile)
Client 2 (Mobile)
Client 2 (Mobile)
Internet
Image Server (CDN)
Pet & User Images
Messaging Server
Adopter & Adoptee Communications
App Server
Current pet listings, CDN links to images
Auth Server
User user/pass information and functionality
Accounts Server
User info, including pet shelter contact info

Client-server Architecture

- **Image Server (CDN):** Server configured as a Content-Delivery Network (CDN) specifically for serving lots of images to users. That way users are able to see pet images quickly, as they're delivered by a specialized server capable of transferring high amounts of binary image data quickly.
- **Messaging Server:** This server will operate at low latency to allow users to contact pet listers on a quick and secure server to facilitate pet adoption.
- **App Server:** This will server as the main server the user's device initially points to, generating/retrieving the pets that are adopted in the user's area, links to pet images, and more relevant information.
- **Auth Server:** Handles user authentication and account creation in a secure manner.
- **Accounts Server:** Stores user info, including pets up for adoption and pets users are interested in.

Project Scheduling



01

Start

Setting a hypothetical start date of December 1st, 2022

06

End

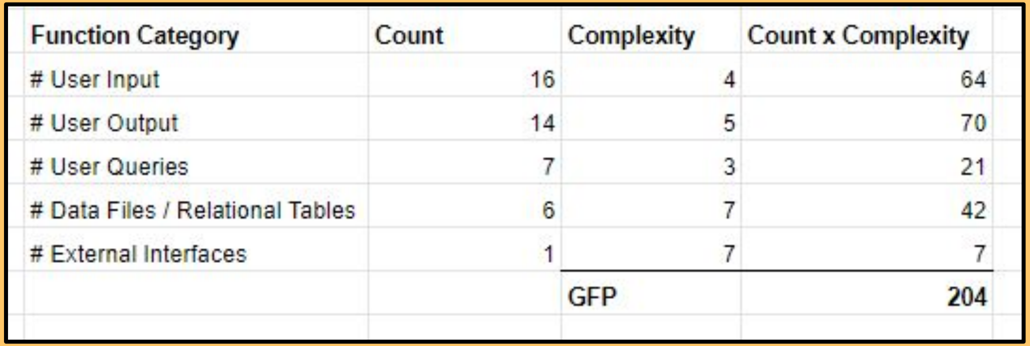
Estimating the development timeline to be about 6 months, based off similar app Tinder

The uncertainty would range from 2x the estimated time to 0.5x the estimated time, which would mean 3-12 months, so any time from February 2023 to December 2023.

Weekends are not counted in this schedule, and we will be working 8 hours per workday

Cost, Effort, & Pricing

Using Function Point (FP) Method:

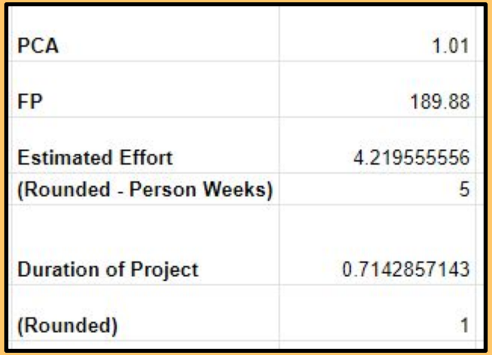


Function Category	Count	Complexity	Count x Complexity
# User Input	16	4	64
# User Output	14	5	70
# User Queries	7	3	21
# Data Files / Relational Tables	6	7	42
# External Interfaces	1	7	7
		GFP	204

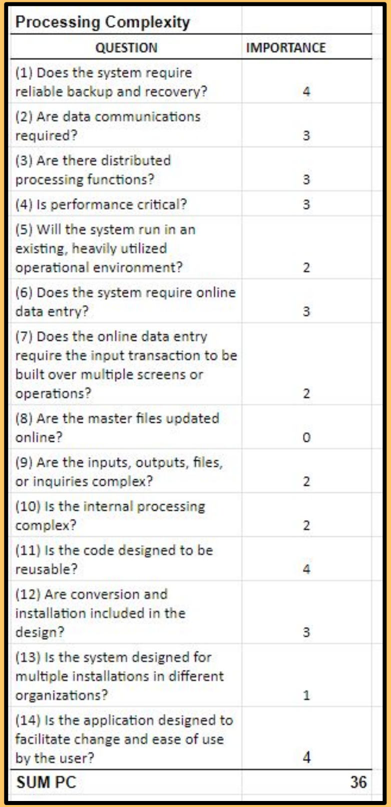
We will have no hardware since we will be using Amazon Web Services to host the information of the app.

Amazon Web Service as a software will cost $13.51 per month.

For the cost of personnel, the current development team has 7 people.



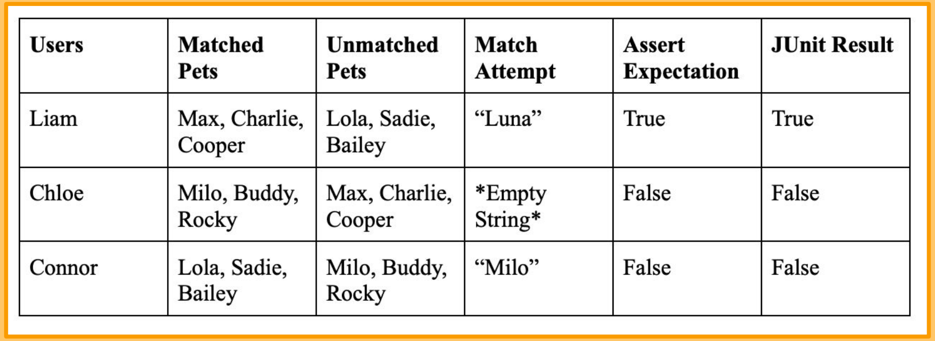
PCA	1.01
FP	189.88
Estimated Effort	4.219555556
(Rounded - Person Weeks)	5
Duration of Project	0.7142857143
(Rounded)	1



Processing Complexity	
QUESTION	IMPORTANCE
(1) Does the system require reliable backup and recovery?	4
(2) Are data communications required?	3
(3) Are there distributed processing functions?	3
(4) Is performance critical?	3
(5) Will the system run in an existing, heavily utilized operational environment?	2
(6) Does the system require online data entry?	3
(7) Does the online data entry require the input transaction to be built over multiple screens or operations?	2
(8) Are the master files updated online?	0
(9) Are the inputs, outputs, files, or inquiries complex?	2
(10) Is the internal processing complex?	2
(11) Is the code designed to be reusable?	4
(12) Are conversion and installation included in the design?	3
(13) Is the system designed for multiple installations in different organizations?	1
(14) Is the application designed to facilitate change and ease of use by the user?	4
SUM PC	36

Test Plan

Link to Github containing code: https://github.com/PetPal-TP/3354-PetPal/tree/main/addToMatchedPets



Users	Matched Pets	Unmatched Pets	Match Attempt	Assert Expectation	JUnit Result
Liam	Max, Charlie, Cooper	Lola, Sadie, Bailey	“Luna”	True	True
Chloe	Milo, Buddy, Rocky	Max, Charlie, Cooper	*Empty String*	False	False
Connor	Lola, Sadie, Bailey	Milo, Buddy, Rocky	“Milo”	False	False

Testing: **addToMatchedPets** method

- This method is called when a user adopting a pet swipes right on the pet browser and adds the selected pet to their list of matched pets in our user database.
- For the test, we created dummy users with lists of matched and unmatched pets. We then attempted to add three different pets to the dummy users. One user matched with a pet that was not present in their unmatched pets, the second user matched with a pet that had an invalid, empty name, and the final user was matched with a pet that was in their unmatched pets list.
- The only user that is expected to have the pet added is Liam since they are given the only valid pet name that is not present on their unmatched pets list

Similar Designs

We found three other applications that are similar to ours.

PetFinder:

- Has a Tinder-like feature
- Communication is facilitated through email
- Users can put their own pet up for adoption

WeRescue:

- Includes a social media section for success stories
- Have a “Learn” section to learn about pet adoption, pet caretaking, etc
- Prompt users to contact the shelter to facilitate adoption

Petdar:

- Prompts users to call or email the pet store

Though our app was similar to PetFinder, our design provides the user with in-app messaging, hopefully leading to a higher retention rate among adoptions

Conclusion

Though our initial idea for the project did not change, some of the implementation decisions were.

Software process model:

- Used spiral as it was agreed to be the most effective for this project

Architecture:

- Used Client-server though pipe/filter and MVC were discussed

Though we were aware of similar applications, we believe our application has the potential to be more effective at retaining adoptions of pets. The main improvement over the closest application, Petfinder, is the in-app messaging feature allowing potential adopters to more easily connect with potential pets.

Sources

[1] J. Allman, “Petdar,” *App Store*, 25-Mar-2015. [Online]. Available: https://apps.apple.com/us/app/petdar/id975535349. [Accessed: 18-Nov-2022].

[2] L. Levy, “5 Ways To Optimise Your App For High User Retention,” *Usability Geek*, 04-Sep-2019. [Online]. Available: https://usabilitygeek.com/optimise-app-user-retention/. [Accessed: 18-Nov-2022].

[3] Nestle, “Petfinder - Adopt a Pet,” *App Store*, 07-Nov-2012. [Online]. Available: https://apps.apple.com/us/app/petfinder-adopt-a-pet/id557228073. [Accessed: 18-Nov-2022].

[4] Pet Ventures LLC, “WeRescue – Adopt a Pet,” *App Store*, 14-Nov-2014. [Online]. Available: https://apps.apple.com/us/app/werescue-adopt-a-pet/id918610594. [Accessed: 18-Nov-2022].

[5] Schweb Design LLC, “Lancaster Puppies,” *App Store*, 03-Apr-2020. [Online]. Available: https://apps.apple.com/us/app/lancaster-puppies/id1478908590. [Acc
18-Nov-2022].

Thanks!

Questions?

CREDITS: This presentation template was created by **Slidesgo**, and includes icons by **Flaticon** and infographics & images by **Freepik**