

CS3354 Software Engineering
Final Project Deliverable 2

PetPal: “Tinder for Pet Adoption”

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

1. [5 POINTS] Well described delegation of tasks, i.e. who did what in the project. Now that your project is complete, you are required to submit the delegation of tasks from beginning of the project until the end. Please make sure to fairly distribute tasks in the team and remember that at the end of the semester, each team member will receive the same grade. See grading policy below for more detail. If no/poor contribution by a member, please specify clearly so that we can grade each student fairly

Tasks delegated to each member:

- Nathaniel Brown
 - First repository commit (the README) for deliverable 1
 - Architectural design for deliverable 1
 - Creating Google Docs for team collaboration for Proposal and Deliverables,
- Noah Acosta
 - Commit with “Project Scope” to GitHub for deliverable 1
 - References for deliverable 2
 - Compilation of Deliverables
 - Submission of Deliverables
 - Address feedback given from the project proposal for deliverable 1
 - Final commits to GitHub (Slides, Deliverable 2, Unit test Code) for deliverable 2
 - Aid in Software requirements for deliverable 1
 - Delegation of tasks for deliverable 2
- Lucas Noack
 - Sequence Diagram for deliverable 1
 - Test Plan and unit code for software for deliverable 2
- Tera Parish
 - Add team members and TA to GitHub repository for deliverable 1
 - Use-Case Diagram for deliverable 1
 - Comparison of work with similar designs for deliverable 2
- Rachel Bang
 - Class Diagram
 - Slides
- Aidan Duong
 - Conclusion
 - Slides
- Ritvik Chilakamarthy
 - Describe software process model for deliverable 1
 - Software Requirements for deliverable 1
 - Project Scheduling, cost, effort, and pricing estimation for deliverable 2

2. [10 POINTS] Project Deliverable 1 content

Please attach here the Final Project draft description (that contains the instructor feedback). It is ok to include a picture of the original document. Address the feedback provided for your proposal by listing what you did / plan to do to comply with those proposed changes and or requests for additions to your project.

Our project can be summed up in one phrase: “Tinder/Bumble/okCupid for pet adoption.” We’re going to create a mobile app that shows users potential pets that need adopting in their area in sequence, and the user will be able to provide feedback via swipes or (hearts/stars) to potentially connect with the shelters that are housing them. The owners and potential adopted pets will each have profiles—for example, someone might want a lazy dog, while others want a high-energy pooch. We will use this information to try to match the best pet with the right owner. For the pet side, we want to support both shelters and individual people to list their pets on our app. We would like to support the adoption of any domesticated animal, including but not limited to dogs, fish, ferrets, cats, lizards, spiders...We will include a messaging system to get potential adopters in touch with pet caretakers in the app itself.

For too long, pet adoption has been a convoluted and abstract process. This complexity has prevented people from connecting with the animals they could give a valuable life. This causes more people to buy pets from breeders, rather than adopting one of the many pets in need. By offering a location-based matching service, users are able to both give away and take in a beloved pet from their local area. We hope that our app will prevent any further animals from being euthanized in shelters (because of never being adopted), abandoned by owners who no longer want them, and so much more.

We expect our app to be used primarily by potential adopters throughout the country, which includes a broad range of demographic groups including first-time homeowners and the elderly. We want to design it well enough that it is a well-known name for pet adoption needs.

Based off the feedback given to us by our professor we are tasked with differentiating our app from those that are already available via App Store/Google Store. Our app is different in that we are taking into consideration ALL possible pets, not just cats and dogs. Our app will allow for private pet owners to place their pets up for adoption as well (after undergoing a thorough background check). Current apps only allow for shelters to advertise their pets however, we understand that it can be heartbreaking to send a loved one to a shelter, thus our app will ease that heart break.

Setting up a Github repository. Please use your utdallas email accounts only for each group member.

GitHub url: <https://github.com/PetPal-TP/3354-PetPal>

Members:

Tera, username PetPal-TP, is the repository holder for the group, which is why she is not on this list.

Manage access

Add people

<input type="checkbox"/>	Select all	Type ▾
<input type="text"/> Find a collaborator...		
<input type="checkbox"/>	 AidanD21 Collaborator	Remove
<input type="checkbox"/>	 Nate Brown natesabrown • Collaborator	Remove
<input type="checkbox"/>	 Noah Acosta PetPal-NA • Collaborator	Remove
<input type="checkbox"/>	 RachelBang Collaborator	Remove
<input type="checkbox"/>	 rvc200000 Collaborator	Remove
<input type="checkbox"/>	 Subash-kolluru Collaborator	Remove

Delegation of tasks: Who is doing what. If no contribution, please specify as it will help us grade each group member fairly.

Tasks delegated to each member:

- Nathaniel Brown
 - First repository commit
 - Address feedback from project proposal
 - architectural design
- Noah Acosta:
 - Commit with “project scope” to GitHub
 - References
 - In charge of Submission and communication with professor.
- Lucas Noack
 - Sequence diagram
 - Test plan for software
- Tera Parish:
 - Add team members and TA
 - Create use-case diagram
 - Comparison of work with similar designs
- Rachel Bang
 - Class Diagram
 - Slides
- Aidan Duong
 - Estimated cost of hardware/software products, & estimated cost of personnel.
 - Conclusion
 - Slides
- Ritvik Chilakamarthy
 - Describing software process model employed in our project
 - List of software requirements (functional & non-functional)
 - Project Scheduling
 - Cost, Effort, & Pricing Estimation

Which software process model is employed in the project and why. (Ch 2)

The software process model to be employed in this project is the spiral model. This is the best software model to use for this project, because at the beginning stages of our project, the specific requirements are not known at this time. The spiral model better allows us to understand, react, and adapt to changing requirements and risks. The spiral model most closely reflects how development progresses in the real world, so it would also be easiest for us to implement. We would constantly be reviewing what was already completed for the project by observing anchor point milestones, so we'd always be on top of the requirements and risks. As the group keeps iterating through multiple cycles, our project will progressively become more and more complete, building upon previous iterations. A proper application of the spiral model will result in the best completed product.

Software Requirements including

a.) Functional requirements.

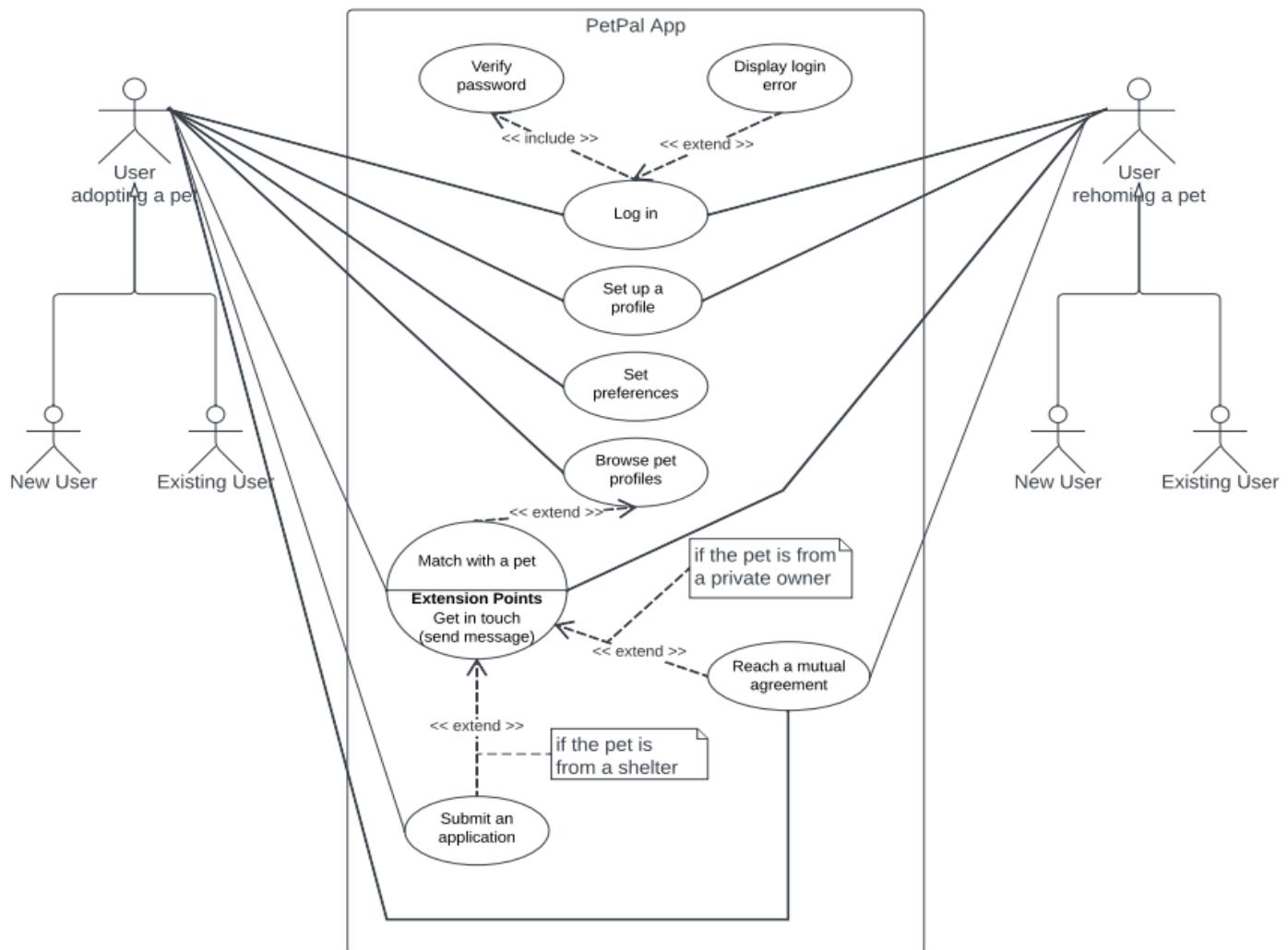
- A user shall be able to create a profile for themselves or their pets, containing relevant information about themselves or their pets
- A user shall be able to message any other user they ‘match’ with
- A user shall be able to select requirements to specify what qualities they wish to look for in their ideal match
- A user shall be able to browse other users which match their requirement and view these other users’ profiles.
- A user shall be able to ‘swipe left’ to reject a candidate, or ‘swipe right’ to accept a candidate
- A user shall be able to view other users who ‘swiped right’ on them, view their profile, and swipe back in return.

b.) Non-functional requirements

- Usability Requirement: 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.
- Performance Requirement:
- Space Requirement: The application shall not take up more than 1GB of space.
- Dependability Requirement: The software development team shall continuously monitor the application to fix all bugs or errors as soon as possible.
- Security Requirement: The application will store passwords in a secure location to avoid access by hostile entities.
- Environmental Requirement: 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 (e.g., it should warn that a Husky should not live in a hot area).
- Operational Requirement: The application must be able to perform all actions requested by the user

- Development Requirement: The software must be programmed in a high-level programming language and implement a database management system such as SQL.
- Accounting Requirement: The application must include a security fail-safe to protect user's account
- Safety/Security Requirement: 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

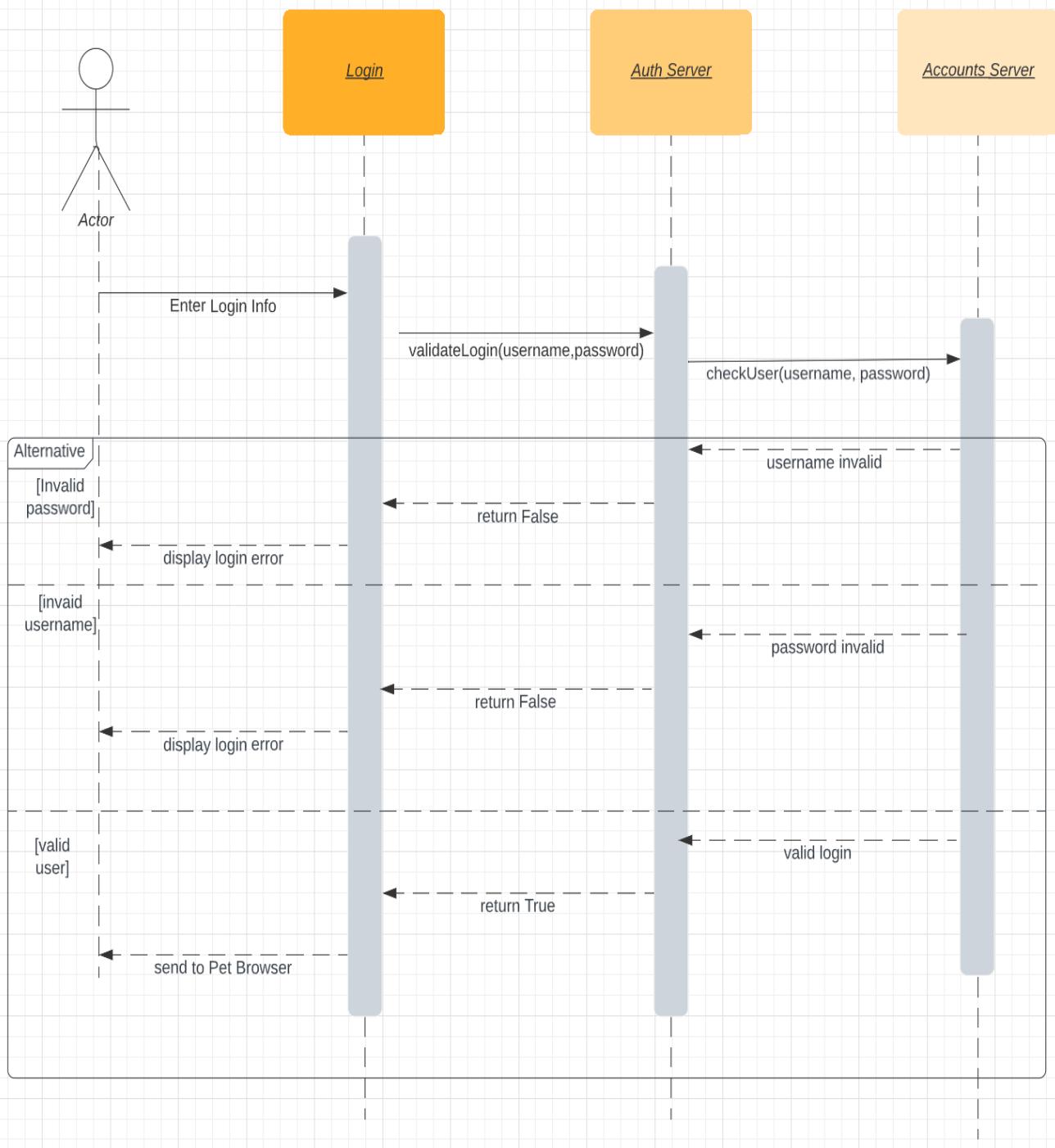
Use case diagram – Provide a use case diagram (similar to Figure 5.5) for your project. Please note than there can be more than one use case diagrams as your project might be very comprehensive. (Ch 5 and Ch 7)



Sequence diagram

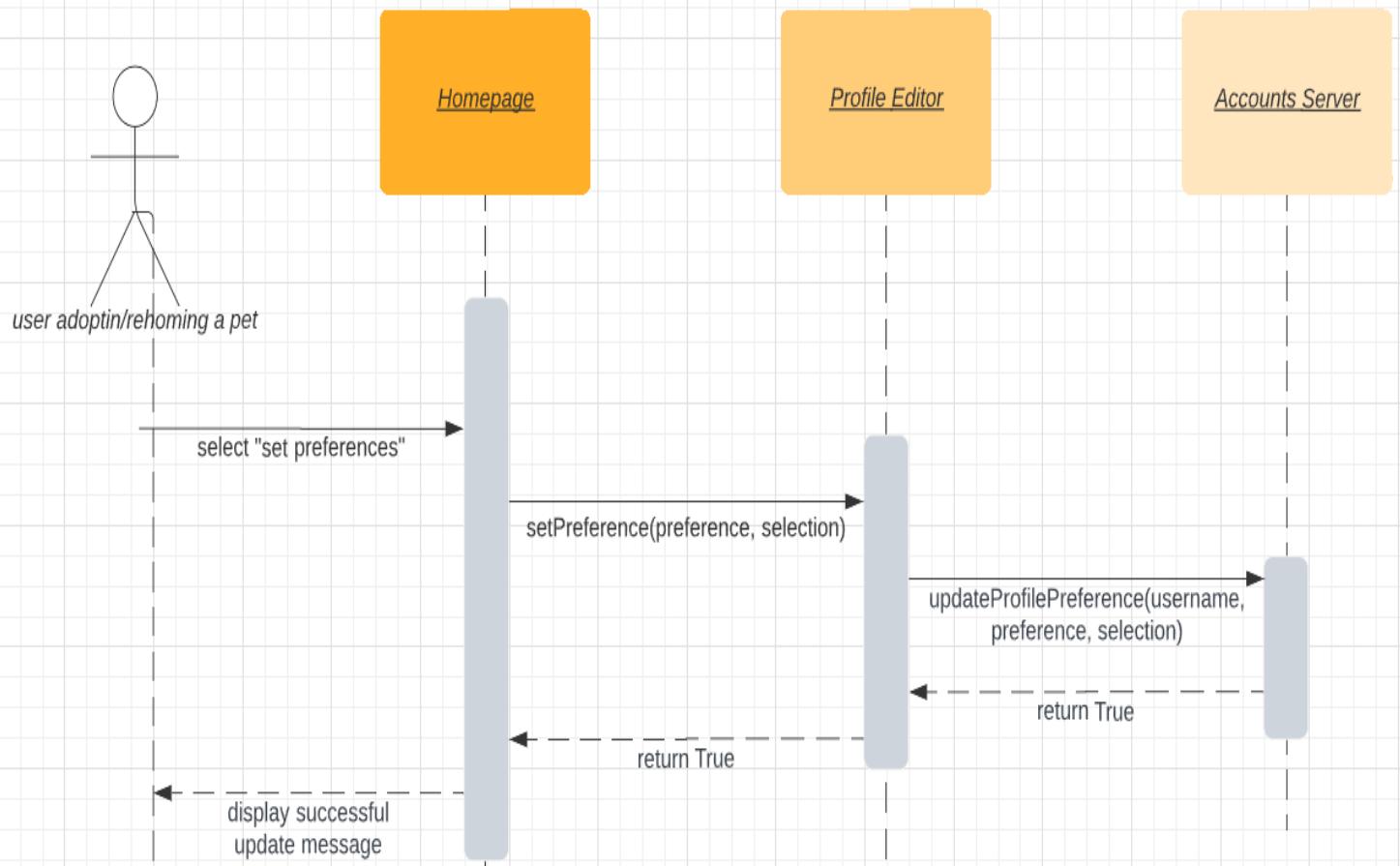
Log-in

Lucas Noack | October 20, 2022



Set preferences

Lucas Noack | October 20, 2022



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)

return False

doesUserExist(username)

return True

return False

return True

validatePassword(password)

return False

return True

return True

addNewUser(userInfo)

return True

display successful creation message

Alternative

[username already exists]

display login error

[Else]

Alternative

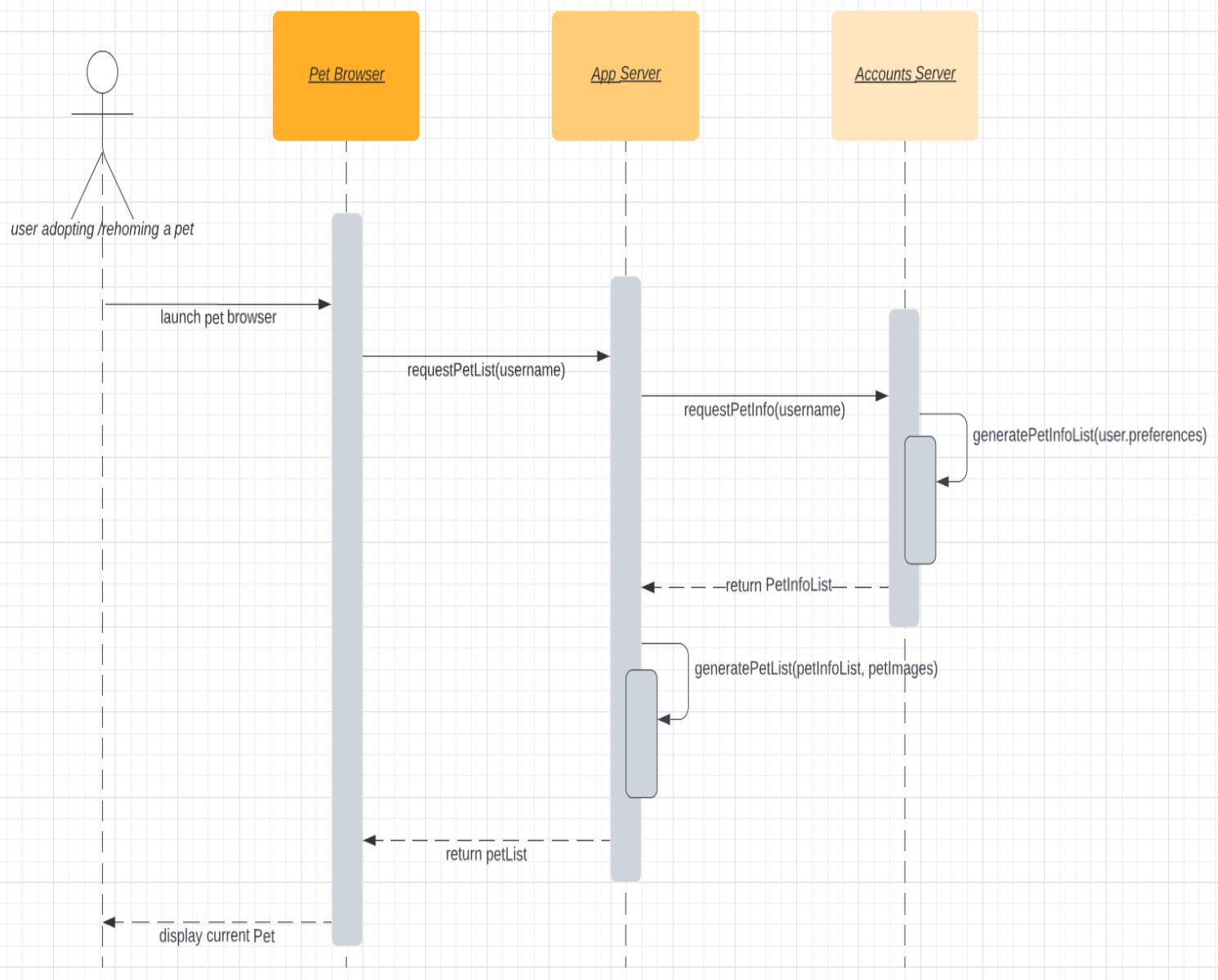
[password invalid]

display login error

[Else]

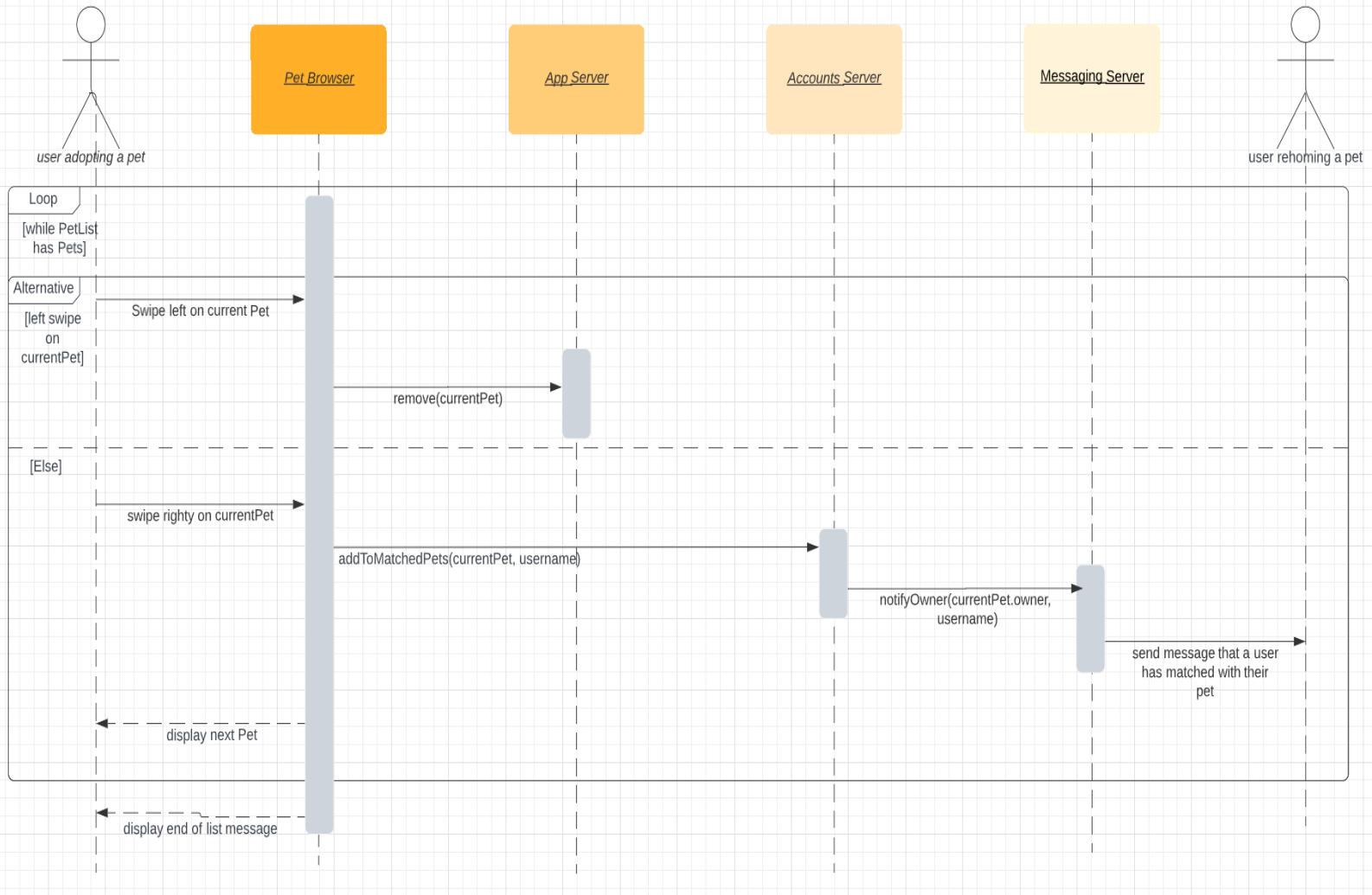
Browse Pet Profiles

Lucas Noack | October 21, 2022



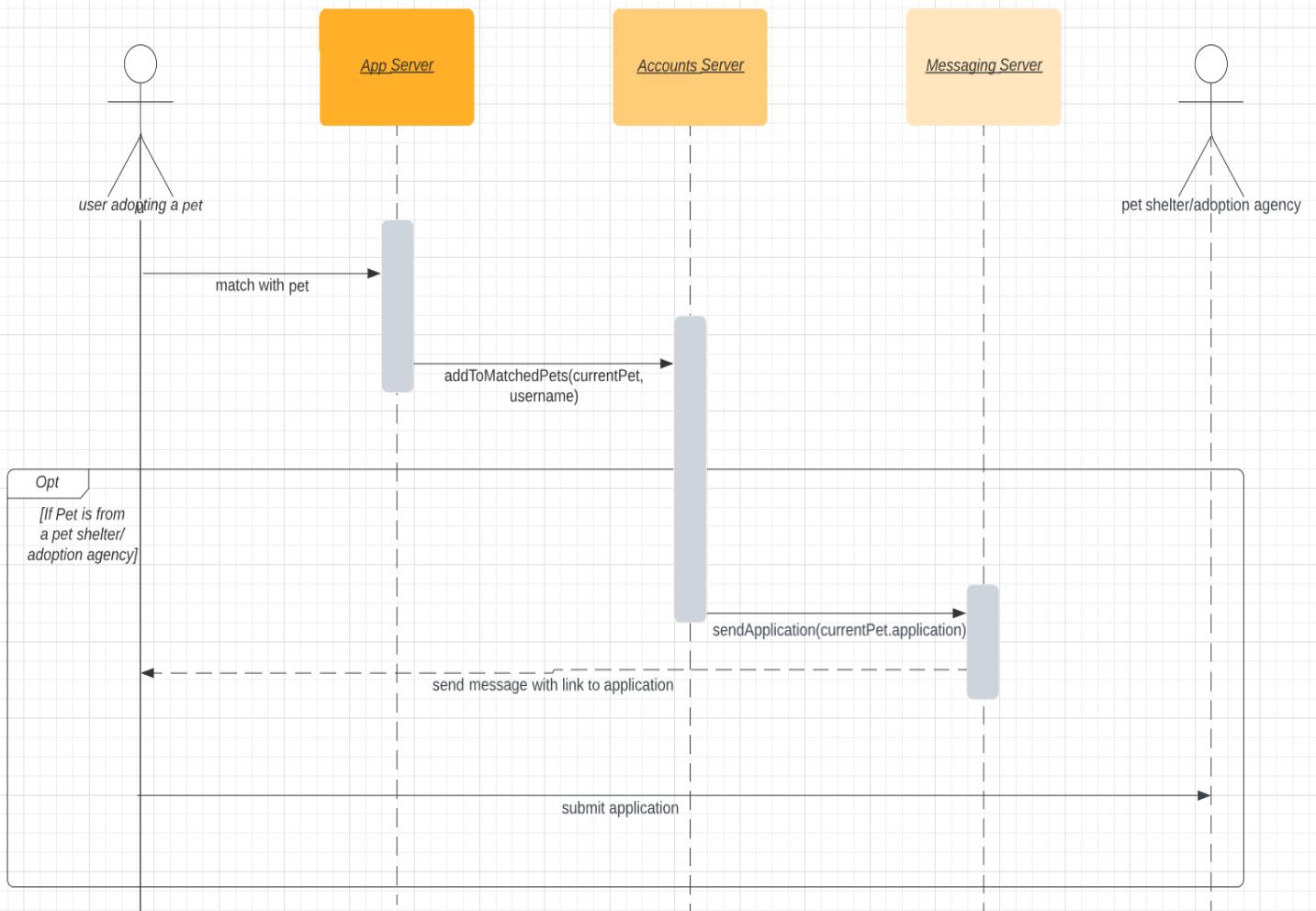
Match With a Pet

Lucas Noack | October 21, 2022



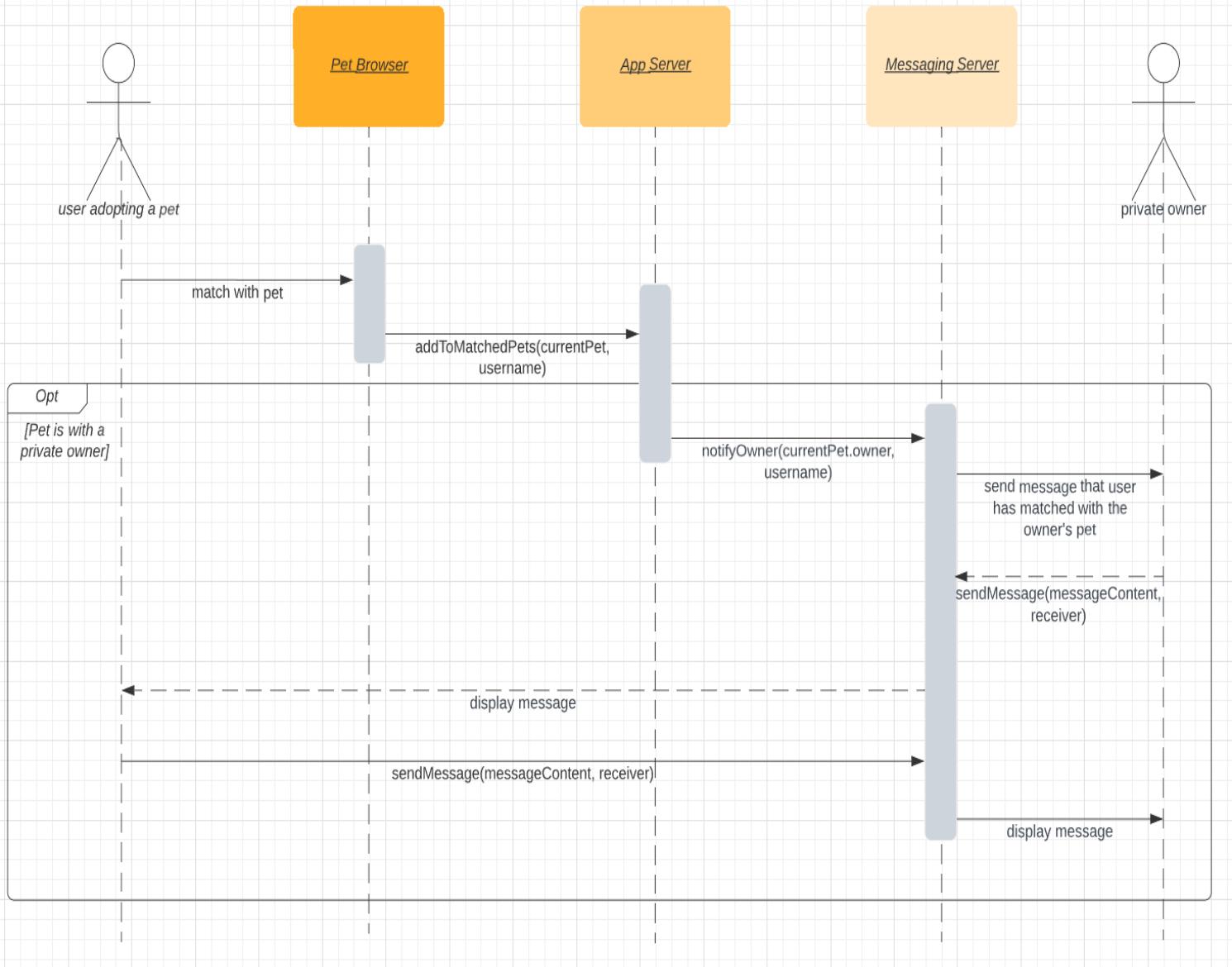
Submit Application

Lucas Noack | October 21, 2022

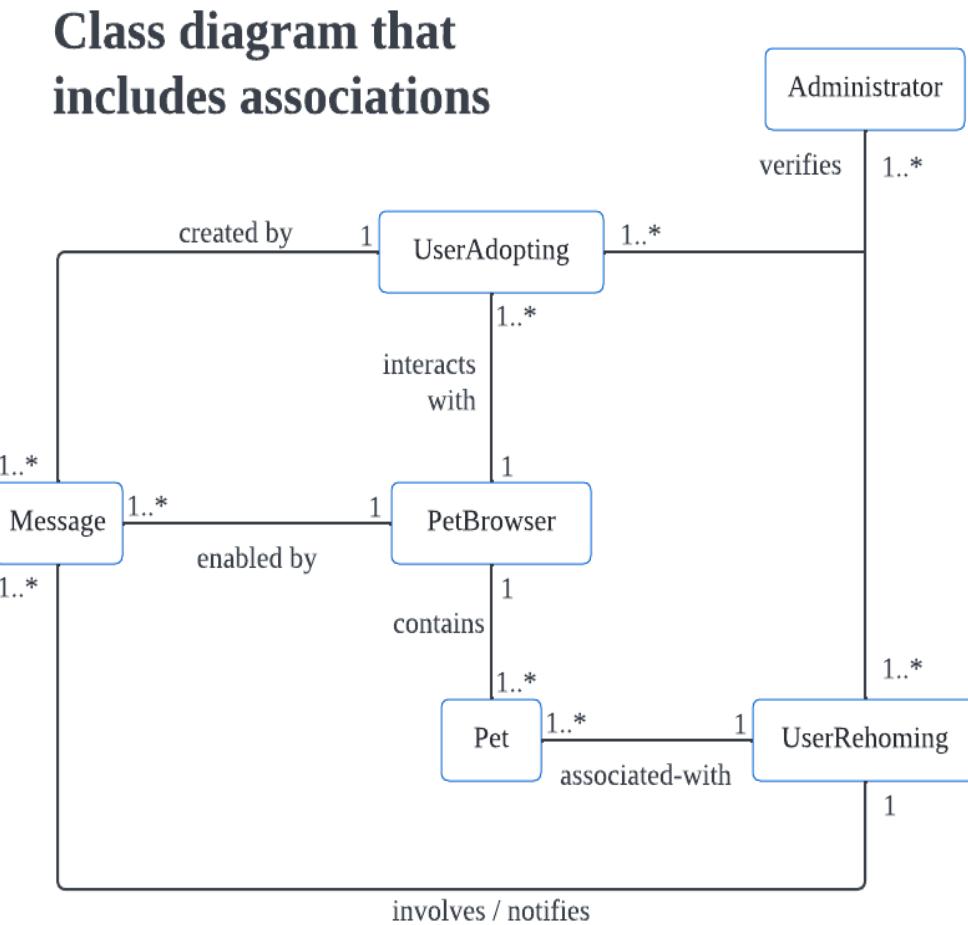


Reach a Mutual Agreement

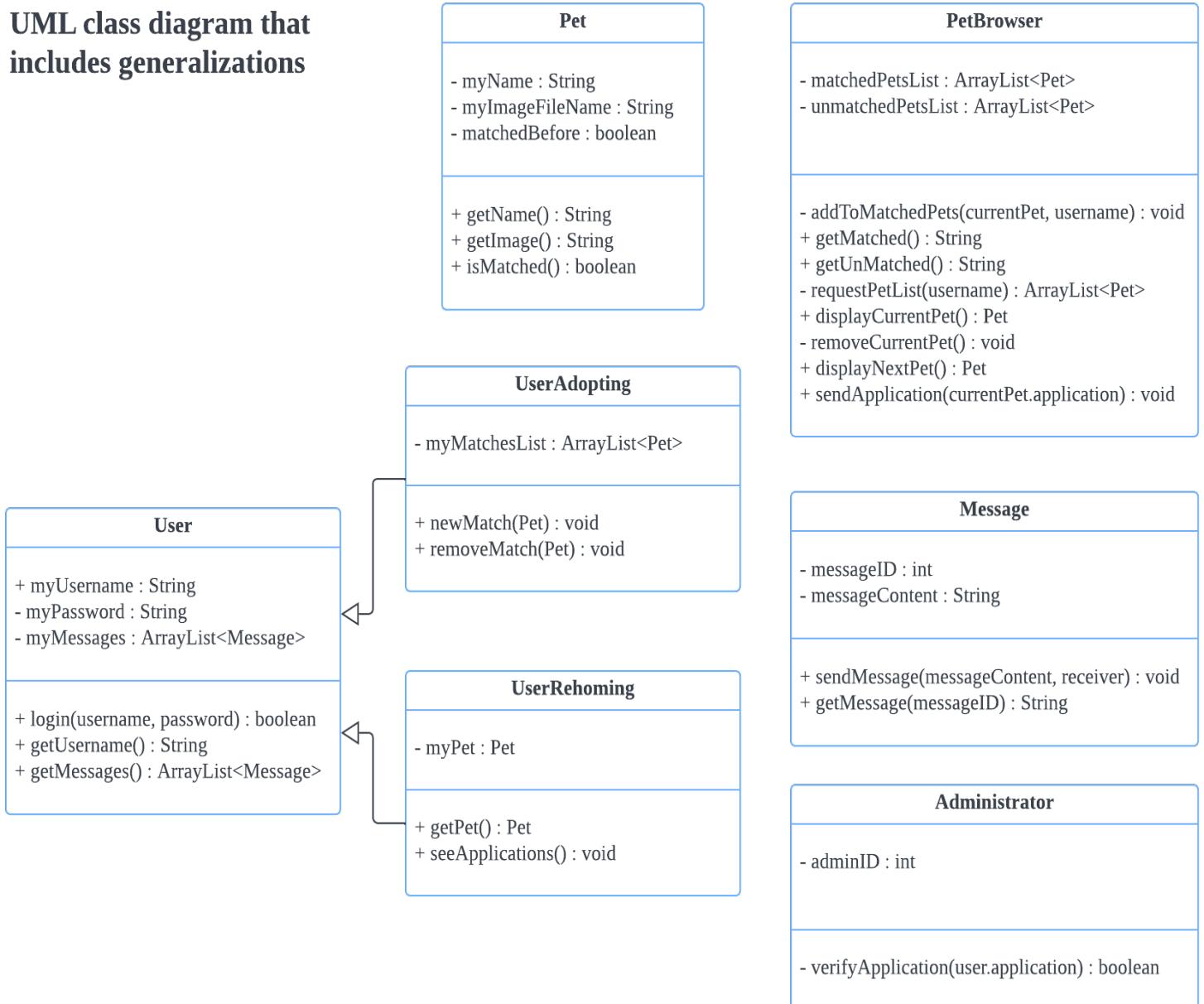
Lucas Noack | October 21, 2022



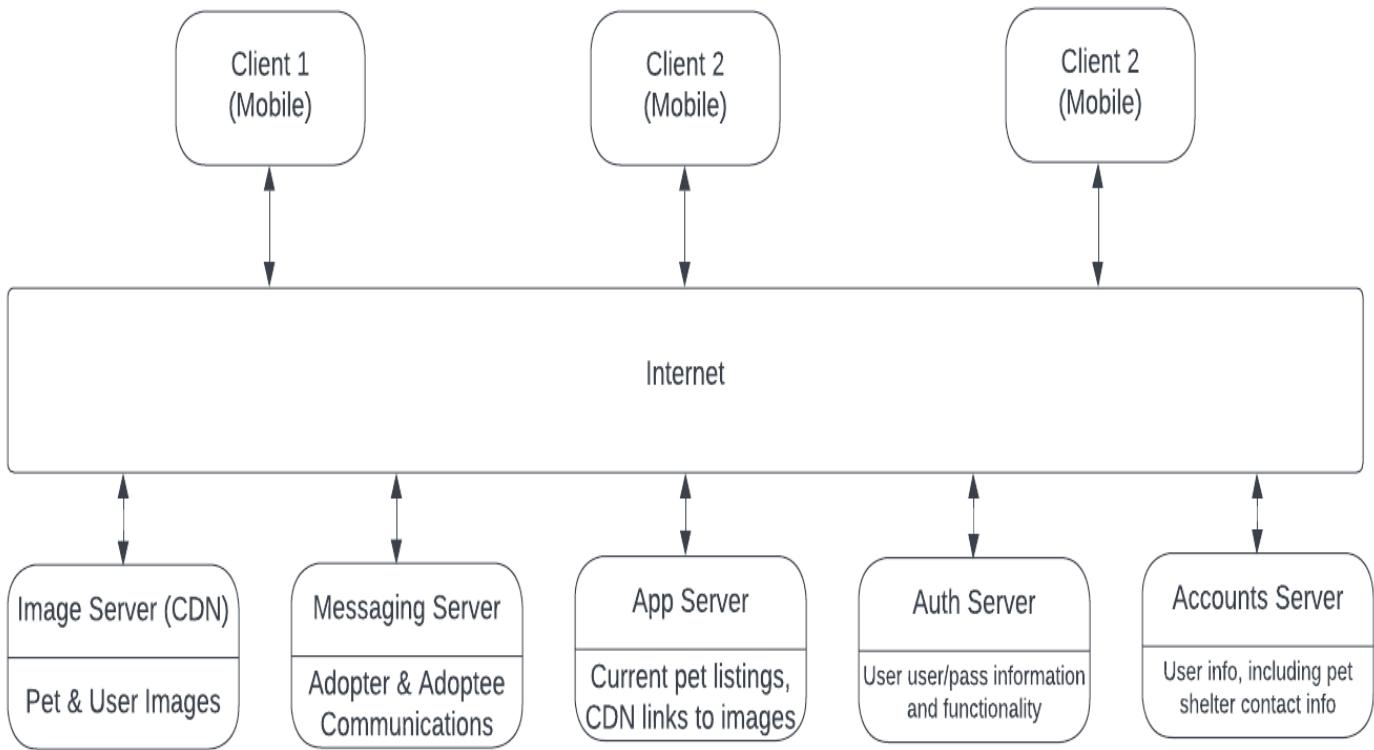
Class diagram



UML class diagram that includes generalizations



Architectural design: Client-server architecture pattern



Our group concluded that the client-server architecture will work best for our product. We can elaborate on the advantages of separating each server out:

- **Image Server (CDN):** Server configured as a Content-Delivery Network (CDN) specifically for serving lots of images to users. That way users can 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 serve 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.

3. [35 POINTS] Project Scheduling, Cost, Effort and Pricing Estimation, Project duration and staffing: Include a detailed study of project scheduling, cost and pricing estimation for your project. Please include the following for scheduling and estimation studies:
- 3.1. [5 POINTS] Project Scheduling. Make an estimation on the schedule of your project. Please provide start date, end date by giving justifications about your estimation. Also provide the details for: - Whether weekends will be counted in your schedule or not - What is the number of working hours per day for the project.
- 3.2. [15 POINTS] Cost, Effort and Pricing Estimation. Describe in detail which method you use to calculate the estimated cost and in turn the price for your project. Please choose one of the two alternative cost modeling techniques and apply that only: - Function Point (FP) - Application composition.
- 3.3. [5 POINTS] Estimated cost of hardware products (such as servers, etc.)
- 3.4. [5 POINTS] Estimated cost of software products (such as licensed software, etc.)
- 3.5. [5 POINTS] Estimated cost of personnel (number of people to code the end product, training cost after installation)

The start date of our project will be December 1st, 2022. To estimate how many hours the project will take, we must first look at the development time of a similar app, Tinder. Tinder took approximately six months to develop, which is the timeframe we will use as our baseline for estimation. As we already know the feasibility, requirements, and design of the code, our time frame estimation is relatively simple. Using the estimate uncertainty graph, with the aspects of our project we already know, 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.

For the cost estimate, we will be using the function point 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

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

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

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.

4. [10 POINTS] A test plan for your software: Describe the test plan for testing minimum one unit of your software. As an evidence, write a code for one unit (a method for example) of your software in a programming language of your choice, then use an automated testing tool (such as JUnit for a Java unit) to test your unit and present results. Clearly define what test case(s) are provided for testing purposes and what results are obtained (Ch 8). Include your test code as additional document in your zip file submitted.

For our tested unit, we chose to test the 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. This data is represented here:

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

5. [10 POINTS] Comparison of your work with similar designs. This step requires a thorough search in the field of your project domain. Please cite any references you make.

Several mobile applications for pet adoption are available on the market, including Petfinder, WeRescue, and Petdar, to name a few. Here's a quick rundown of each application's features:

Petfinder [3]

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

WeRescue - Adopt a Pet [4]

- Include 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 - Find adoptable pets near you! [1]

- Prompt users to call or email the pet store.

Among those mentioned, our design is most like Petfinder, which is sponsored by the well-known pet food company Purina, and many other applications are powered by APIs on Petfinder. To set our app apart from the competition, especially Petfinder, we plan to have an "in-app" messaging feature, which can potentially lead to a higher retention rate for pet adopters.

6. **[10 POINTS]** Conclusion - Please make an evaluation of your work, describe any changes that you needed to make (if any), if things have deviated from what you had originally planned for and try to give justification for such changes

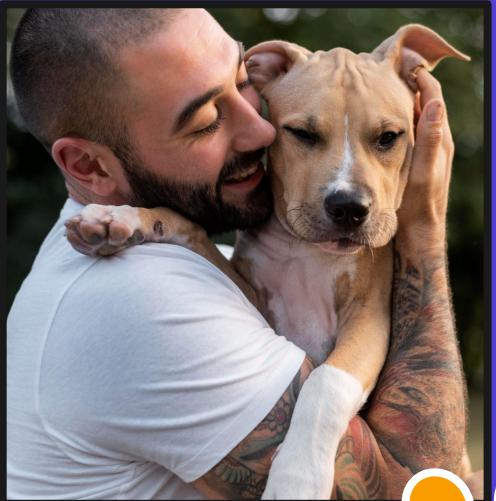
Though several different ideas were discussed throughout the process, our final version was thought to be ideal. There were not any significant changes to the initial idea. Our original idea was creating an app like Tinder which would let users swipe through pets and match with them to potentially adopt them. The first thing decided was the software process model. There wasn't much disagreement with this, and it was agreed upon that it would be the most effective. Next, we decided which architecture to use. Though most of our group decided on client-server architecture, other options such as pipe-filter and MVC were discussed. We decided to separate the servers into image, message, app, authentication, and account servers. Using the function point method, we estimated that project should take around 5-6 weeks. JUnit was used to test a method that added matched pests to a database. Though we initially knew that our application is like PetFinder, an app sponsored by Purina, we believe that the in-app messaging feature would lead to a higher retention rate with pets, a feature that PetFinder lacks.

7. [5 POINTS] References

- [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>. [Accessed: 18-Nov-2022].

8. **[10 POINTS]** Presentation slides.

Adopt



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.

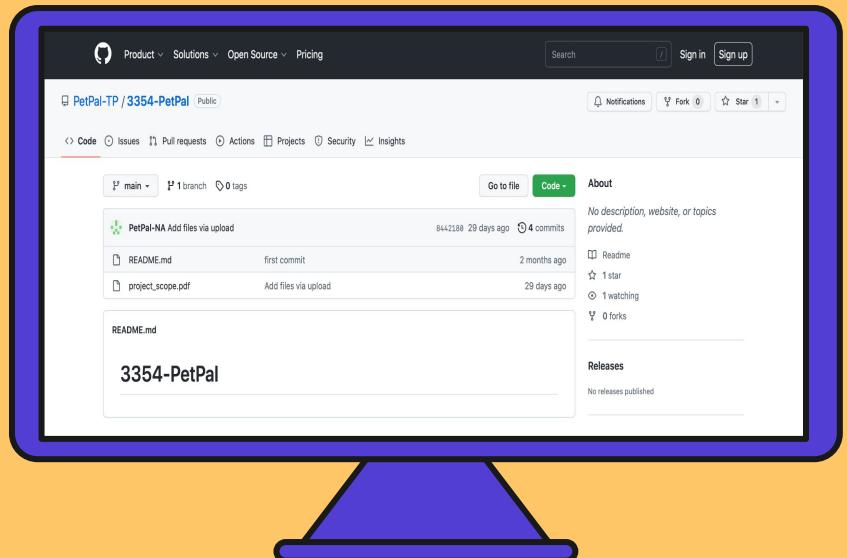


Personal

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 heartbreak, so we seek to be the bridge between two loving homes.

Our Github Repository

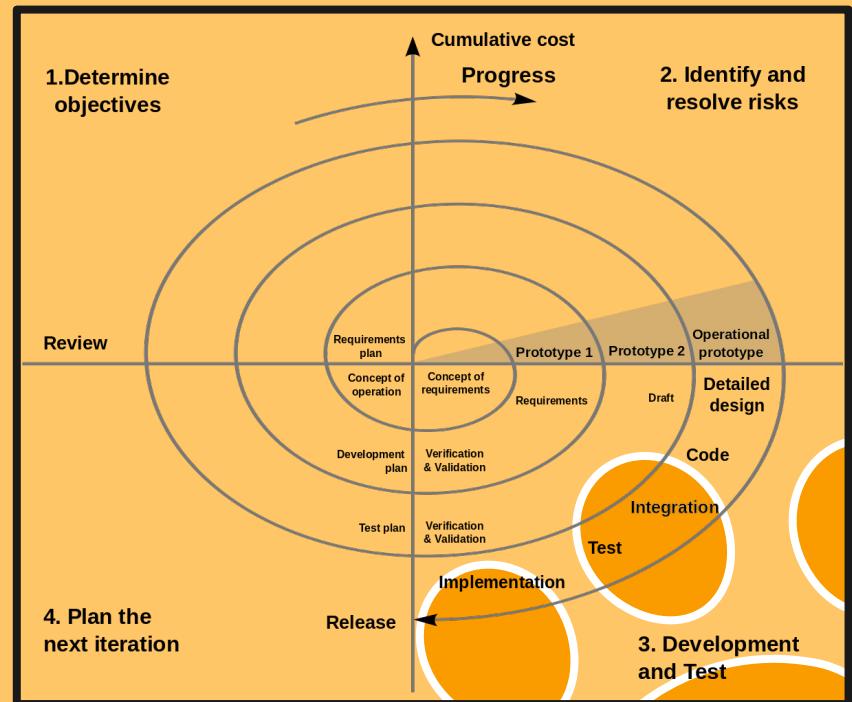
<https://github.com/PetPal-TP/3354-PetPal>



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

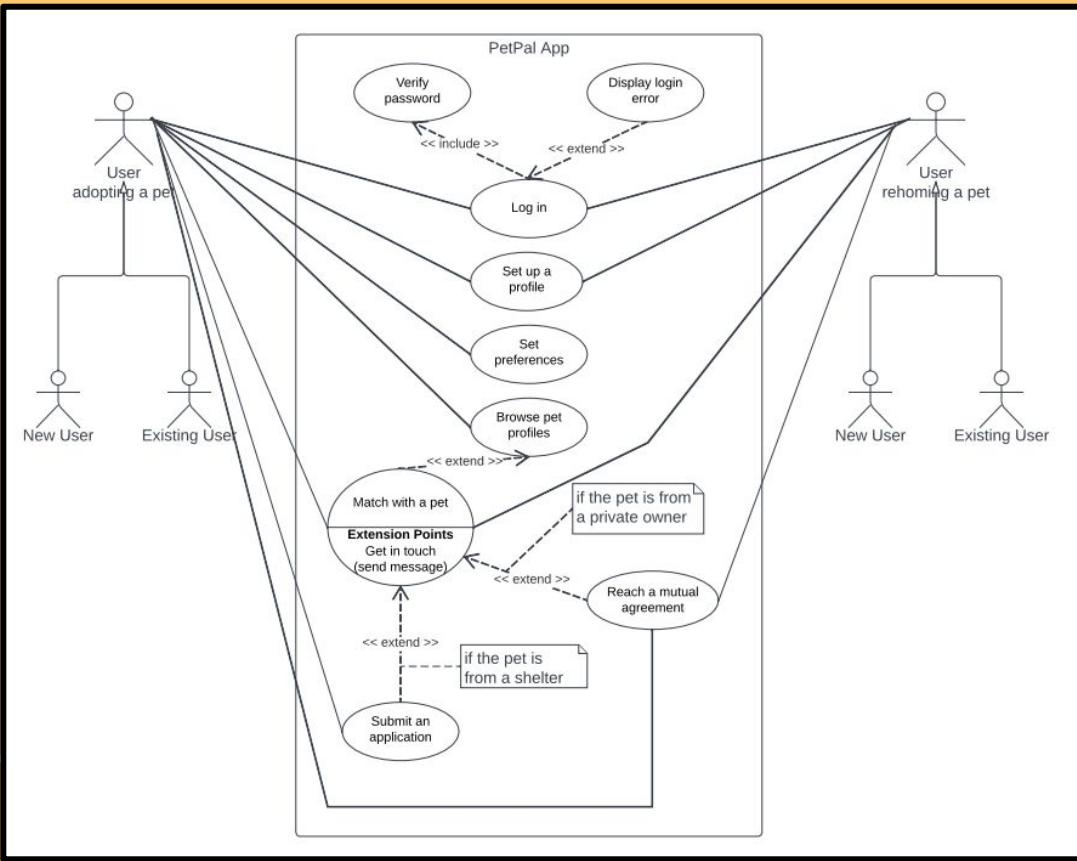
'Swipe left' to reject a candidate, or 'swipe right' to accept a candidate

View other users who 'swiped right' on them, view their profile, and swipe back in return

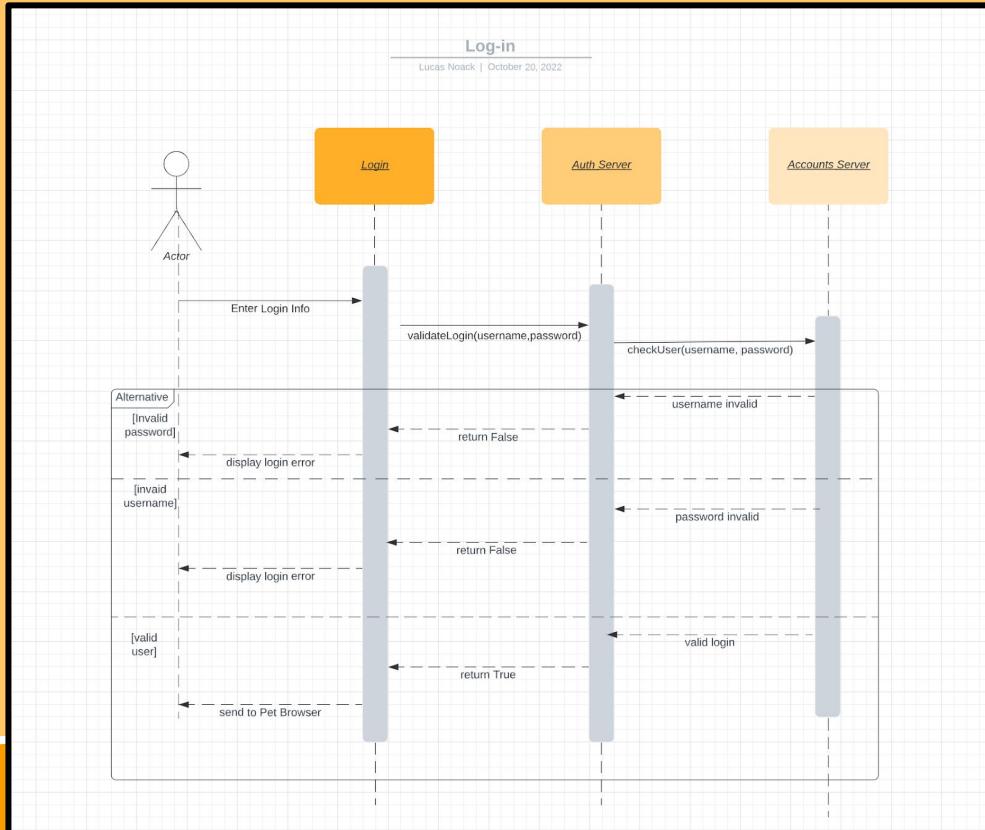
Non-functional Software Requirements

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

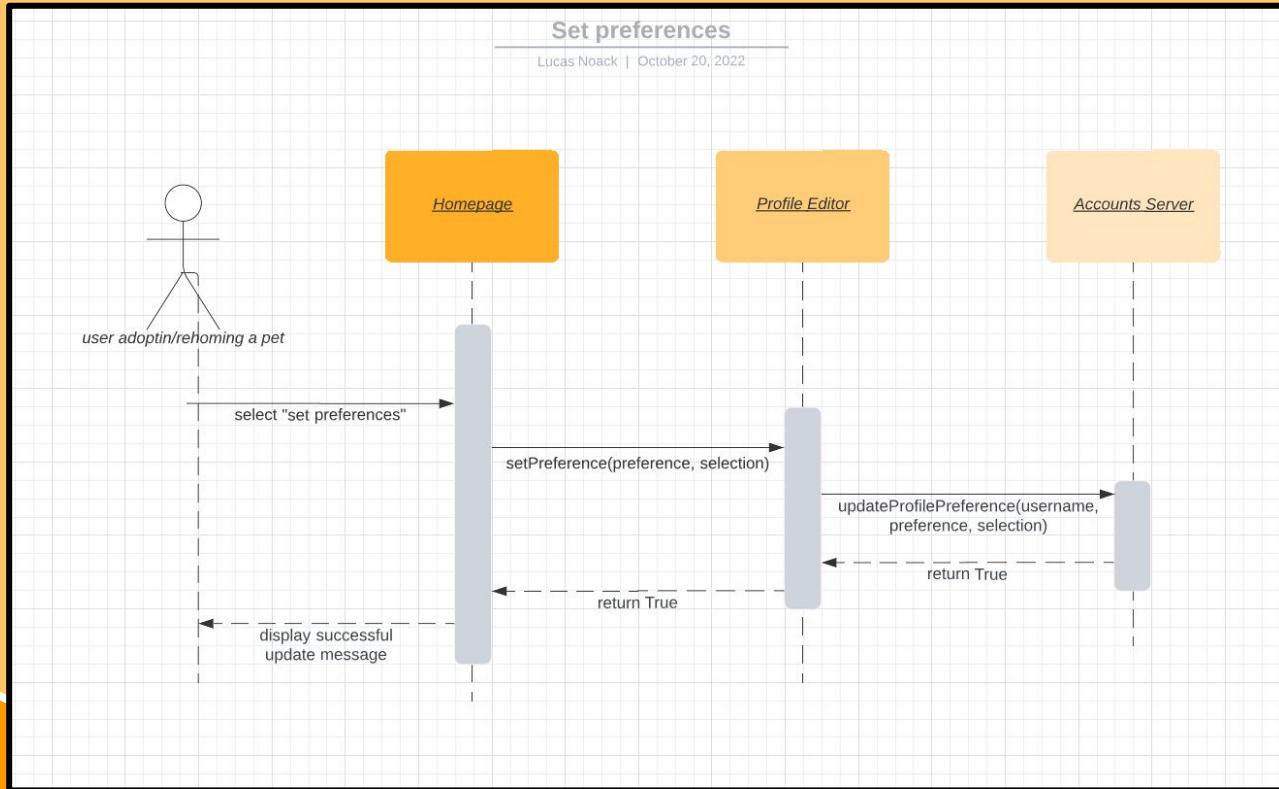
Use Case Diagram



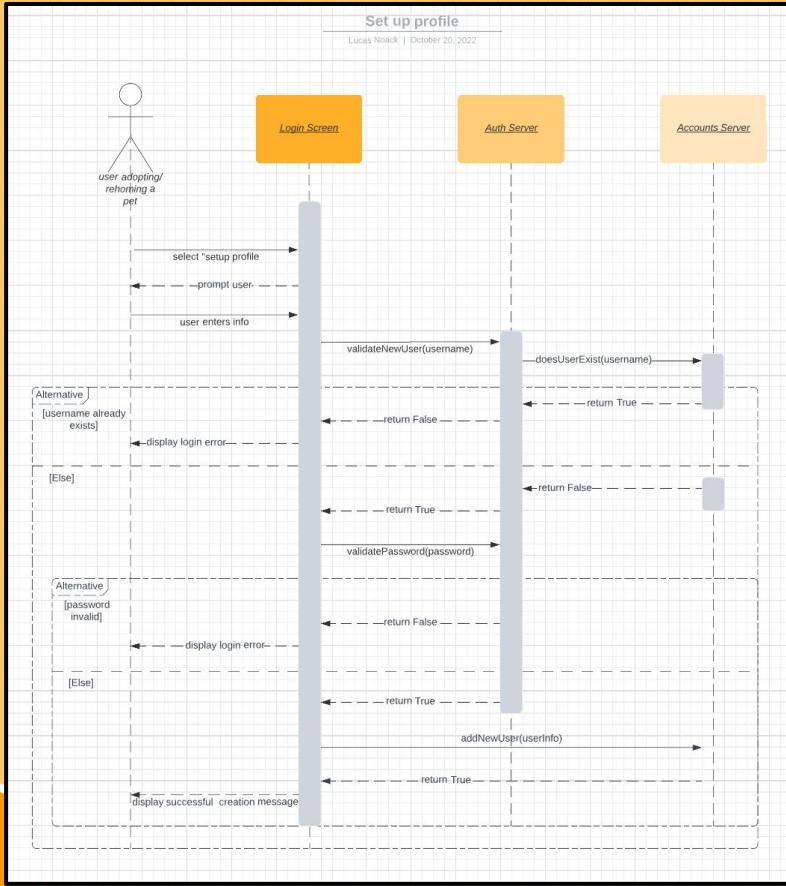
Sequence Diagram 1



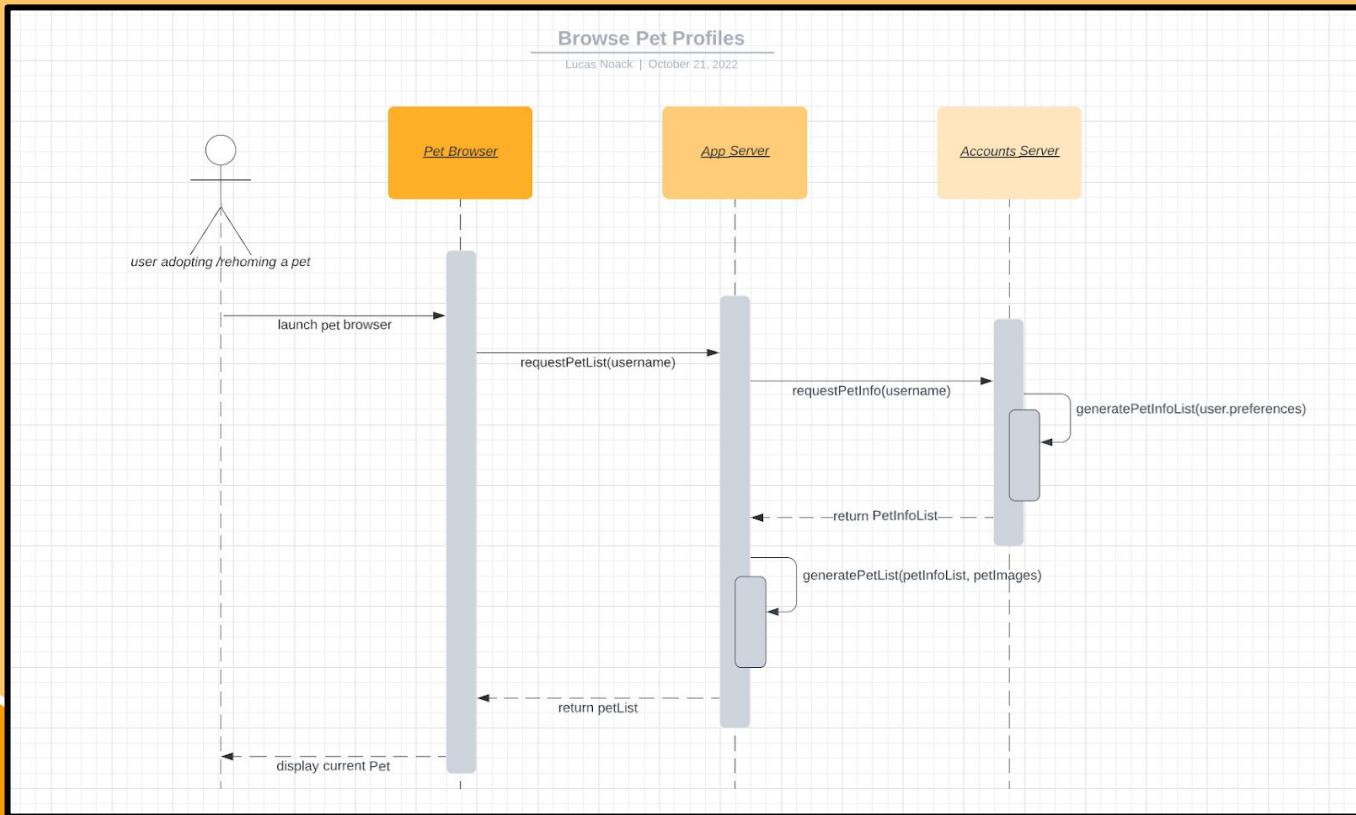
Sequence Diagram 2



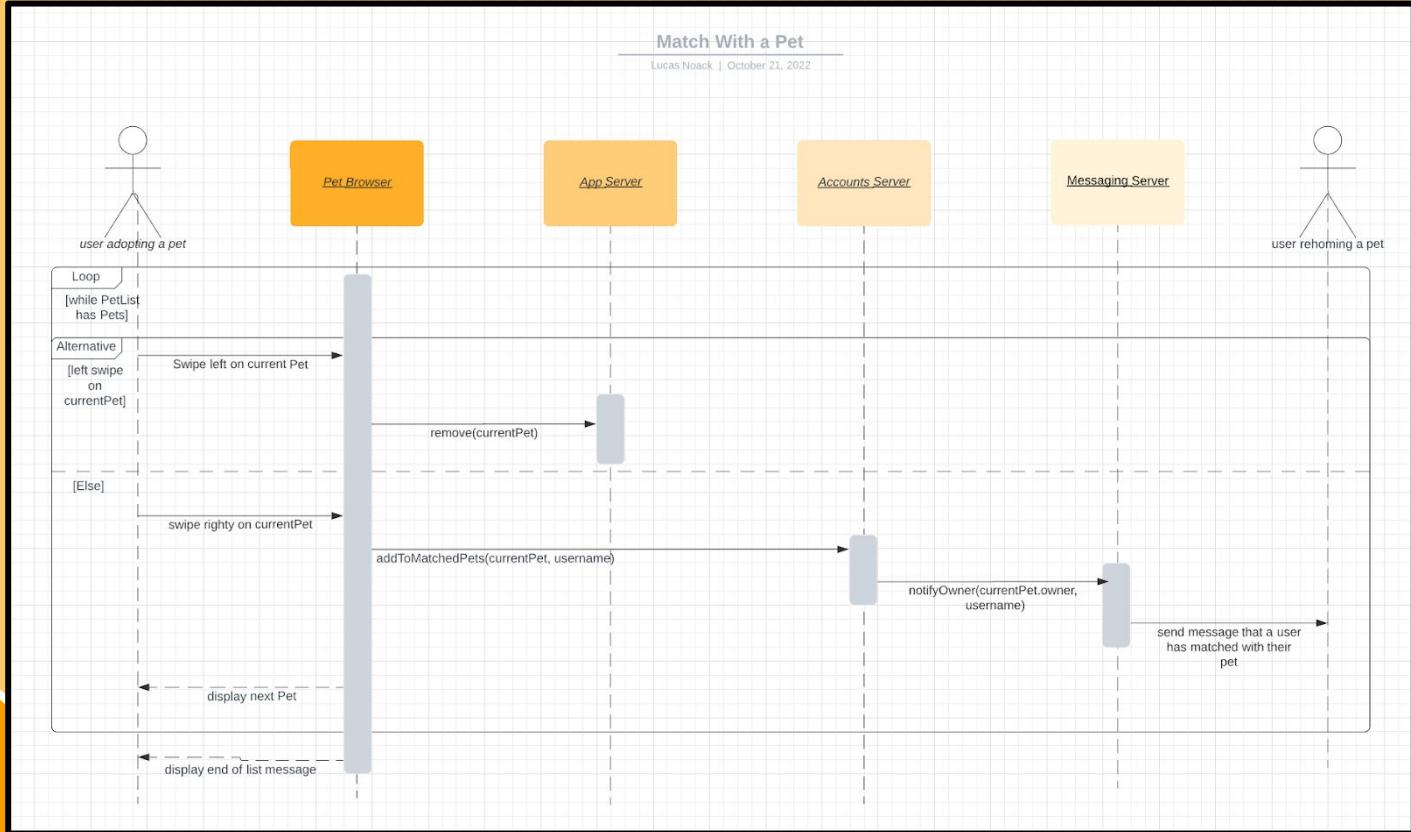
Sequence Diagram 3



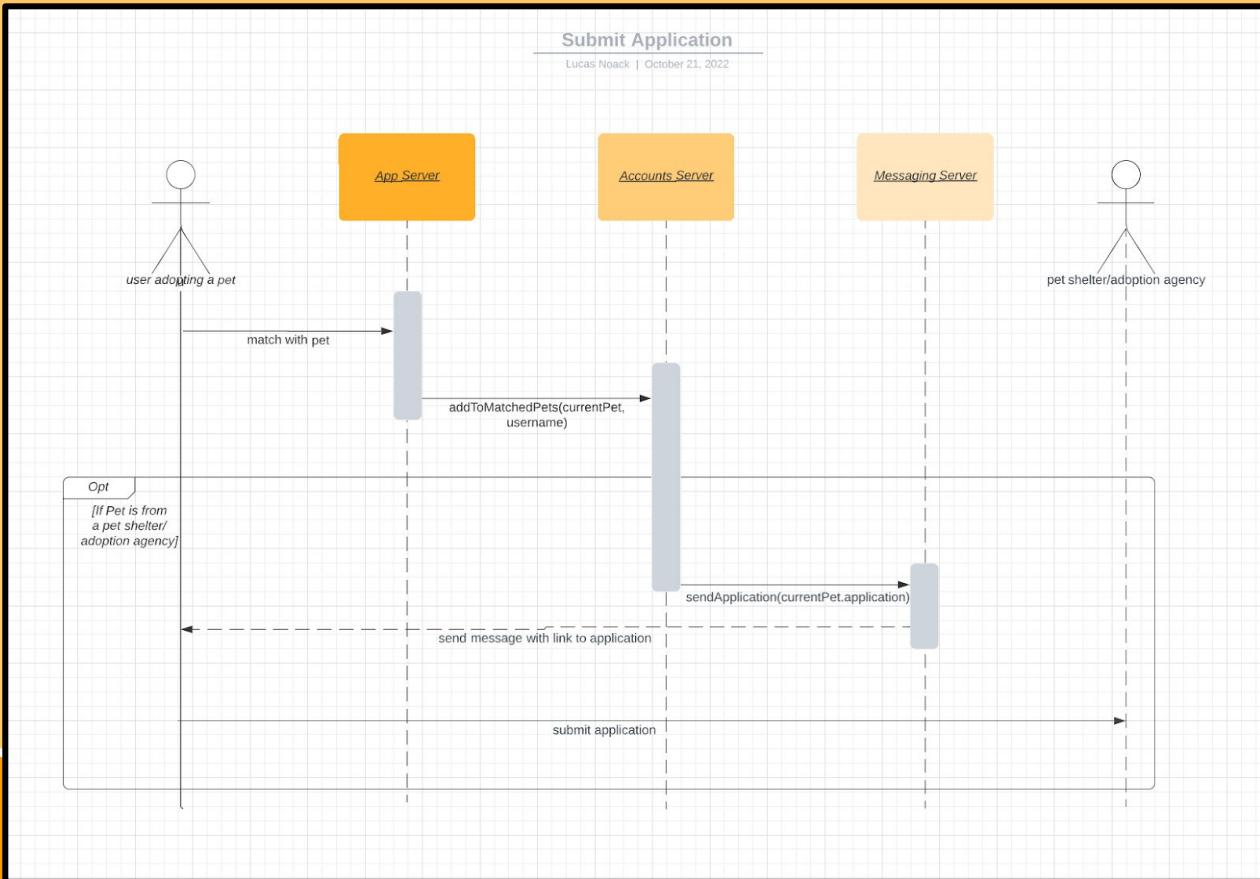
Sequence Diagram 4



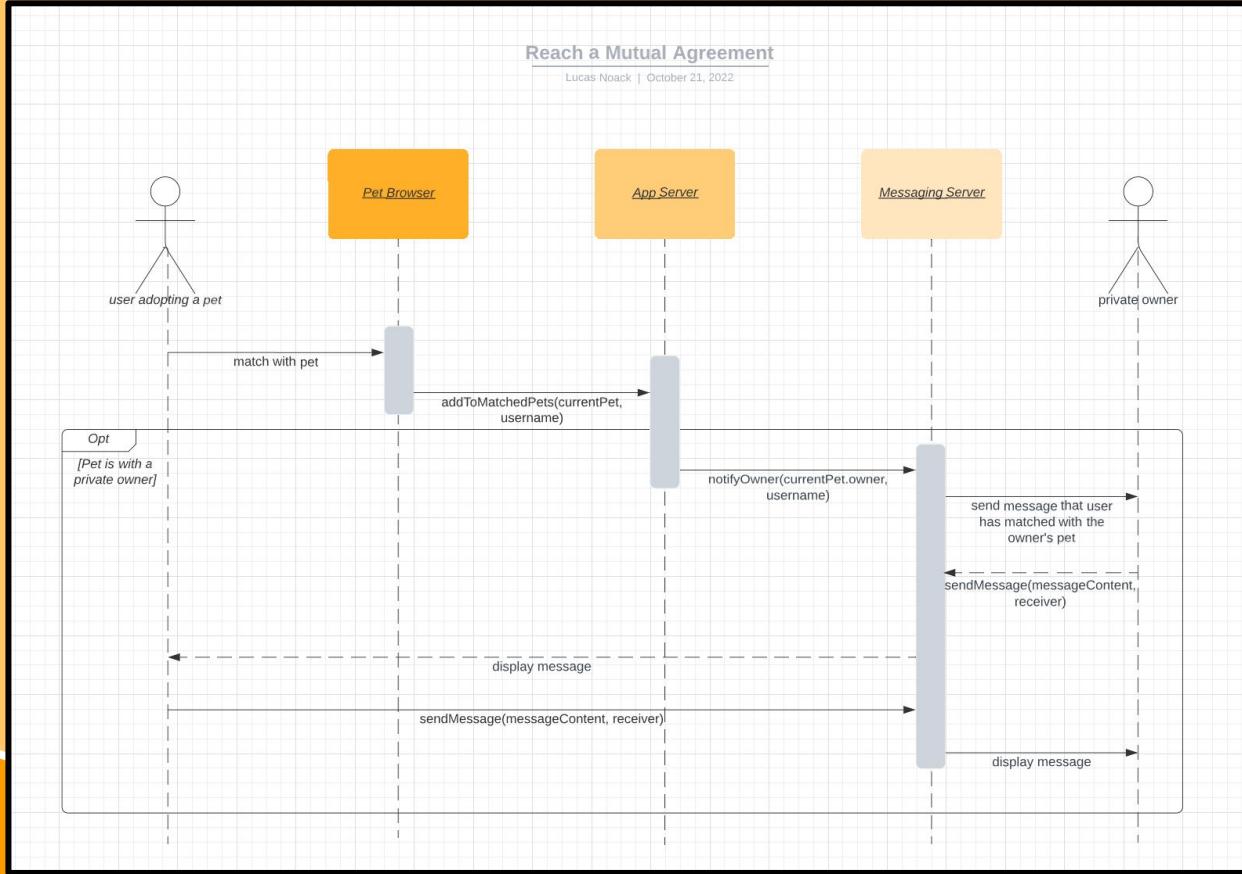
Sequence Diagram 5



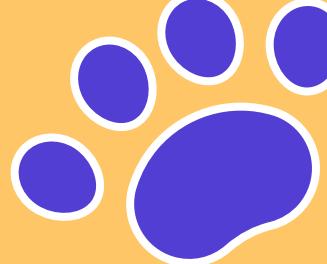
Sequence Diagram 6



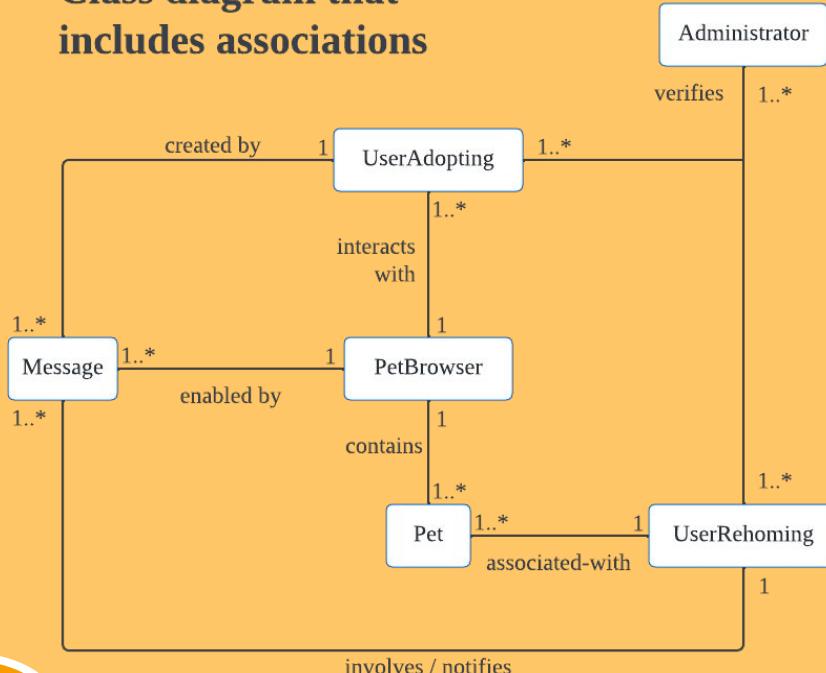
Sequence Diagram 7



Class Diagram 1

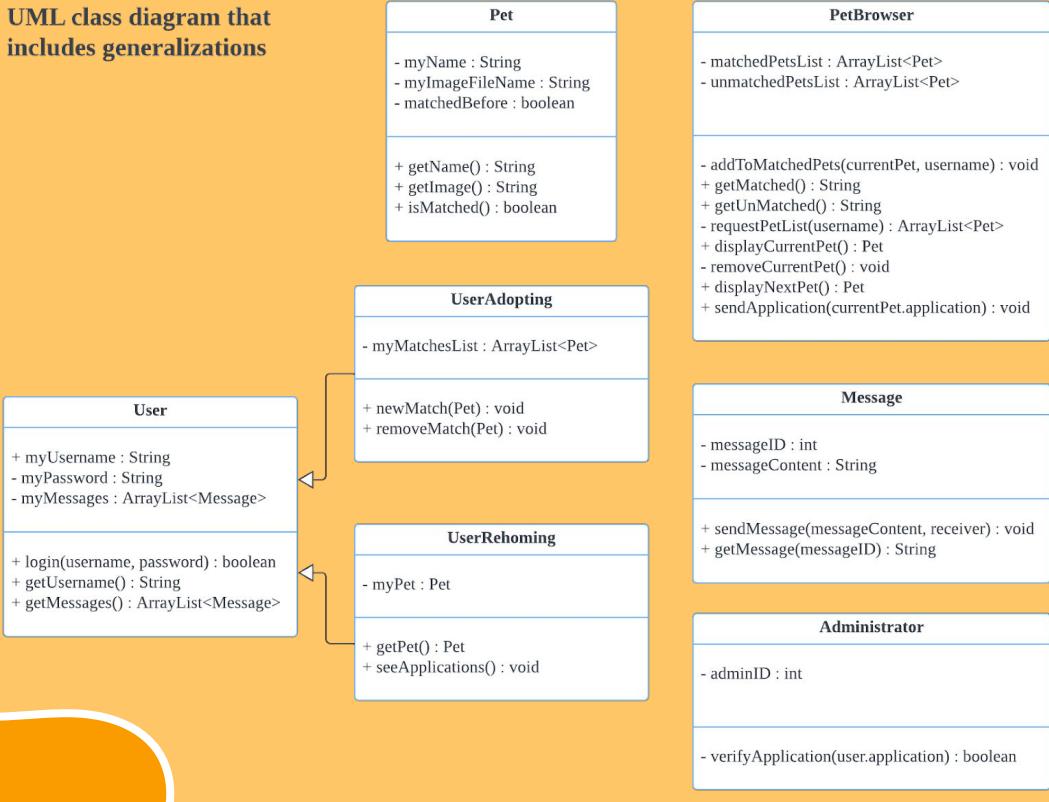


Class diagram that includes associations

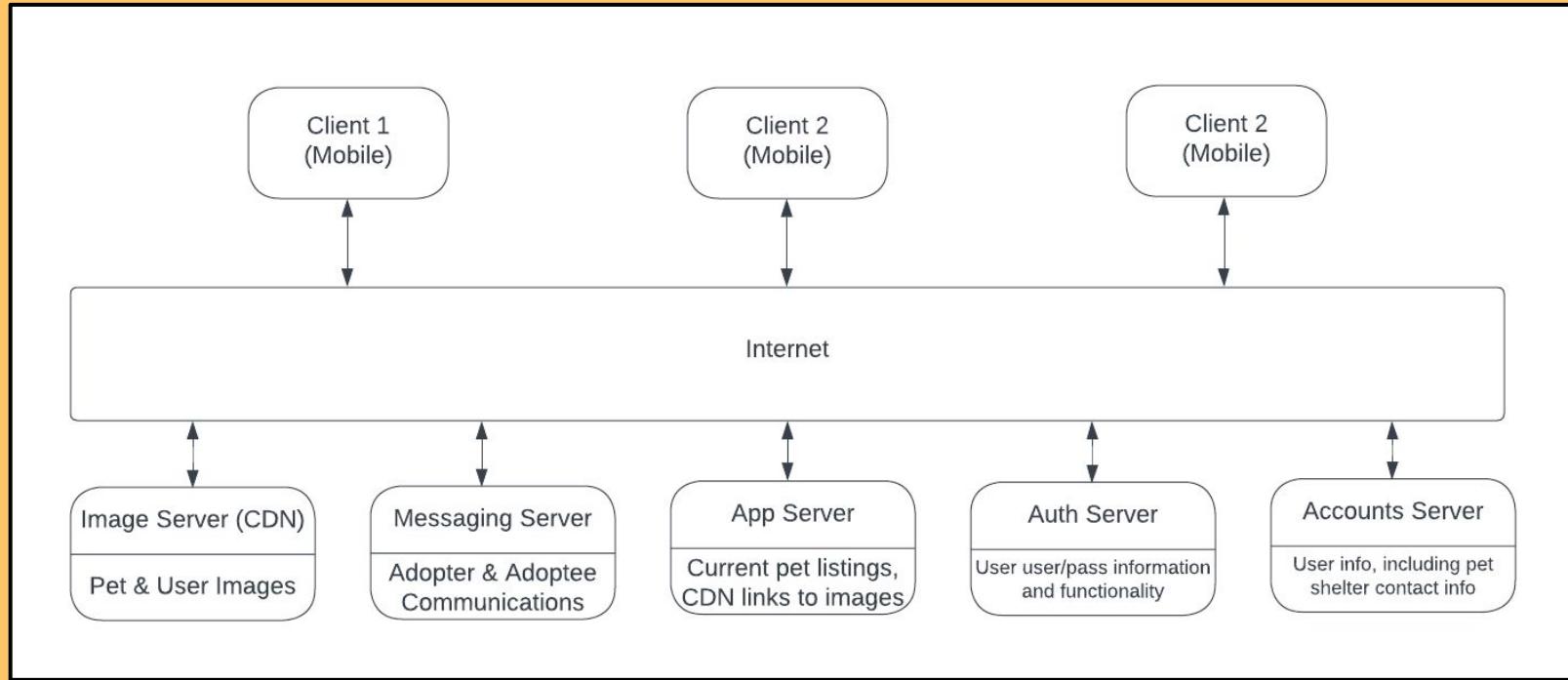


Class Diagram 2

UML class diagram that includes generalizations



Client-server Architecture



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 serve 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



Start

Setting a hypothetical start date of December 1st, 2022

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>. [Accessed: 18-Nov-2022].

Thanks!

Questions?



CREDITS: This presentation template was created by [Slidesgo](#), and includes icons by [Flaticon](#) and infographics & images by [Freepik](#)



9. [5 POINTS] GitHub requirement: Make sure at least one member of your group commits everything for project deliverable 2 to your GitHub repository,

- i.e. - Your final project deliverable2 report
- Unit test code for a sample unit of your project
- Presentation slides

The screenshot shows a GitHub repository page for 'PetPal-TP / 3354-PetPal'. The repository is public. The 'Code' tab is selected. The main repository area shows a list of files and their commit history:

File	Commit Message	Author	Time Ago
PetPal-NA Add files via upload	Add files via upload	add461b	1 minute ago
addToMatchedPets	Add files via upload		8 hours ago
CS3354_PetPal_PresentationSlide...	Add files via upload		1 minute ago
CS3354_PetPal_Slides.pdf	Add files via upload		3 minutes ago
README.md	Update README.md		7 hours ago
project_scope.pdf	Add files via upload		29 days ago

On the right side, there are sections for 'About', 'Releases', 'Packages', and 'Contributors'. The 'About' section notes 'No description, website, or topics provided.' The 'Releases' section says 'No releases published' and 'Create a new release'. The 'Packages' section says 'No packages published' and 'Publish your first package'. The 'Contributors' section lists 'PetPal-NA Noah Acosta'.

Please note: Deliverable 2 report is now added, screen shot was provided to prove everything else was committed as well.