swipet

by

Max Graman, Kyle Marler, Paramjyot Sandhu, Torcato Vaz

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology/Cybersecurity

Tvaz	04/24/22
Torcato Vaz	Date
MGraman	04/24/22
Max Graman	Date
PSandhu	04/24/22
Paramjyot Sandhu	Date
Kmarler	04/24/22
Kyle Marler	Date
yahya M Gilany	04/24/22
Yahya Gilany, Faculty Advisor	Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services

April 2022

Table of Contents

Index of Figures

Index of Tables

Abstract

Significant barriers exist when trying to adopt a pet from a shelter. Adoption agencies lack the resources and tools to share crucial details and important demographics about their animals online. This lack of access and availability inhibits potential adopters from finding and adopting the right pet from an animal rescue shelter in their area. The obstacles adopters and agencies face surrounding digital communication, the growing number of animals in shelters, and adoption requirements emphasize a clear and current need for a solution. Swipet aims to streamline the adoption process by creating a web application that engages users to quickly find the right pet and providing shelters a digital tool to post their pets online. Swipet is a two-sided web application that caters to adopters and shelter organizations. On the adopter-facing portion of the app, Swipet will display animal images, bios, medical history, and other details about the pet to potential adopters in a design that encourages pet adoption. On the shelter/organization side of the app, Swipet will display users with a dashboard interface that handles adoption requests, activity monitoring, and supports animal uploads, updates, or deletions. With this approach, Swipet will eliminate many of the existing barriers between adopters, animals, and shelters. Utilizing emergent front-end frameworks and a reliable backend stack, the team at Swipet has developed a clean and engaging solution.

Introduction

Project Summary:

Swipet is a web-based platform that caters to people looking to adopt a pet, *adopters,* and organizations, or *shelters,* who are dedicated to the well-being of homeless animals. The goal is to eliminate barriers in the pet adoption process and get sheltered animals into the homes of adopters. Upon signing up as an adopter, Swipet will present users with a catalog of animals from which they can browse, filter, and start the adoption process. While *shelters* will post, monitor activity, and review adoption applications for any pet in their shelter. Swipet will connect passionate adopters with homeless animals stuck in shelters.

Problem Statement:

People struggle to find the pet they want and love from animal rescue shelters. Animals are suffering, and the quality of life goes down the longer these animals remain in shelters[1]. Animal shelters barely have the time to give their animals the care and attention they need. Reports estimate that shelters only allot on average 15 minutes a day toward each animal. Out of that, 6 minutes are for feeding, and only 9 minutes are for cleaning and care. The care animals receive in shelters is the bare minimum requirement for maintaining a healthy animal[1]. More animals are finding their way into shelters, and more people than ever are looking to adopt[2].

With national pet numbers seeing all-time highs and demands to adopt these pets growing at a steady rate[2], there is a clear need in the industry for an intuitive and friendly application connecting shelters with people looking to adopt a pet. According to Madigan[2] , "Demand for industry services may grow over the next five years to 2025 due to increasing consumer preference to adopt pets rather than purchase them from a breeder or mill, a trend that has been accelerated by stay-at-home orders." It is apparent that we need to connect adopters with animal shelters in their local communities. The overall pet adoption rate is projected to increase over the next few years[2]. The US Business Environment Profiles[3] reports that, "The number of pets has increased over the past five years, at an annualized rate of 1.3% to 185.6 million in 2021," and the "forecast value for pets in 2026 is expected to be 194.3 million" [3]. How can the industry keep up with this evident surge in individuals looking to adopt their pets from shelters?

Furthermore, a qualitative study conducted by Best Friends Animal Society[3] indicates that there are definitive barriers for potential adopters. First, during the initial search, adopters

---

[1] Turner, Patricia, Jim Berry, and Shelagh Macdonald. "Animal Shelters and Animal Welfare: Raising the Bar." *The Canadian Veterinary Journal* 53, no. 8 (2012): 893–96. Accessed September 20, 2021. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3398531/

[2] Madigan, J. "Animal Rescue Shelters." Oclc.org. 2020. Accessed 2021. https://my-ibisworld-com.uc.idm.oclc.org/us/en/industry-specialized/od5794/major-companies

[3] "Number of Pets (Cats and Dogs)." oclc.org, 2018. Accessed September 19, 2021. https://my-ibisworldcom.uc.idm.oclc.org/us/en/business-environment-profiles/f306/business-environmentprofile

cannot find "the kind of pet or breed that they want, pets located near them, and detailed information about the pets[3]." Secondly, after initial inquiries about an individual pet, many adopters run into shelters or rescues that "do not respond to their questions, refuse to provide details about a pet until an adoption application is submitted and approved, or refuse to place a pet outside of their geographic location[3]." The study also mentions that barriers exist when applying for adoption. Many find that the applications "have extensive and strict requirements, are very lengthy, are rejected based on criteria, or require additional steps such as background checks or home checks before being approved or denied[3]."

Solution:

Swipet will solve the problem by guiding users through an intuitive interface matching sheltered animals with potential adopters. The application will present users with a minimal app design that encourages pet matching and adoption. Swipet will match adopters with *sheltered* animals based on interest, location, and preference selections. After using the web app, Swipet will eliminate many of the barriers potential adopters face when searching for shelter animals and communicating with agencies online. Swipet reduces the time and energy needed to adopt a pet and gears users to find animals from rescue shelters in their community. This key usability factor sets Swipet apart from other solutions out there. Ideally, Swipet will include various features enabling shelter accounts to verify, process, and accept potential adopter applications. Secondary features may include payment processing, background checks for adopters, and document creation/signatures to handle the tedious administrative work. Swipet team members will work internally to preserve the confidentiality and privacy of data.

There will be two main profiles:

1. Adopters (general user)
2. Shelter or Organization account (animal rescue shelters/pet adoption agencies)

Upon logging in, Swipet will present adopters with a pet browsing screen filled with positive images of various animals. After finding a pet they like, they will click on a pet card and view specific details relating to the animal. These details will include information about the animals' personality, behavior history, health history, and stats to ensure the adopter knows precisely the kind of animal they are looking at. If they like what they see, they can “like” or “save” this animal and opt to begin the adoption process. After the adopter has completed the application and adoption process, the shelter/organization account the pet is registered with must approve their request. Once approved, the adopter will be able to go into the shelter and pick up their new pet.

Shelter or organization users will be presented with a dashboard screen to manage all their posted animals. From here, they can add new animals, update existing information, and monitor activity such as views and likes. Organization or shelter accounts will have the ability to view and accept adoption applications submitted by the adopters. To combat user bias, Swipet may include features to promote animals who receive little or no activity from users.

Project Source:

The project team formed more than a semester before the start of senior design. All our members were originally in a group together for the enterprise web application class. After finishing the course and project successfully, we discussed teaming up for next year's senior design project. We created a team chat and kept in touch with each other. When the time came, we decided to move forward and complete senior design as a group.

Initially, the team had several ideas for the final project. Approaching the last couple weeks of class, we had to filter and move forward with a single project idea, something we all shared a passion about. That's when Max brought up his struggle with pet adoption. He said the following: "My girlfriend and I recently went through this process; I thought it would be super simple with all the hype around adopting pets. Unfortunately, it was not. Lots of tiresome paperwork and interviews; the most challenging part was finding the right pet. I talked to several of my friends who have shared similar experiences, which confirmed this was a current problem that needs to be solved. Further bolstering this point, a news story broke in mid-August[4] that I happened to turn on my TV too randomly." The rest of the team empathized with the frustrations in the adoption process and knew that there was potential for a digital solution that could help both adopters (like Max) and shelter organizations. The project foundation was born here and Swipet began to take shape.

[4] Nuss, Allaire. "Animal Shelter in Germany Uses Tinder to Help Pets Find Their 'Purrfect Match.'" Daily Paws, 2021. Accessed 2021. https://www.dailypaws.com/pet-news-entertainment/feel-good-stories/munich-animal-shelter-tinder-pet-adoption

Discussion

Project Objectives/Goals:

This section highlights some of the major objectives and goals Swipet hopes to accomplish. Upon completion of each goal, Swipet will solve different elements of the problem at hand. These goals will eliminate many of the barriers and issues discussed in our *problem* section and will serve as a guideline during the development of the solution.

- Profiles for users looking to adopt
    - Adopters can sign up
    - Adopters will be able to browse pets near them
    - The ability of users to set their preferences
    - Swiping functionality
    - Adopters can easily find pet information and related details
- Profile for Organization/Shelter accounts
    - Views associated with this account type
    - “Profiles” and set preferences for animals
    - Dashboard to monitor activity
    - Functionality to approve adoption requests
    - Functionality to upload, edit, and remove animals to/from Swipet
- Intuitive and engaging UI/UX
    - Provide our users with an interface that is easy to use and encourages pet adoption
    - On the shelter side, we hope to provide a medium that shelters can use to post their pets and communicate with adopters
- Account authentication
    - Verify that users are real using google captcha
- Matching preferences
    - Pet and shelter matching based on user preferences and location
    - Ability to browse a catalog of rescue animals
- Develop with confidentiality and privacy of data in mind
- Develop with ethical considerations in mind

Project Scope:

Our team will develop a functional application that lets users quickly find preferred and adoptable pets in their area by utilizing the following features and functionality.

1. Adopter profile to display themselves to shelters.
    - 1.1. Personal account creation
    - 1.2. Account authentication
    - 1.3. Views associated with this account type
        - 1.3.1. Home page (adopter)
        - 1.3.2. Pet detail page
2. Shelter/Organization profile to manage animal listing and activity
    - 2.1. Account authentication
    - 2.2. Views associated with Shelter account:
        - 2.2.1. Shelter/Organization dashboard
3. An intuitive interface designed to maximize turnover for pets.
    - 3.1. Adopter Home screen designed to match the user with a pet
        - 3.1.1. Ui displaying a catalog of posted pets
    - 3.2. Pet/animal detail page
        - 3.2.1. Display animal info, i.e., type, bio, vaccine info, etc.
    - 3.3. Shelter/Organization dashboard
        - 3.3.1. Give shelters the ability to create “profiles” for each animal they wish to list on Swipet.
        - 3.3.2. UI dashboard for shelters to view activity on posted animals
            - 3.3.2.1. Likes, views
4. Pet and shelter matching based on location and preferences.
    - 4.1. Location tags
    - 4.2. Animal type tags
5. Provide a platform for shelters and adoptees to communicate.
6. Shelter pilot program

Potential features

- 6.1. Pre-screening individuals who wish to adopt pets for profile verification.
- 6.2. Implement external document creation and signature functionality.
- 6.3. Implement external payment processing.

Throughout project planning and project management, the team concluded that the features should be built on top of the main objective of Swipet and align with the project vision and goals. The vision for the project assumed the team would acquire core application architecture, user interface, and user workflows. These workflows would enable users to pursue the objective and goals outlined above by the end of first semester. The team assumption at this point was to have version 1 major release, which would be more than just a minimum viable product. We also envisioned that business and functional logic, additional features such as a communication platform, along with other feasibility features would be added in the second

semester as the team agreed on the importance of each. The team hoped to have completed our version 2 major release in time for the sprint ending before the IT Expo with the assumption that these features, functional tests, and bugs fixes were to be completed. Although the assumptions for the first semester came true and goals were achieved, the team faced some constraints in the second semester. Time was the biggest constraint, some of the features like applications and matches tracking required re-designing our workflows and database schemas. This took additional time out of the iterations we had planned to create a communication platform. Thus, we had to limit the way of post application approval communication, between our user personas to deliver our application in time.

Quick Project Timeline:

The project timeline breaks down the scope of our project into 16 measurable tasks and sprints. Swipet will use the table below to ensure deadlines and objectives are met in a timely manner.

Table 1 Project Timeline Table

Task #	Task Name	Duration	Start Date	End Date
#1	Team Contract	1 month	Aug 23	Sept 20
#2	UI Mockups	1 month	Sept 20	Oct 10
#3	Branding	1 week	Sept 20	Sept 28
#4	UML Diagrams	1 week	Sep 28	Oct 10
#6	Use Case Diagrams	1 week	Oct 11	Oct 18
	Development			
#7	Sprint #1: Adopter Profile and Org/Shelter Profile – UI. Workflows and routing	2 weeks	Oct 18	Nov 1
#8	Sprint #2: Org/Shelter Profile Functionality	2 weeks	Nov 1	Nov 15
#9	First Semester Demo		Nov 15	Nov 29
#10	Sprint #3: Adopter Profile Functionality	2 weeks	Dec 1	Dec 15
#11	Sprint #4: Matching and Preference Build	2 weeks	Dec 1	Dec 15
#12	Sprint #6: Clean-up	Break	Dec 15	Jan 14
	Major Release v1.0.0			
#13	Sprint #1: Shelter Org Pilot	2 weeks	Jan 18	Feb 1
#14	Sprint #2: Integrations	2 weeks	Feb 1	Feb 15
#15	Sprint #3: Clean Up	2 weeks	Feb 15	Feb 29
	Final Report			
#16	Presentation/IT Expo		Mar 21	Apr 4
	Major Release: v2.0.0			

Technologies Used:

This stack uses some of the newest frameworks and technologies around. When considering what technology to use it is important to consider what you already know versus what you want to learn. In our case, these are the technologies we are familiar with. This stack is proven, quick, and reliable. For more details, please see the GitHub repository[5].

Several different tech stacks exist for web development. Deciding what will work best depends largely on project specifications and team skills. For the front-end portion of our app, the main technologies used include React, TypeScript, and a compatible styling library called Chakra UI. Not only have members of Swipet had positive previous experience with the technologies, but we believed this would generate the performance and flexibility in delivering an interactive UI; a fundamental design goal for Swipet. Other popular stacks are LAMP (Linux, Apche, MySQL, PHP), ASP.NET (Microsoft's framework), MEAN (MongoDB, Express, Angular, and Node), and MERN (same as MEAN but uses React instead of Angular). The team decided to go with a stack that utilized React as it is known for its ability to use codes on browsers and servers simultaneously. This would help with front and backend development and was a preferred technology within the team. From there, Swipet created a stack similar to the popular MERN stack with adjustments to the backend for compatibility and ease-of-use with self-hosting. The below section outlines all the technologies used in the project.

Main development stack:

- Redis - for fast, in memory server caching of authentication and cookies.
- PostgreSQL - Database for storing user information. Users are individuals searching for pets and organizations.
    - Managed by PGAdmin, a graphical front end for PostgreSQL.
- Node.js - JavaScript server back-end.
    - Using Apollo, Express.js, and GraphQL.
- React - Front-end JavaScript framework.
- Chakra-UI - Front-end UI framework.
- Typescript - JavaScript with syntax for types.
- GraphQL - Uniform communication between front-end and back-end regardless of platform.

Hosting stack:

- Proxmox - platform running virtual machines for our environment.
- Ubuntu Linux - OS running our Rancher instance and GitHub workflow runners.
- Rancher – enterprise Kubernetes management.
- Docker – containerizing everything so individual virtual machines don't overuse server resources.
- GitHub – CI/CD pipelines.

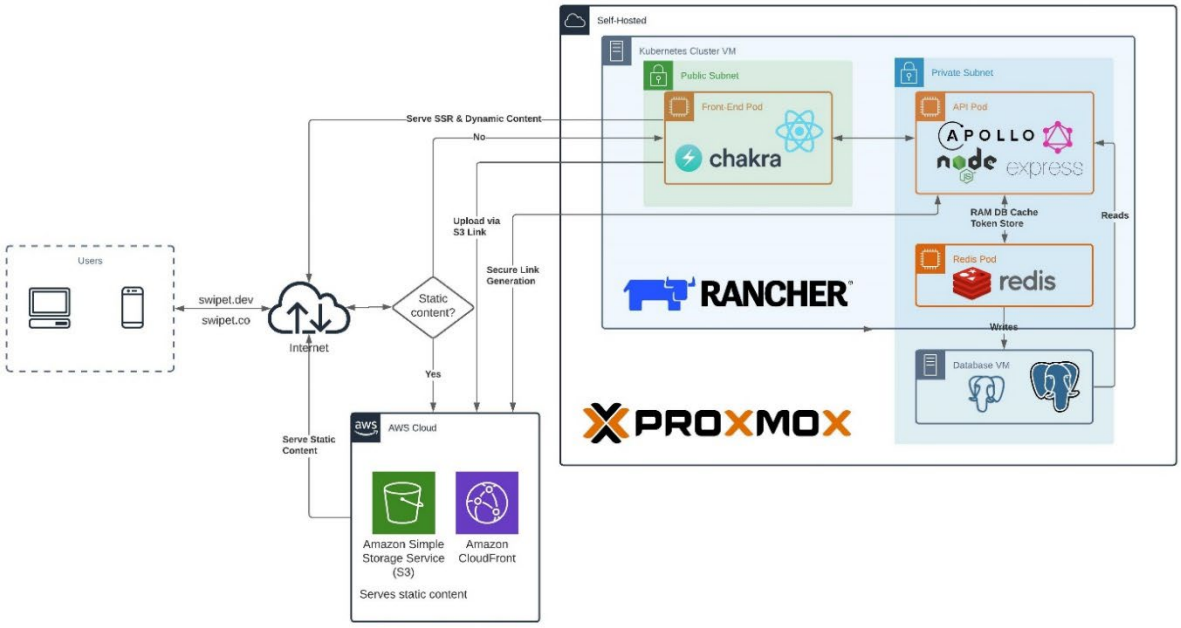
[5] Github Repository URL: https://github.com/palepinkdot/senior-design

- AWS
    - S3 - for serving static content.
    - CloudFront – CDN.
    - Lamda – uploading images to S3 bucket for animals.
- Hover – DNS and domain name provider.

Technical Architecture Diagram:

The diagram below illustrates the inner machinations of Swipet. Starting on the left, a user (either on mobile or desktop) navigates to the swipet website and is served static content from Amazon Web Services S3 buckets which are cached on CloudFront. Dynamic content is retrieved via a backend using Node.js, express, Apollo, GraphQL, Redis, and PostgreSQL. All these backend services are hosted on a self-hosted server stack inside of docker containers, managed by Kubernetes via Rancher. The bare metal hypervisor in use is Proxmox. Users can upload content such as images through the front end of the application to S3 buckets via securely generated links, then they are served as static content.

Figure 1 Technical Architecture Diagram



User Personas:

The tables below illustrate the several different user personas for our project. When planning, designing, and building out our user personas for Swipet, the following characteristics were front and center: who might use our application, what their demographics might be, the behavior we may expect from them, what pains they may encounter, and what needs & goals within our project we may need to satisfy. There are two main categories for our users: **adopters** and **organization/shelters**. Within each of these categories, we thought several key personas were most important to highlight. The tables below detail our two user personas and the respective users within them.

Table 2 User Persona Table

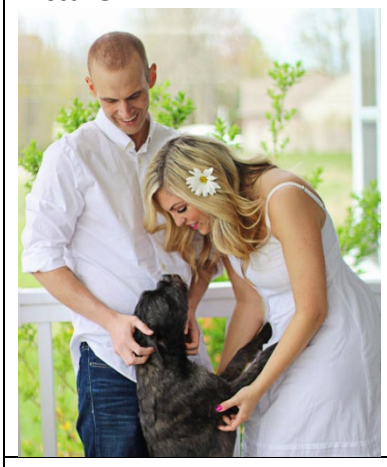
<table>
<tr><td colspan="2">User Persona: New Couple (adopter)</td></tr>
<tr><td rowspan="4">Picture
  </td><td>Title: Ms.</td></tr>
<tr><td>Name: Julia Adams</td></tr>
<tr><td>Age: 28</td></tr>
<tr><td>Gender: Female</td></tr>
<tr><td>Behavior</td><td>• Newly engaged couple
• Buys a new house and just moved in
• Looking for a new pet for the home</td></tr>
<tr><td>Pain</td><td>• They want to adopt a new dog but does not know where to start
• Is conscious about breeding mills and conflicts associated with mills</td></tr>
<tr><td>Needs & Goals</td><td>• Looking to adopt a pet
• They want to go through the shelter route
• Build a family
• Cant' find animals in local rescue shelters</td></tr>
</table>

Table 3 Single Parent Persona

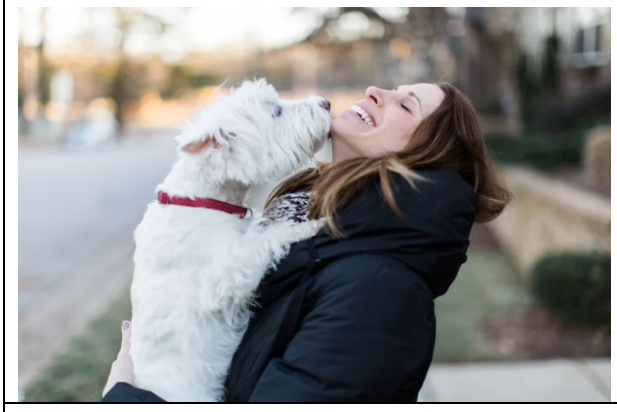
<table>
<tr><td colspan="2">User Persona: Single Parent (adopter)</td></tr>
<tr><td rowspan="4">Picture
  </td><td>Title: Ms.</td></tr>
<tr><td>Name: Elizabeth Jacobs</td></tr>
<tr><td>Age: 24</td></tr>
<tr><td>Gender: Female</td></tr>
<tr><td>Behavior</td><td>• A single parent with a three-year-old child
• Works at a local restaurant as a bartender
• Living in an apartment
• Their father stops by to take care of the child when they are at work</td></tr>
<tr><td>Pain</td><td>• Does not have the time to visit many different shelters to find a cat that meets their needs.
• The child requires much socialization</td></tr>
<tr><td>Needs & Goals</td><td>• They want to adopt an older cat from a rescue
• Desires cat to have certain personality traits such as being playful and well-adjusted to humans
• Needs a cat to be good with children</td></tr>
</table>

Table 4 Emotional Support Animal User Persona

<table>
<tr><td colspan="2">User Persona: Emotional support needs Individual (adopter)</td></tr>
<tr><td rowspan="4">Picture
  </td><td>Title: Mrs.</td></tr>
<tr><td>Name: Jane Edward</td></tr>
<tr><td>Age: 35</td></tr>
<tr><td>Gender: Female</td></tr>
<tr><td>Behavior</td><td>• Experienced trauma recently
• Is less likely to get support through friends</td></tr>
<tr><td>Pain</td><td>• Spending most time alone
• Less likely to socialize</td></tr>
<tr><td>Needs & Goals</td><td>• Needs activity for better physical health
• Needs activity for better mental health
• Feel loved, supported, not alone</td></tr>
</table>

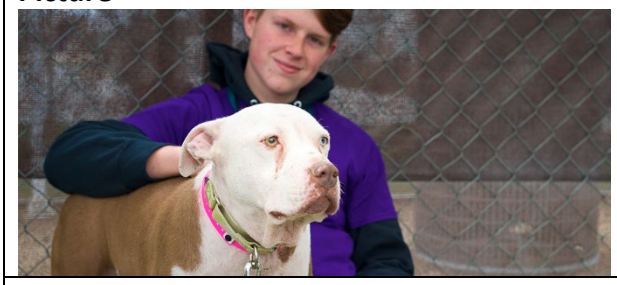
Table 5 Accessibility Needs Persona

<table>
<tr><td colspan="2">User Persona: Individuals with Accessibility needs (adopter)</td></tr>
<tr><td rowspan="4">Picture
  </td><td>Title: Mr.</td></tr>
<tr><td>Name: Cory Zed</td></tr>
<tr><td>Age: 65</td></tr>
<tr><td>Gender: Male</td></tr>
<tr><td>Behavior</td><td>• Recently retired
• Living alone</td></tr>
<tr><td>Pain</td><td>• Have a tough time navigating path due to lowered vision
• Have a tough time remembering daily tasks, to-do items</td></tr>
<tr><td>Needs & Goals</td><td>• Need someone to help them navigate the everyday path
• Need someone to call for help in case of emergency
• Need someone to remind them of specific tasks such as walking, medicines, etc.</td></tr>
</table>

Table 6 Shelter/Owner Admin Persona

User Persona: Shelter Owner/Admin (Shelter)	
Picture 	**Title**: Ms. **Name**: Rose Potter **Age**: 28 **Gender**: Female
Behavior	• Is the current owner of a family-owned shelter • Overlooks adoption process of 100s of animals
Pain	• Has a tough time finding suitable adopters • Has tough time keeping contact with potential adopters • It is harder for them to be able to manage time to promote pets to adopters
Needs & Goals	• Successfully find a home for each pet • Successfully screen adopter applicants to find the right owner for each pet

Table 7 Shelter Volunteer Persona

<table>
<tr><td colspan="2">User Persona: Shelter Volunteer (Shelter)</td></tr>
<tr><td rowspan="4">Picture
  </td><td>Title: Mr.</td></tr>
<tr><td>Name: John Goodwin</td></tr>
<tr><td>Age: 18</td></tr>
<tr><td>Gender: Male</td></tr>
<tr><td>Behavior</td><td>• Full-time Biology student at local university
• They live in a dormitory
• Volunteers at shelter part-time
• They have volunteered at shelters since he was 13</td></tr>
<tr><td>Pain</td><td>• Animal exposure minimal
• Old website technology is not intuitive to manage</td></tr>
<tr><td>Needs & Goals</td><td>• Find a way to get their animals more exposure to potential adopters
• Help potential adopters get more information about animals available for adoption</td></tr>
</table>

Use Cases:

When planning, designing, and building out the use cases for Swipet, the following characteristics were kept in mind: actors involved, triggers, frequency, preconditions, happy flows, alternate flows, and finally, any postconditions that may be involved. The use cases were created by combining Swipet's key objectives/goals and the perspectives of the previously identified user personas. The tables below illustrate the different use cases for our project.

Table 8 Adopt A Pet Use Case

Use Case ID	UC: #1
Use Case Name	Adopt a pet
End Objective	User successfully adopts a pet
User/Actor	Adopter - any of the adopter personas
Trigger	The user wants to find a pet to adopt from a shelter
Frequency of Use	Completes this process once for every animal adopted
Preconditions	User has been registered, verified, gone through the adoption application process, and finally accepted by the org/shelter
Basic Flow	A 1. Signs up as a general user (adopter) 2. Browses for animals 3. Finds animals they like 4. Initiates adoption process 5. Accepted by shelter 6. Goes through the application process 7. Picks up the animal from a shelter
Alternate Flow	B 1. Signs up as a general user (adopter) 2. Find the animal from saved list 3. Initiates adoption process 4. Accepted by shelter 5. Goes through the application process 6. Picks up the animal from a shelter
Postconditions	Adopter takes the pet home; pet and owner are compatible.

Table 9 Browse Catalog of Pets Use Case

Use Case ID	UC: #2
Use Case Name	Browse the catalog of pets
End Objective	Users can successfully browse, get more information, like/save animals, and find a pet available for adoption that matches their preferences.
User/Actor	Adopter - Any of the Adopter personas
Trigger	Users want to browse the saved animals or the updated availability of all animals in shelters around them
Frequency of Use	High frequency of use. Users repeat this process every time they are interested in adopting an animal and continue to find the preferred animal.
Preconditions	The user is a potential adopter, ready to move forward and through the complete adoption process.
Basic Flow	A 1. Signs up as a general user (adopter) 2. Lands on the Application homepage, to browse animals 3. Accesses quick look of currently displayed animal 4. Decides to proceed (swipes right to save) 5. The user repeats the process until they have narrowed their selection to only the animals they wish to adopt
Alternate Flow	B 1. User Logs In 2. Lands on the Application homepage 3. Navigates to saved animals list 4. User browses through the list of saved animals
Postconditions	The user's decision (Swipe left/right) is acted upon accordingly. Users can browse again if they want to.

Table 10 Create Adopter Account Use Case

Use Case ID	UC: #3
Use Case Name	Create adopter account
End Objective	Users create an account for themselves to access features of the app
User/Actor	Adopters –Any of the personas
Trigger	The user wants to access features of the application
Frequency of Use	Once per User
Preconditions	The user is not a currently active user of the application already
Basic Flow	1. User lands on the application landing page 2. User clicks 'Get Started' 3. User is brought to the page asking them to confirm if they are an adopter of the organization 4. User fills the prompted registration form 5. User clicks the 'Sign Up' button 6. The user is registered. 7. User is sent back to the Adopter homepage
Alternate Flow	Not Applicable
Postconditions	Users can log in to their account and use the 'Adopter' features of the application

Table 11 Create Shelter/Org Account Use Case

Use Case ID	UC: #4
Use Case Name	Create shelter/org account
End Objective	A user creates an account for their shelter or organization
User/Actor	shelter
Trigger	The user wants to create an account for their shelter
Frequency of Use	Once per user
Preconditions	The user is an existing shelter or organization that wants to create an account and proves that their shelter or organization exists.
Basic Flow	1. User lands on the application landing page

	2. User clicks ‘Get Started’ 3. User is brought to the page asking them to confirm if they are an adopter of the organization 4. User fills the prompted registration form 5. User clicks the 'Sign Up' button 6. User is registered. 7. User is brought back to the Shelter homepage
Alternate Flow	Not Applicable
Postconditions	Users can log in to their account and use the 'Shelter' features of the application

Table 12 Post New Animals Use Case

Use Case ID	UC: #5
Use Case Name	Post new animals in Swipet up for adoption
End Objective	A profile of an animal, ready for viewing by users.
User/Actor	Shelter/Org account
Trigger	A shelter wants to list a new pet for adoption.
Frequency of Use	As needed - many times per day to a few times a day depending on shelter size and adoption of the program.
Preconditions	The user must be logged in. The user must be an organization account.
Basic Flow	1. User navigates to the pet upload page through the side navigation bar 2. User correctly inputs information about the animal they are uploading. 3. User uploads an image of the animal. 4. User clicks the submit button for processing. 5. Animal is posted and can be seen by ‘Adopters’ while browsing
Alternate Flow	Not Applicable
Postconditions	Adopting users can view the new animal.

Table 13 Edit/Update/Delete Existing Animal Entries Use Case

Use Case ID	UC: #6
Use Case Name	Edit/Update/Delete existing animals for adoption
End Objective	Mange a shelter's listings
User/Actor	Shelter User
Trigger	The user needs to change the details of an uploaded pet.
Frequency of Use	As needed.
Preconditions	An animal must be uploaded and have an existing profile on Swipet
Basic Flow	1. User Logs In 2. User navigates to landing page 3. User selects 'Animals' tab on the page view 4. User finds animal on the list to edit 5. User clicks the edit/del button 6. User changes needed information/User selects delete animal.
Alternate Flow	Not Applicable
Postconditions	An edited animal with correct information available to users

Table 14 Application/Adoption Process Use Case

Use Case ID	UC: #7
Use Case Name	Verify adopter and step through application/adoption process.
End Objective	Guarantee that an adopter is a legitimate person and a suitable match for the animal.
User/Actor	shelter
Trigger	Shelter/Organization will request adopter verification when the adopter applies for one of their animals
Frequency of Use	As needed.
Preconditions	Adopter is interested in an animal available at the shelter and agrees to adhere to the process and guidelines of the shelter
Basic Flow	Pending Teacher Approval for this feature
Alternate Flow	Not Applicable

Postconditions	Adopter is verified and successfully adopts the animal

Table 15 Browse Submitted Adoption Applications

Use Case ID	UC: #8
Use Case Name	Browse submitted adoption applications
End Objective	Users should be able to browse through all applications submitted by adopters for animals in their shelter
User/Actor	shelter
Trigger	User wants to access applications for information, approvals and declining.
Frequency of Use	High frequency, each user can repeat the process as many times to access applications
Preconditions	User has a valid shelter account registered.
Basic Flow	1. User Logs In 2. User clicks on side-bar menu 3. User Navigates to 'Applications' Tab 4. User scrolls through list of applications 5. User clicks on any application to view details and perform any necessary actions on it.
Alternate Flow	1. User Logs In 2. User navigates to landing page 3. User makes sure 'Applications' tab is selected in quick overview 4. User browses applications with recent activity 5. User clicks on any application tile 6. The tile expands and brings user to application detail page
Postconditions	Shelter persona user was successfully able to access applications

Table 16 Check Status of Received Application

Use Case ID	UC: #9
Use Case Name	Check status of received application
End Objective	To check the progress status of any application received from adopters
User/Actor	shelter
Trigger	User wants to check application status

Frequency of Use	As needed. Few times for each active application
Preconditions	Users have a valid shelter account. The application is still active.
Basic Flow	1. User Logs In 2. User clicks on side-bar menu 3. User Navigates to 'Applications' Tab 4. User scrolls through list of applications 5. User selects the application for which user wants to check progress status 6. User is brought to application detail page 7. User views the activity progress status of the application
Alternate Flow	1. User Logs In 2. User navigates to landing page 3. User makes sure 'Applications' tab is selected in quick overview 4. User browses applications with recent activity 5. User clicks on any application tile 6. The tile expands and brings user to application detail page 7. User views the activity progress status of the application
Postconditions	Shelter persona user was able to check the status of the application

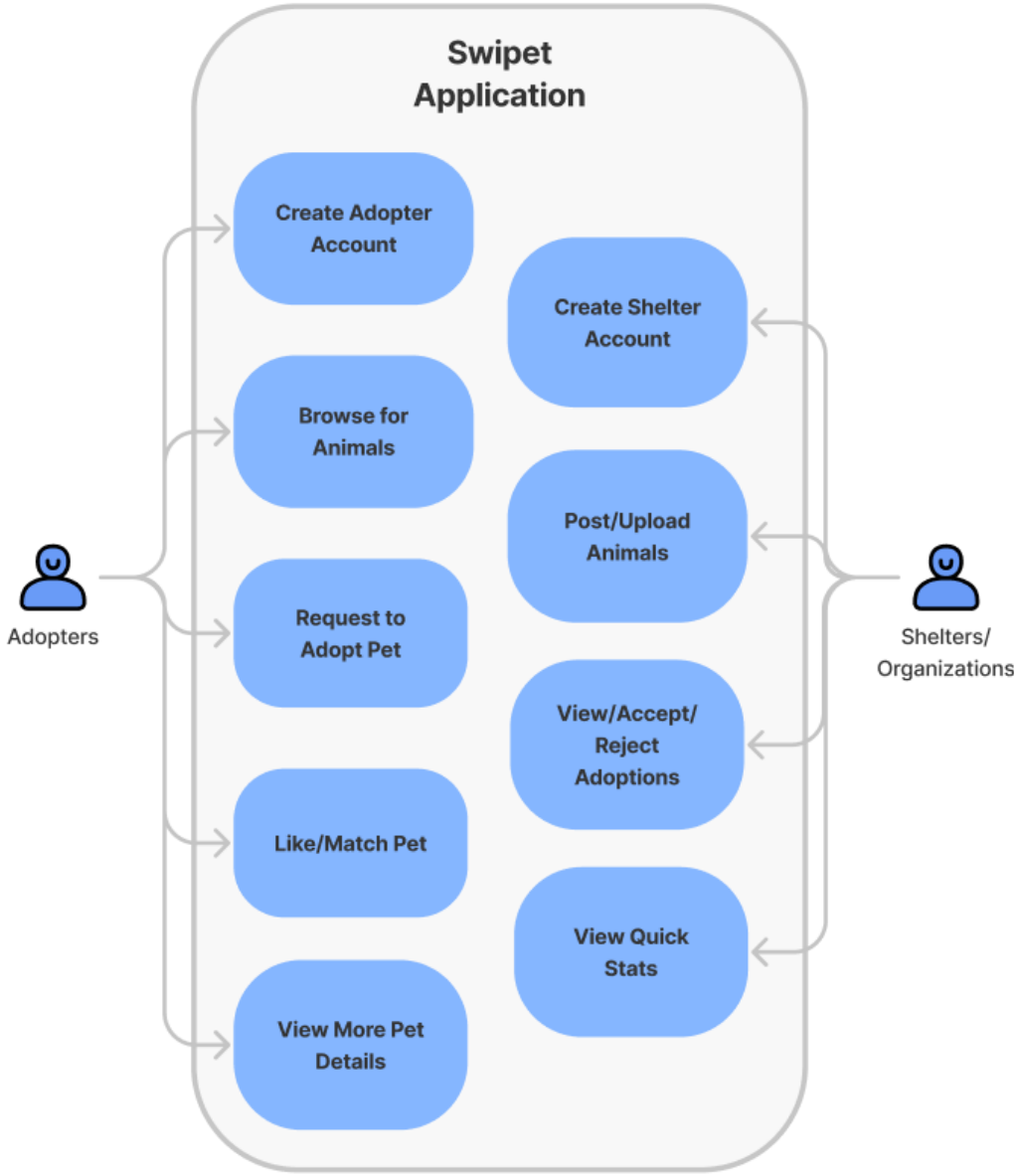
Table 17 Check Status of Submitted Application

Use Case ID	UC: #10
Use Case Name	Check status of submitted application
End Objective	To check the progress status of any application that a user submitted
User/Actor	Adopter
Trigger	User wants to check application status
Frequency of Use	As needed. Few times for each active application
Preconditions	Users have a valid Adopter account. The application is still active.
Basic Flow	1. User Logs In 2. User clicks on side-bar menu 3. User Navigates to 'My Applications' Tab 4. User scrolls through list of submitted applications 5. User selects the application for which user wants to check progress status 6. User is brought to application detail page 7. User views the activity progress status of the application
Alternate Flow	Not Applicable
Postconditions	Adopter persona user was able to check the status of the application

Use Case Diagram:

Below is the use case diagram. It visually explains all use cases identified in the section above and how they are related to each user profile. On the adopter side, users can create adopter accounts, browser for animals, and request to adopt a pet. On the shelter side, users can create a shelter account, post new animals, and edit/update/delete postings.

Figure 2 Use Case Diagram



Testing Plan:

Overview

This section includes an overview of the methodology, steps, and testing processes that Swipet overtook to ensure application security, workflows, seamless operations, and functionality of implemented features. Every measure taken is to ensure the application runs smoothly and as expected.

Methodology

For the testing methodology, there were a mix of several different approaches. Starting at the development level, the team implemented GraphQL verification on build. This verification ensures that any data requested from the API to the front-end is valid and will return in a predictable shape. Additionally, as this is considered critical infrastructure, it acts as a test to ensure there are no significant issues in the application and serves as the first barrier when testing.

Continuous Integration and Deployment are a crucial part of our development strategy. Using GitHub Actions the committed and pull-requested code is:

- Front-end Build Tested
- API endpoints and data generation Tested
- Front-end Units Tested
- API Unit tested
- Docker image build tested
- DockerHub image push tested
- Deployed to a test environment

By leveraging a reflective test environment, we can thoroughly test features as if they are hosted and deployed in the production environment since the production environment is a copied mirror of the development environment.

User Acceptance Testing (UAT) will be incorporated into the project development as well. With this approach, the team hopes to test the application from the perspective of our users. Swipet will set up a test environment that incorporates test data and guided use case scenarios for the test users to interact with the project. With this approach, the team will capture crucial feedback reflecting how the solution should operate, any bugs in the solution will be discovered, and plans to solve them will be made.

Functional testing is incorporated using Cypress end-to-end testing framework to ensure that the application's front-end is robust. We have added tests that help automate frontend testing at various levels, meaning, we have implemented tests that would walk through the application as if it were a black box, where steps are inspired from UAT. Additionally, we have added tests to

ensure that the main workflows and intended objectives are always working and bug free. It helps save time for developers by quickly walking through the application pages and acts as a last line of defense in case a feature affecting other app areas was implemented without core testing.

Test Logs and Procedures

Table 18 Test Logs and Procedures

Test Case ID	Test Obj. ID	Category	Condition	Expected Results	Actual Results	Requirement ID
001	01	UAT	Adopter Sign Up	User successfully registered themselves.	Expected results accomplished. User was then unable to sign in after creating account.	01
002	02	Man.	Registered account	Can swipe, select adopt, and view details of specific pets.	Noticed a couple inconsistencies when swiping. Mobile view needs a lot of work.	02
003	03	UAT	Org Account	Can register for an org. account	Expected results but missing sign in functionality.	03
004	04	Man.	Org User	Can upload pets to DB	Was unable to access Org side due to bug when testing, but this functionality is working.	04
005	05	Man.	Adopt a pet	Adopter can request to adopt an animal and the org should see be able to see this request.	Nothing happens after clicking the adopt now button. Not connected to applications or org side.	05

Testing Review

Through manual testing, done with the use of in-browser developer tools and on multiple devices, we have learned that our web application is not fully responsive to different screen sizes. We need to ensure our code covers responsiveness for all new elements added moving on.

Change Management Plan:

The following section illustrates how the Swipet team will manage and process changes. A change is an addition, modification, or removal of anything that could directly or indirectly affect the project. This section was developed using guidelines from in class lecture, Swipet team specifications, and notes from the editorial team at "Bit.AI Blog: Change Management Plan: What is it and How to Create it?"

The change management plan identifies the following:

1. **To identify WHO can make a change and in WHAT circumstance Swipet will use the following guidelines:**
    a. In the circumstance that a change needs to be made to the development of Swipet
        i. All Swipet internal team members can request changes to the project
        ii. Swipet advisors can request changes to the project
        iii. Senior Design professors can request changes to the project
    b. In the circumstance that a change needs to be made involving roles/responsibilities of team members
        i. All Swipet internal team members can request
    c. In the circumstance that a change needs to be made surrounding team contract and final report
        i. All Swipet internal team members can request changes to the project
        ii. Swipet advisors can request changes to the project
        iii. Senior Design professors can request changes to the project
    d. In the circumstances that a change needs to be made to the testing structure of the application
        i. All Swipet internal team members can request changes to the project
        ii. Swipet advisors can request changes to the project
        iii. Senior Design professors can request changes to the project
    e. In the circumstances that a change needs to be made in the workflow of the application UI/Service layer based on the input from piloting/user testing
        i. All Swipet internal team members can request changes to the project
        ii. Swipet advisors can request changes to the project
        iii. Senior Design professors can request changes to the project
        iv. Users of application Adopter/Shelters

2. **To PROCESS and MANAGE the change Swipet will use the following guidelines:**
    a. The CCB (Change Control Board) will **receive** *change requests*
        i. Change Control Board Includes:
            1. All internal team members of Swipet
    b. The CCB will **review** *change requests* according to the following:

REVIEW PROTOCOL:[6]

i. Define the change and align it to business objectives. Keeping in mind the following:
    1. The change requirements
    2. Change definition
    3. Scope of the change and impacts on other areas of application
    4. Comparison of proposed change and current solution
        a. How does it help, or how is it better than the current solution?

ii. Determine the impact of the change and whom it may affect. Considering the following:
    1. Whom will the change affect the most?
        a. Review the effect on each business unit and how it translates through the organizational structure to that person
    2. What are the impacts of the change?
        a. Use this data to generate the blueprint for where training and support are needed the most to mitigate the effect of the change
    3. How will the change be received?

c. The CCB will **communicate** the change using the following:

COMMUNICATION STRATEGY:[6]

i. Develop a communication strategy
    1. How will feedback be managed?
    2. How will the change be communicated?
        a. Develop a timeline for appropriate communication channels to use, key messages, and how the change will be communicated

ii. Upon review and approval of change request, communication to our team and advisor will take place
    1. The team will relay information gathered from *review protocol* to appropriate advisors, sponsors, professors, etc.
    2. Depending on severity and impact of the change, communication mediums may include:
        a. Meeting via teams
        b. Ping via teams
        c. In-person meeting

iii. Provide adequate training (if necessary)
    1. What skills and behaviors are required to achieve business results?
    2. Which training delivery methods will be most effective?

[6] Team, Editorial. "Change Management Plan: What Is It and How to Create It?" Bit Blog, February 3, 2020. Accessed October 21, 2021. https://blog.bit.ai/change-management-plan/

3. The team must be informed and taught of new knowledge and skills required to work efficiently as the change is rolled out

d. The CCB will **authorize** or **veto** the change according to the following:

AUTHORIZATION PROCEDURES:

1. Product Owner and Project Manager have the power to authorize or veto change requests
    a. These members may consult advisors before authorizing or vetoing any change request
    b. Internal team members can overturn these decisions based on majority consensus

e. The CCB will **measure** the impact of the change according to the following:
    i. Measure the Change Process
        1. Did the change help in achieving business goals?
        2. What could have been done differently?
        3. Was the change management process a success?

3. To track and organize change requests Swipet will use the following guidelines:

a. The CCB will track all changes using the Central Changelog
    i. Our central changelog will be in GitHub
    ii. Navigate to project *changelog*
        1. This will track and collect all performance changes
        2. Evaluate progress and success over time of each change

Budget**:**

The budget assesses the cost and fiscal responsibilities associated with Swipet. It will help consumers and stakeholders understand the value and expenses incurred when developing, implementing, and maintaining the project. When calculating the total budget for Swipet, our team factored in labor costs (internal and external), software/hardware costs (internal and external), and any other miscellaneous incurred costs.

To estimate labor cost, we set an hourly rate based on personal co-op experience and market research of average developer-related salaries. We then multiplied that with the total work effort or hours needed to complete the project. Our software and hardware costs include laptops, development IDE's, hosting services, server equipment, internet, external software integrations, and team collaboration/communication tools.

The project budget table below details specific budget considerations associated with Swipet and any information discussed above.

Table 19 Initial Cost Budget

Initial Cost - Upfront Budget				
		Units or Hours	**Cost per Unit or Hour**	**Total Cost**
Labor - IT	Ui/UX Designer	80	$20	$1,600
	Front End Dev	150	$25	$3,750
	Backend Dev	150	$30	$4,500
	Full-Stack Dev	200	$45	$9,000
	Project Manager	80	$35	$2,800
	Scrum Master	75	$30	$2,250
			Total	**$23,900**
Software - Internal	IDE - JetBrains	1	$60	$60
	Hosting Services	1	$65	$65
	Misc.	1	$100	$100
			Total	**$225**
Software - External	Domain	2	$30	$60
	Collaboration Tools	1	$25	$25
	MS Office - Family	1	$99	$99
	Misc.	1	$100	$100
			Total	**$284**
Hardware - Internal	Server Equipment	1	$1,000	$1,000
	Laptops	4	$1,500	$6,000
	Misc.			
			Total	**$7,000**
			Total Upfront Cost:	**$31,409**

Table 20 Ongoing Budget

Annual Cost - Ongoing Budget				
		Units or Hours	**Cost per Unit or Hour**	**Total Cost**
Labor - Maintenance	Ui/UX Designer	10	$20	$200
	Front End Dev	25	$25	$625
	Backend Dev	40	$30	$1,200
	Full-Stack Dev	50	$45	$2,250
	Project Manager	10	$35	$350
	Scrum Master	10	$30	$300
			Total	**$4,925**
Software - External	AWS	8760	$0.0138	$121
	Hosting Services	1	$65	$65
	Misc.	1	$100	$100
			Total	**$286**
Software - External	Domain	2	$30	$60
	Collaboration Tools	1	$25	$25
	MS Office - Family	1	$99	$99
	Misc.	1	$100	$100
			Total	**$284**
			Total Annual Cost:	**$5,495**

Problems Encountered and Analysis of Problems Solved:

Throughout the project, there were a few issues that we had faced. Below, we have listed some of these problems and the solutions that we came up with for them.

No elegant solution for an image carousel.

- We tried to find existing libraries to support this feature implementation, but we could not find a library that worked with our tech stack or was secure and not overloading unnecessary code. We went ahead and developed a standalone solution using ChakraUI.

Did not want to host static assets privately.

- Moved static assets to AWS S3 bucket instead of hosting on private machine.

Running out of GitHub actions minutes.

- Migrated to self-hosted runners instead of GitHub actions runners.

AWS CORS errors on S3 upload.

- Honestly, we are not sure what is going on with this problem, attempted to fix by making settings identical to a known working version but still refuses to upload. This issue remains unresolved as of now. Planning to resolve next semester.

Conclusion

People struggle to find the pet they want and love from animal rescue shelters and animal shelters struggle to find potential adopters for their homeless animals. Swipet provides an interface for potential adopters and shelters to connect digitally, eliminating the extensive search process previously performed by adopters. Swipet enables adopters looking for pets to browse through an array of available animals, suggested based on preferences, while having the most essential details of those pets available at the click of a button. Swipet provides shelters the means to promote their available animals through a dashboard interface where they can upload personalized details and images of their animals for general users to view and adopt. Swipet is a web-based application using cutting-edge technologies such as TypeScript, ReactJS, Node.js, Redis, PostgreSQL, and AWS that delivers innovative solutions to tackle concerns and needs of our users.

Over the last two semesters, we learned how to manage a software development project and developed necessary organization skills to collaborate and delegate work within a team. Navigating hybrid work environments, conflicting team schedules, and technical setbacks proved to be some of the biggest challenges the team faced, but with effective meetings, group work sessions, and extensive debugging the team was able to overcome the roadblocks and deliver a suitable digital solution to the problem at hand – Swipet.

References

Best Friends. “Pet Adoption: Barriers and Solutions | Network Partners.” Best Friends Animal Society, 2016. Accessed September 28, 2021. https://network.bestfriends.org/tools-and-information/roadblock-free/pet-adoption-barriers-and-solutions

Madigan, J. “Animal Rescue Shelters.” Oclc.org. 2020. Accessed 2021. https://my-ibisworld-com.uc.idm.oclc.org/us/en/industry-specialized/od5794/major-companies

“Number of Pets (Cats and Dogs).” oclc.org, 2018. Accessed September 19, 2021. https://my-ibisworld-com.uc.idm.oclc.org/us/en/business-environment-profiles/f306/business-environment-profile

Nuss, Allaire. “Animal Shelter in Germany Uses Tinder to Help Pets Find Their ‘Purrfect Match.’” Daily Paws, 2021. Accessed 2021. https://www.dailypaws.com/pet-news-entertainment/feel-good-stories/munich-animal-shelter-tinder-pet-adoption

Team, Editorial. “Change Management Plan: What Is It and How to Create It?” Bit Blog, February 3, 2020. Accessed October 21, 2021. https://blog.bit.ai/change-management-plan/

Turner, Patricia, Jim Berry, and Shelagh Macdonald. “Animal Shelters and Animal Welfare: Raising the Bar.” *The Canadian Veterinary Journal* 53, no. 8 (2012): 893–96. Accessed September 20, 2021. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3398531/