

SW Engineering CSC648/848 Spring2015

Project Name :	Pet Basket
Group Number :	03
Team Members :	Allan Gamboa (Team Lead) <i>agamboa@mail.sfsu.edu</i> Christopher Beaton <i>cbeaton@mail.sfsu.edu</i> Nathan Lee <i>nthanlee@mail.sfsu.edu</i> Sisi Wei <i>swei@mail.sfsu.edu</i> Stanley Seeto <i>sseeto@mail.sfsu.edu</i>

Milestone 3
04/30/2015

Revision History Table

Date	Version	Description	Author
04/21/2015	1.0	Changes in accordance to meeting feedback	
05/16/2015	1.5	Update UI Mockup	

1.Data Definition/Glossary

1. **Browser:** Non registered user
2. **Registered User:** Person who registered and has an account on the website (Not required to post a pet for listing)
 - a. **Poster:** Person who posts a pet into the animal listing for adoption
3. **Administrator:** Person responsible for image and content inspection.
4. **Pets:** Animals prior to being selected for adoption. Not eligible for adoption.
5. **Adopter:** Person interested in an adoptee(s).
6. **Verification Page:** Agreement page showing adopter recognizes special needs of pet and responsibilities of adopting a pet.
7. **Clickable Detail:** Function in the browsing page used to expand basic information without needing to load a different page
8. **Adoptee:** Pet targeted for adoption.
9. **Featured:** Pet targeted for special presentation in user interface
10. **Poster:** Registered user who is looking for a adopter for his/her pet.
11. **Detailed Pet Page:** Page that will consist of all of the **provided information** about a pet
12. **Provided Information:** Information provided by the **poster** of a pet. May consist of: Name, Age, Breed/Type of Animal, Weight, Any Special Needs, Comments, Size
13. **Pet Basket:** Cart functionality that allows all users to select pets and save for later side by side inspection.
14. **Collected Pet / Followed Pet:** Pet in the pet basket.
15. **Related Content:** Website content, such as articles, videos, ancillary to the main pet adoption focus.

3.Initial list of functional specifications

Priority 1

1. Administrators shall be able to approve what pets are shown.
2. Administrators shall be able to decline what pets are shown.
3. Administrators shall have control over registered user accounts.
4. Users shall be able to search by animal species.
5. Users shall be able to search by animal age.
6. Users shall be able to search by animal weight.
7. Users shall be able to search by animal breed.
8. Users shall be able to search by animal sex.
9. Users shall be able to search breeds through a drop-down menu.
10. Users shall be able to apply for adoption of any posted pet.
11. Users shall be able to see basic pet information on the browsing pages.
12. Users shall be able to expand the basic pet information on the browsing pages into a detailed information about a pet.
13. Users shall be able to access the detailed pet page by clicking the image.
14. Browsers shall be able to register.
15. Registered Users shall be able to submit postings of animals for adoption
16. All users shall have access to minimal **Pet Basket** functionality which allows users to add pets to a queue.
17. Registered users shall have access to all **Pet Basket** functionality.
18. Registered users shall be able to edit pets they have posted.
19. Registered users shall be able to edit their contact information

Priority 2

20. Users shall be able to search by pet name.
21. Users shall be able to contribute to a pet by providing money for purchasing food, toys, and other pet related products.
22. Registered Users shall be able to include any special needs within the submission form.
23. Administrators shall be able to issue warning messages to registered users for any misconduct of the site.
24. Administrators shall have control over what featured pets get shown on the homepage.

Priority 3

25. Users shall be given the option to sort their browsing list.
26. Pet recommendations shall be based on search.
27. The homepage shall display related pet news.

28. Users shall be able to sign up for the newsletter.

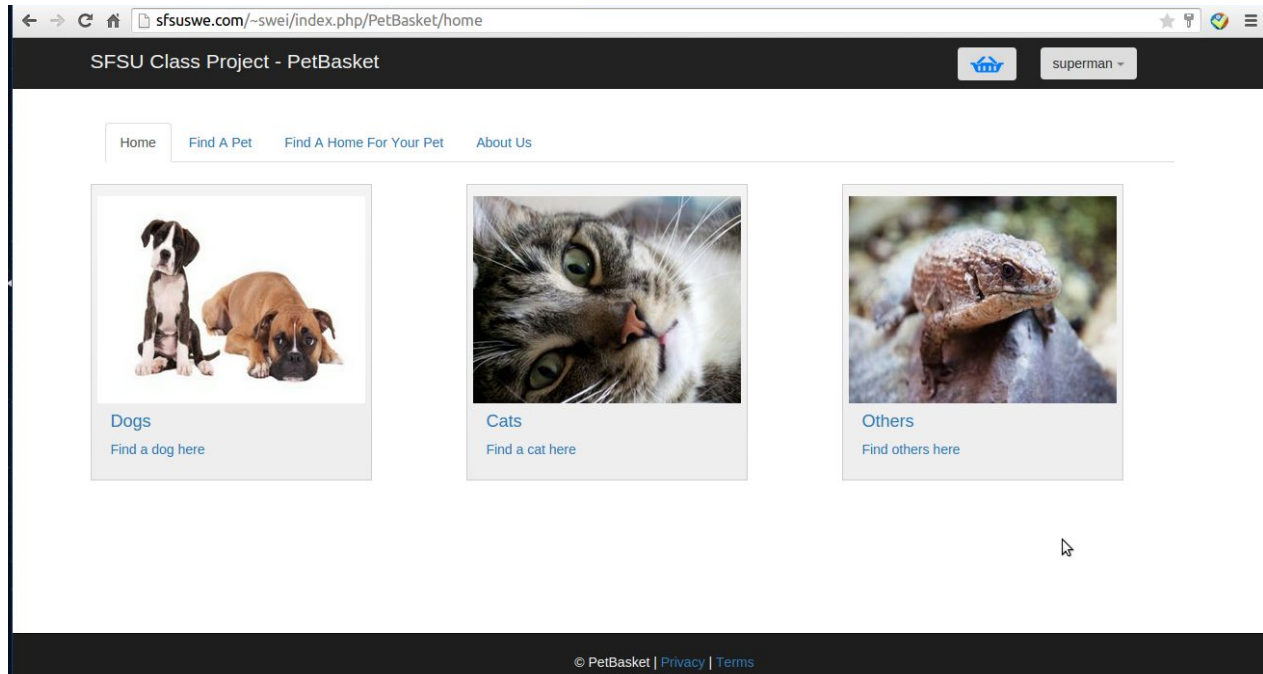
4. Initial list of non-functional specifications

1. The language used shall be English.
2. The uploaded pet images shall be viewed on one page in the detailed pet page.
3. The logo shall be on the upper left of all pages.
4. The search functionality shall be on the left side of the page.
5. We shall incorporate a horizontal navigation bar.
6. The website's about page shall prominently display CSC648 group project.
7. The website shall be able to host at most 50 concurrent users.
8. Users shall have secure account creation.
9. Users shall have secure login.
10. The website shall be safe.
11. The website shall provide friendly user experience.
12. The website shall be easy to use.
13. The website shall include Google Analytic specifications.

5. UI Mock Up and Story Board

5.1 UI Mock Up

- Homepage




- Login

← → ↻ 🏠 📄 sfsuswe.com/~swei/index.php/PetBasket/home?logout=true&referrer=home ⭐ 🔑 🌐 ☰

SFSU Class Project - PetBasket

Home Find A Pet

 Dogs
Find a dog here

Login to PetBasket.com

Already a member?

Email or Username:
superman

Password:

Login

Not a member yet?

Username: Alphanumeric characters only

Email: example@gmail.com
example@email.com

Password: At least six characters

Confirm Password:

☐ I am at least 18 years old and I agree to the [Terms and Conditions](#).

Register

© PetBasket | [Privacy](#) | [Terms](#)

- Browsing Page

SFSU Class Project - PetBasket

Home Find A Pet Find A Home For Your Pet About Us

Filter

Species: Dog

Breed: All

Gender: ☐ male ☐ female


Size: ☐ x-small ☐ small ☐ medium ☐ large

Age:

Reset Submit


Dog 5 results

Killer Contact the owner Add to basket



Breed: Affenpinscher
Age: 100
Sex: male
Weight: 150 lbs

Rocker Contact the owner Add to basket



Breed: All
Age: 55
Sex: male
Weight: 130 lbs

© PetBasket | Privacy | Terms Page 1


- Detail pet information Page

SFSU Class Project - PetBasket

Back To Search

Killer

Species: Dog Breed: Affenpinscher Weight: 150 Age: 100 Sex: male



Description:
The meanest dog in the world. BETTA RECOGNIZE

Special Needs:
Requires 20 lbs of raw beef twice a day. FOR REALZ

• Dog • Affenpinscher • 150 lbs • 100 Yrs • male

Contact the Owner Add to basket

© PetBasket | Privacy | Terms

- Posting Submission

SFSU Class Project - PetBasket



Login

[Home](#)

[Find A Pet](#)

[Find A Home For Your Pet](#)

[About Us](#)

Find a Home For Your Pet

Note: * Fields Required

Name: *	<input type="text"/>	Age:	<input type="text"/>	Weight:	<input type="text"/>
Species: *	<input type="text" value="Dog"/>	Breed: *	<input type="text"/>	Sex:	<input type="text" value="Male"/>

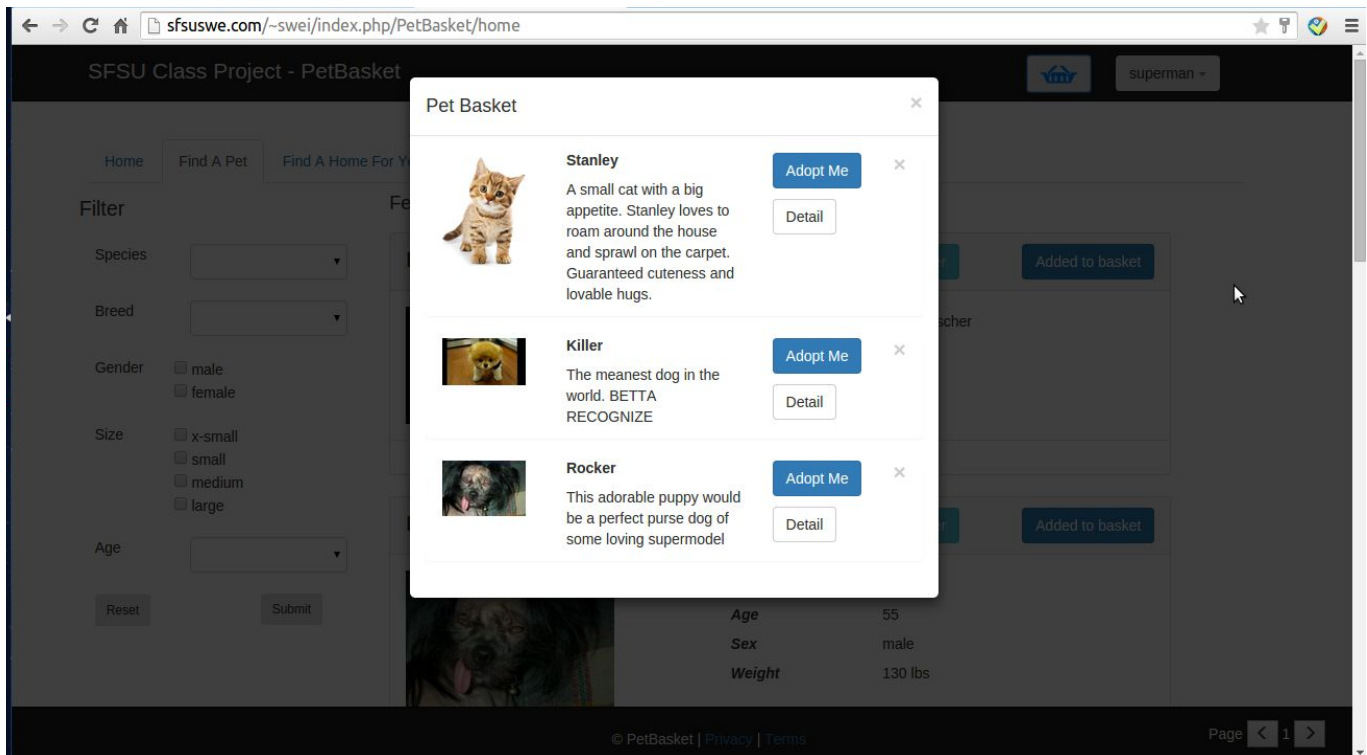
Description

Special Needs

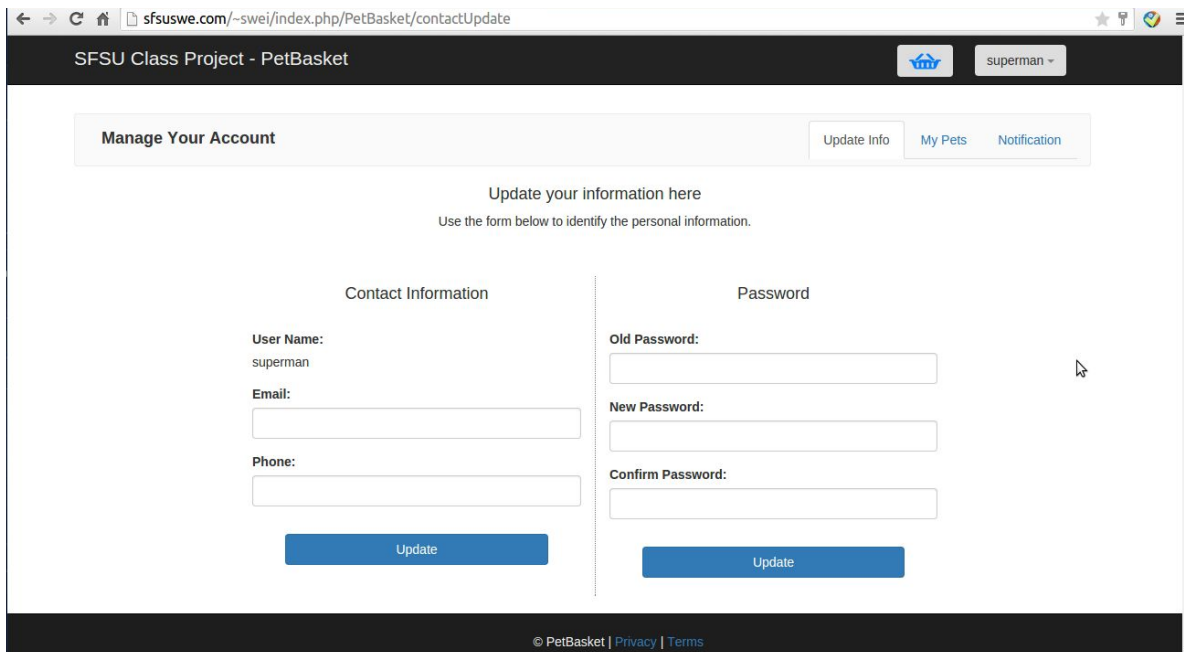
Select Image to upload:

No file chosen

- Pet Basket UI



- Edit the Poster's contact information



- Edit the Poster's past posts page


← → ↻ 🏠 📄 sfsuswe.com/~swei/index.php/PetBasket/myPets ⭐ 🔑 🌐 📄 ☰

SFSU Class Project - PetBasket 🏠 superman ▾

Manage Your Account [Update Info](#) [My Pets](#) [Notification](#)

Find your previous posts here
You can either edit your previous posts or add a new post in this page.

Your Posts: [+ New](#)

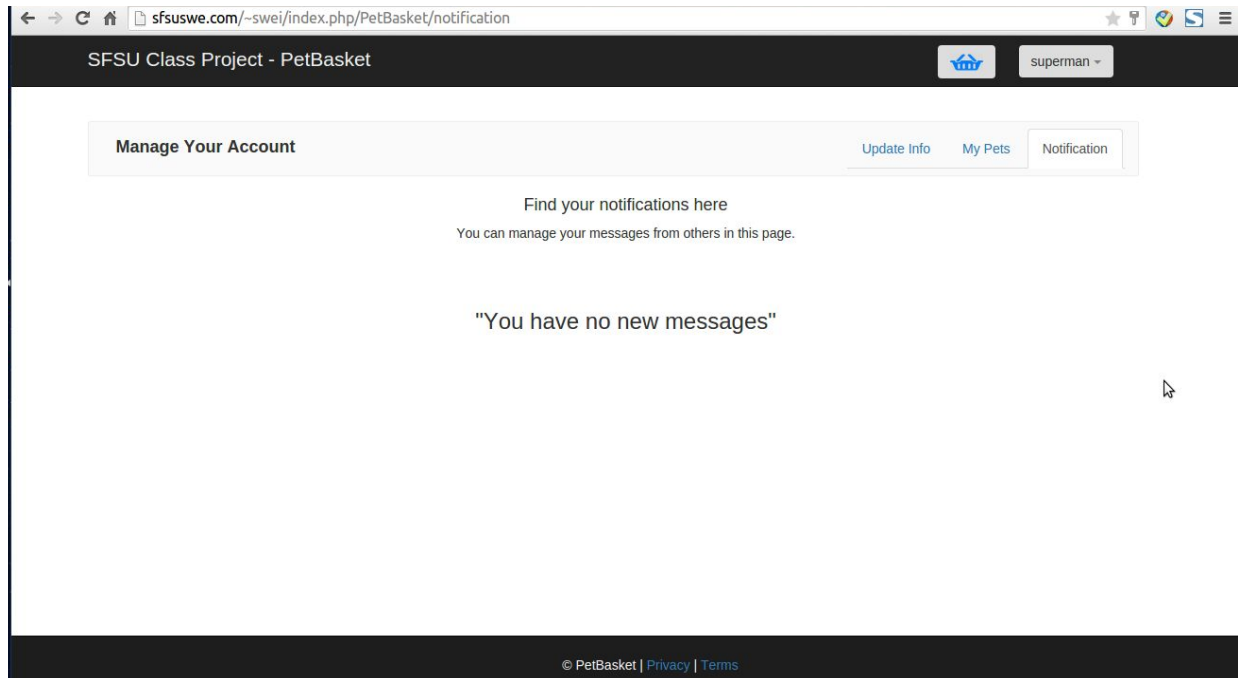


Date Posted: 05/16/2015
Status: Pending
Name: Sammy
Age: 6

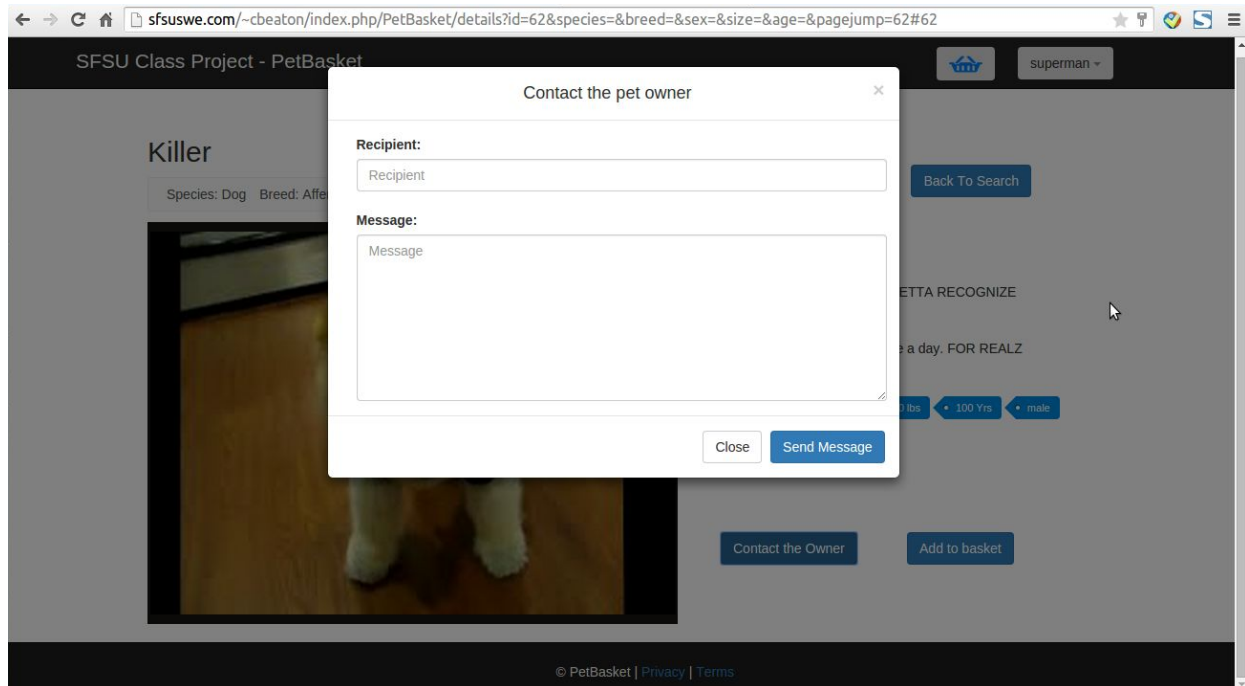
[✎ Edit](#) [🗑 Remove](#)

© PetBasket | [Privacy](#) | [Terms](#)

- Notification Page



- Contacting the Poster for Adoption



- Administrator New Page

Administrator Page : New

height size unit
according to length
of data entered

Administrator				
New		All		
New				
picture of pet	description	special needs	user info	
				① ✓
				② ✗

if reject
pet
cancel
cancel
message
to be sent
to poster

- ① if check mark clicked pet is added to DB along with user if user does not already exist
- ② pet is rejected

Administrator Page : All

Administrator				
	New	All		
All	Search Filter & order options			
Pet picture	date added	number of pending Admin requests sent	date poster last logged on	R
				F
		0		
		0		
		0		
		0		

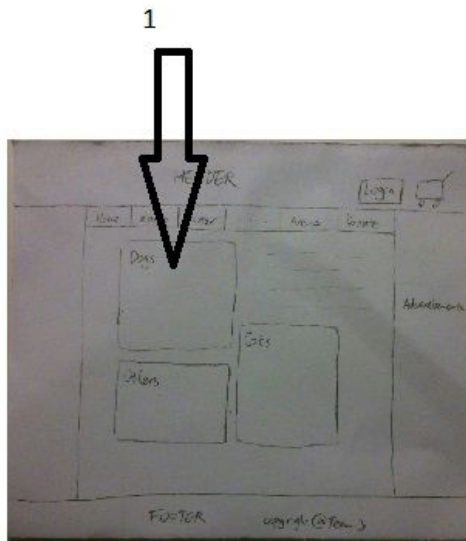
if R/F pressed message appearing again

↑
R: message to be sent to poster
F: Reason for featuring

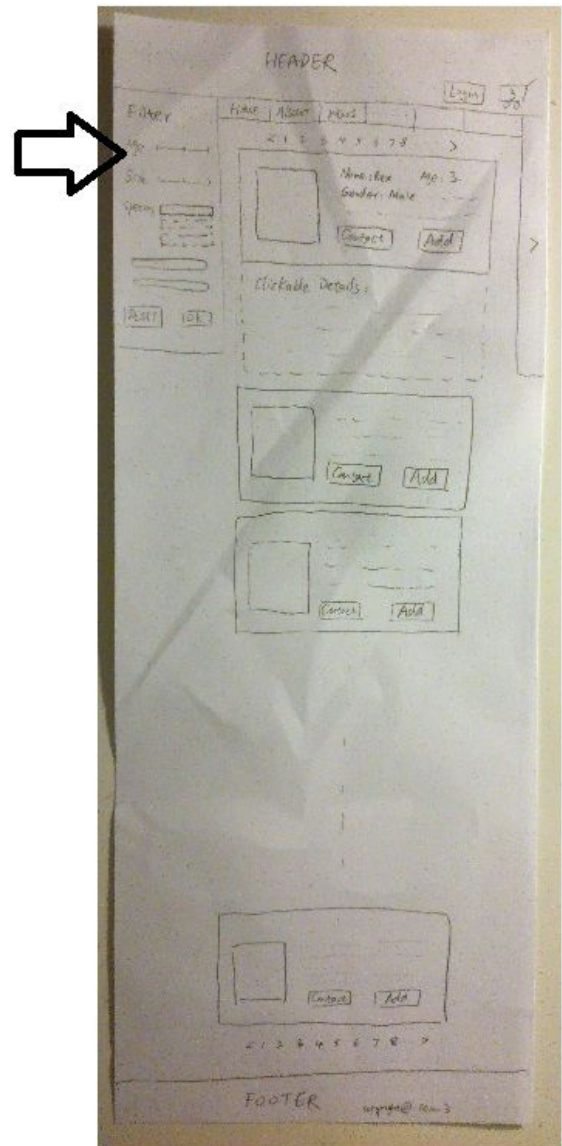
..if the new button is pressed the administrator is brought to the page containing all pets currently in our DB

5.2 Story Board

Functional Requirement #4 - 10:

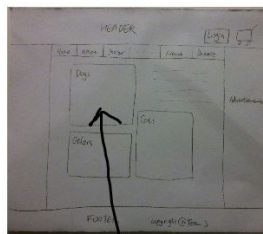


2

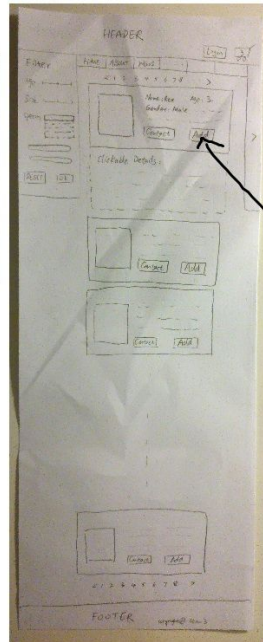


The user selects one of the subcategories of pets (1) which leads to a waterfall browsing page and uses the search (2) for animal type, animal age, animal weight, animal breed, animal sex, pet name, breeds and sub-breeds

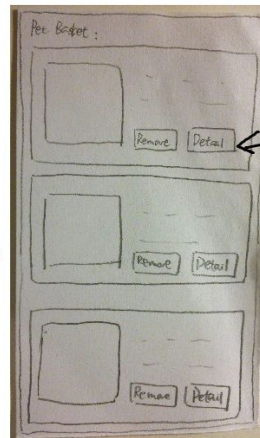
Functional Requirement #11:



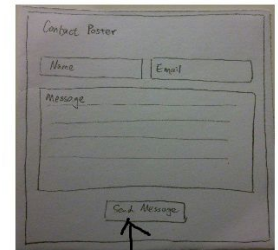
(1) User clicks subcategory



(2) User adds to his basket



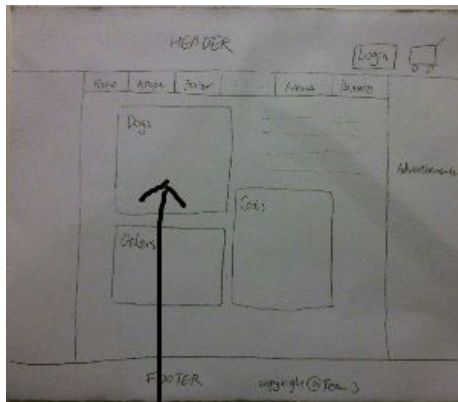
(3) Need to add apply button



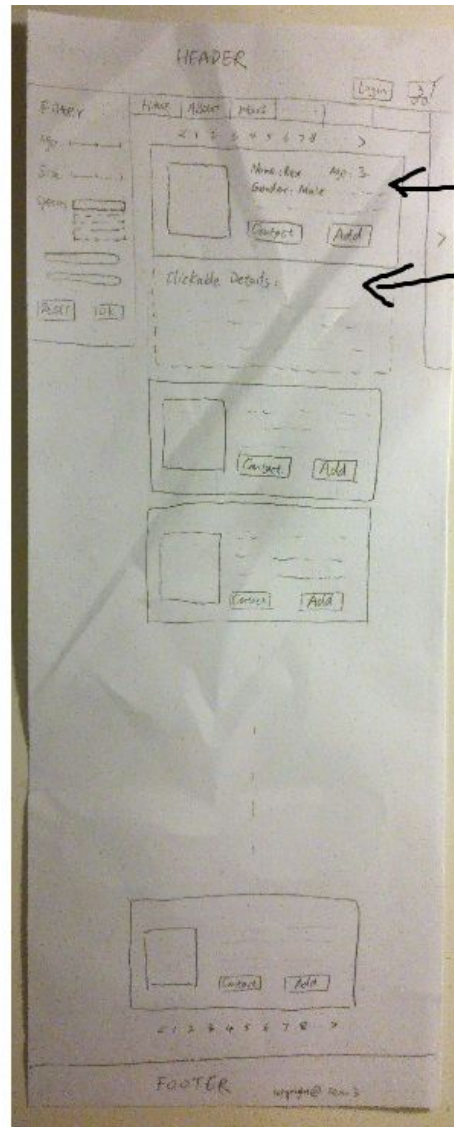
(4) Applies by directly contacting the poster

The user selects one of the subcategories to enter the browsing page, adds the pet to his cart, and applies for adoption by contacting the poster. Application does not need to go through the administrator since only registered users can request adoption. Registered Users agrees to being a valid person during registration and the administrators are not liable for any falsified information.

Functional Requirement #12-14



(1) User clicks subcategory

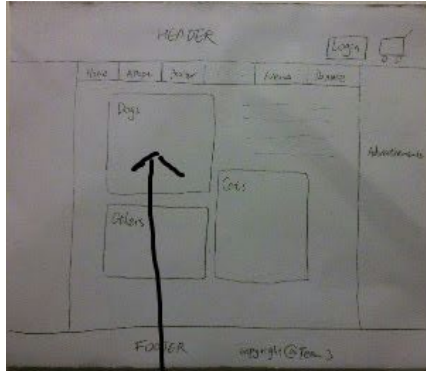


(2) User can see basic pet info on browsing page

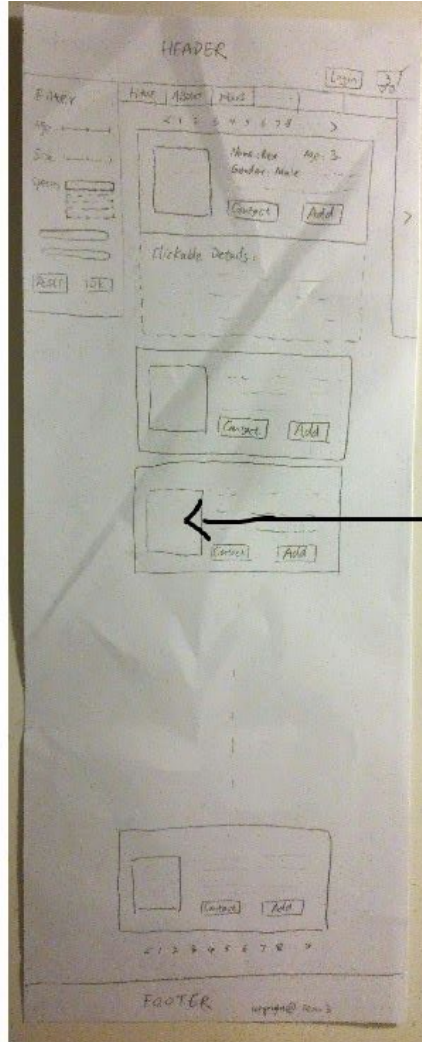
(3) User clicks to expand details

The user selects a sub-category (1) and is brought to the browsing page, where he can see the basic pet information (2) and click in to expand for additional details (3) in a drop down menu

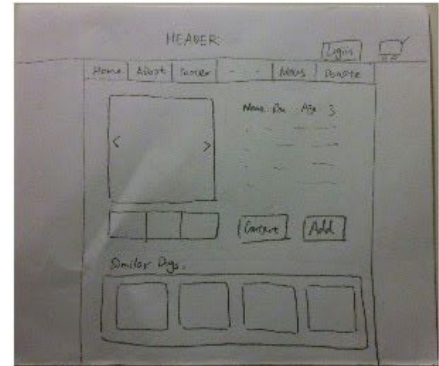
Functional Requirement #15



(1) User clicks



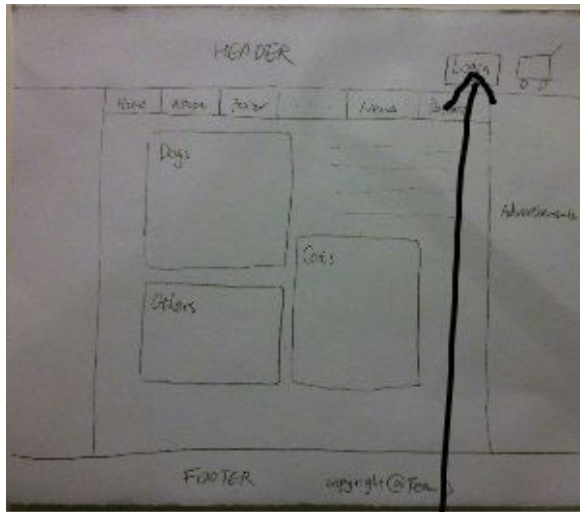
(2) User clicks



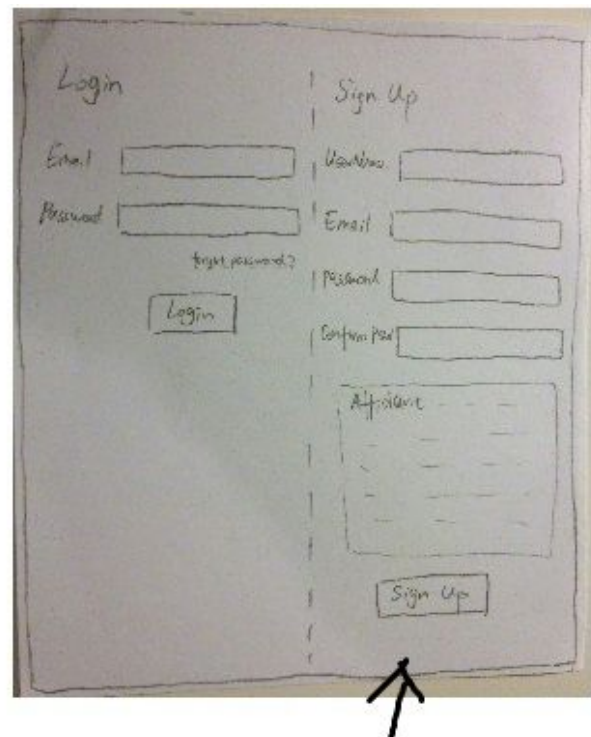
(3) Detailed pet page displayed

User clicks a sub-category and clicks the image to open the detailed pet page, which displays the description and any poster added comments.

Functional requirement #16



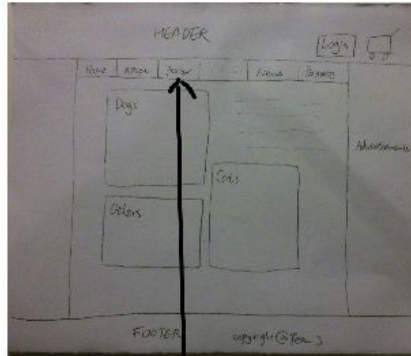
(1) User clicks Login button



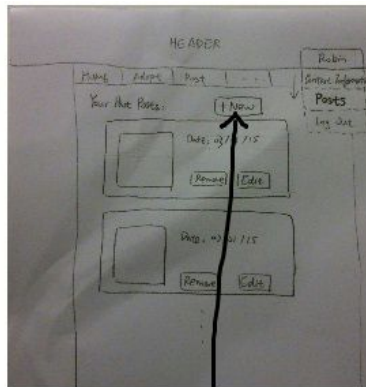
(2) User can register through sign up

User clicks login button and if he does not have an account, he can register an account by signing up.

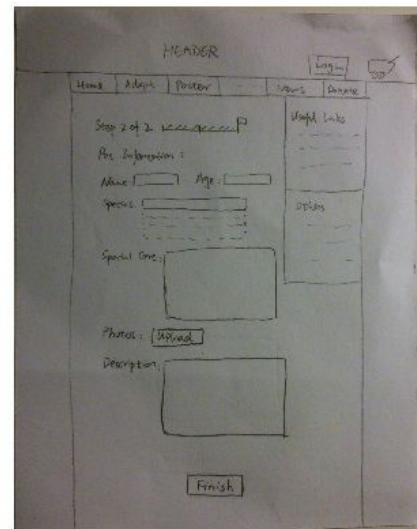
Functional Requirement #17 + 21



(1) Registered User selects



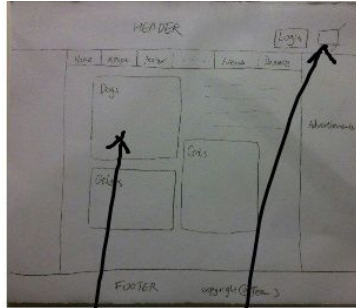
(2) Registered User selects New Posting



(3) Registered User completes form to submit a posting

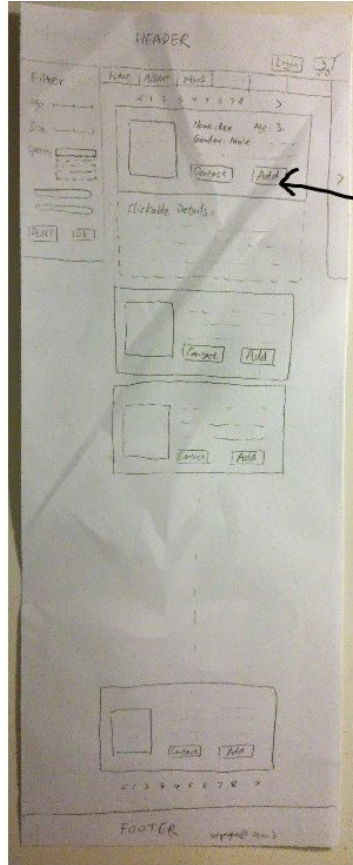
Registered User selects the poster's button to look into his postings. Registered User creates a new posting and completes the form and includes any special needs.

Functional Requirement #18 + 19

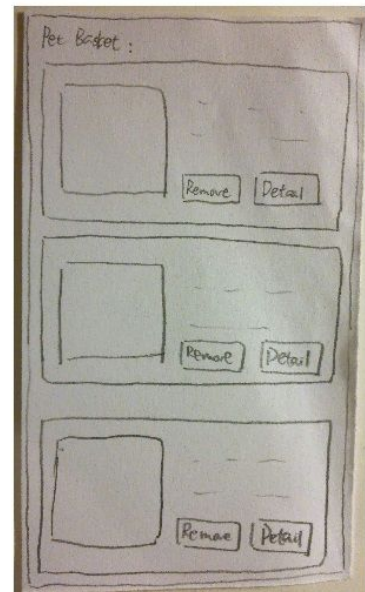


(1) User can look into his basket

(2) To add, he browses the pet



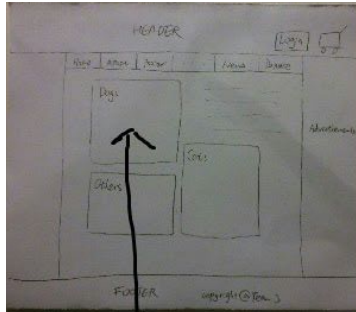
(3) Adds to basket



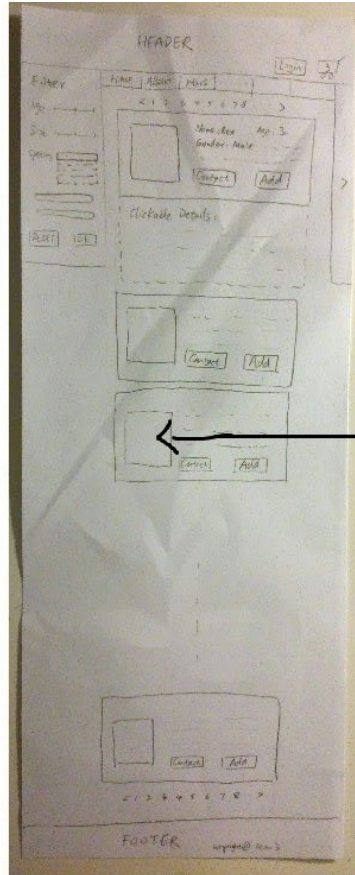
Pet basket display to remove/adopt

Users can add and look into their basket. The basket's information will only be stored when the user is a registered user.

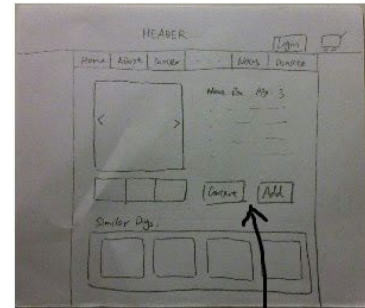
Functional Requirement #20



(1) User clicks



(2) User clicks



(3) Detailed pet page displayed

(4) Contributions would be a button located in this page

User can contribute from the detailed pet page, navigation is shown at requirement # 15

6.High-level system architecture

- This project will be hosted in Amazon Cloud.
- This project shall be hosted on a virtual machine running the Ubuntu 12.xx.x LTS operating system.
- This operating system shall run the Apache HTTP web server, MySQL database server, and PHP version 5.3.
- Database administration shall take place via phpMyAdmin and/or the MySQL command-line tool.
- The HTML and CSS portion of the Bootstrap framework shall be used, as shall the jQuery JavaScript library.
- User interactions with the project shall be measured using the Google Analytics API.
- This project shall support the Safari web browser versions 6 and 7 for OS X, the Chrome web browser version 41 for OS X and Windows 7, and the Firefox web browser version 36 for OS X and Windows 7.
- Remote administration of the virtual machine previously stated shall take place over SSH.
- This project will follow a Model-View-Controller design pattern.
- Primary project classes:
 - ❖ Pet
 - ❖ Image
 - ❖ Adoption
 - ❖ Basket
 - ❖ User
 - Member
 - Admin
 - ❖ Controller
 - PetBasketController
 - ❖ Model
 - Pet
 - Image
 - Adoption
 - Basket
 - User
 - ❖ View
 - ❖ Db
 - ❖ Uri
 - ❖ Route
 - ❖ Input
 - ❖ Response
 - ❖ Exception

- Data Organization

Users posting pets for adoption will upload text and image data that will be used to create searchable pet listings. Users will submit information describing the animal they're posting for adoption, such as the breed, size, age, special needs and description of the animal. Users will also be able to upload between zero and twelve images per pet. Text data will be stored and served in UTF-8 format. Users will only be able to upload images in png, jpeg, and gif format. Images shall not exceed 20mb in size. Images exceeding 1mb in size will be resampled to a dimension resulting in a file 1mb or less, and the original uploaded image will be discarded. A large, medium and small thumbnails will be generated for each uploaded image. Metadata describing the location, size and name of each thumbnail will be kept in json and stored along with the original image metadata in the image database table. All image files will be kept on the filesystem.

- Search Architecture

The search algorithm shall take as input an enumerated type of pet breed or all breeds, a specific size (enumerated - small, medium, large) or all sizes, a boolean value indicating preference for pet listings with images (true) or with and without images (false). These parameters shall be used to construct a SQL statement that will be used to query the database for results.

- Pet Basket

When a member collects a pet into their pet basket, the ids of the pet and user will be stored in the basket table. This information will be used to populate user's pet baskets upon login. The user id of a non member shall be set and retrieved using the session token if a session cookie can be set.

7.1 Data Base Organization

- **message Class/Table:** register users shall be able to message each other in regards to a pet of interest.

id	sender_id	recipient_id	thread_id	message_time	visibility
primary key - integer - auto-increment	identification of sender of message	identification of recipient of message	identification of message conversation	time message creation	boolean that determines if the recipient can see the message
Message					
content					

- **Adoption Table:** This table is to maintain a record of adoptions in process after a user has submitted their verification to the poster. The **id** field keeps track of the current adoption. The **pet_id** keeps track of which pet is being adopted. The **user_poster_id** keeps track of the id of the poster and the **user_adopter_id** keeps track of the potential adopter. **Timestamp** keeps a record of when the entry is added.

id	pet_id	user_poster_id	user_adopter_id	timestamp
integer auto_increment	id of the pet being adopted	id of the poster	id of the adopter	CURRENT_TIME STAMP

- **Image Table:** This table stores meta data about uploaded images.

id	name	file_name	file_size	width	height
integer auto-increment	varchar user supplied file name	varchar system file name	integer byte size of file	float width of image	float height of image

description	thumbnails	created	updated		
varchar user supplied image description	varchar meta data of thumbnails generated from image	integer unix time in seconds	integer unix time in seconds		

- **Pet Table:** This table stores meta data about pets either currently or formerly available for adoption

id	name	description	weight	breed	age
integer auto-increment	varchar name of the pet	varchar description of the pet	float weight of the pet	varchar breed of the pet	integer age of the pet
userId	adoptionId	created	updated	visibility	
varchar id of the user who put this pet up for adoption	varchar id of the adoption event	integer unix time in seconds	integer unix time in seconds	boolean true if pet is not deleted false if pet is deleted	

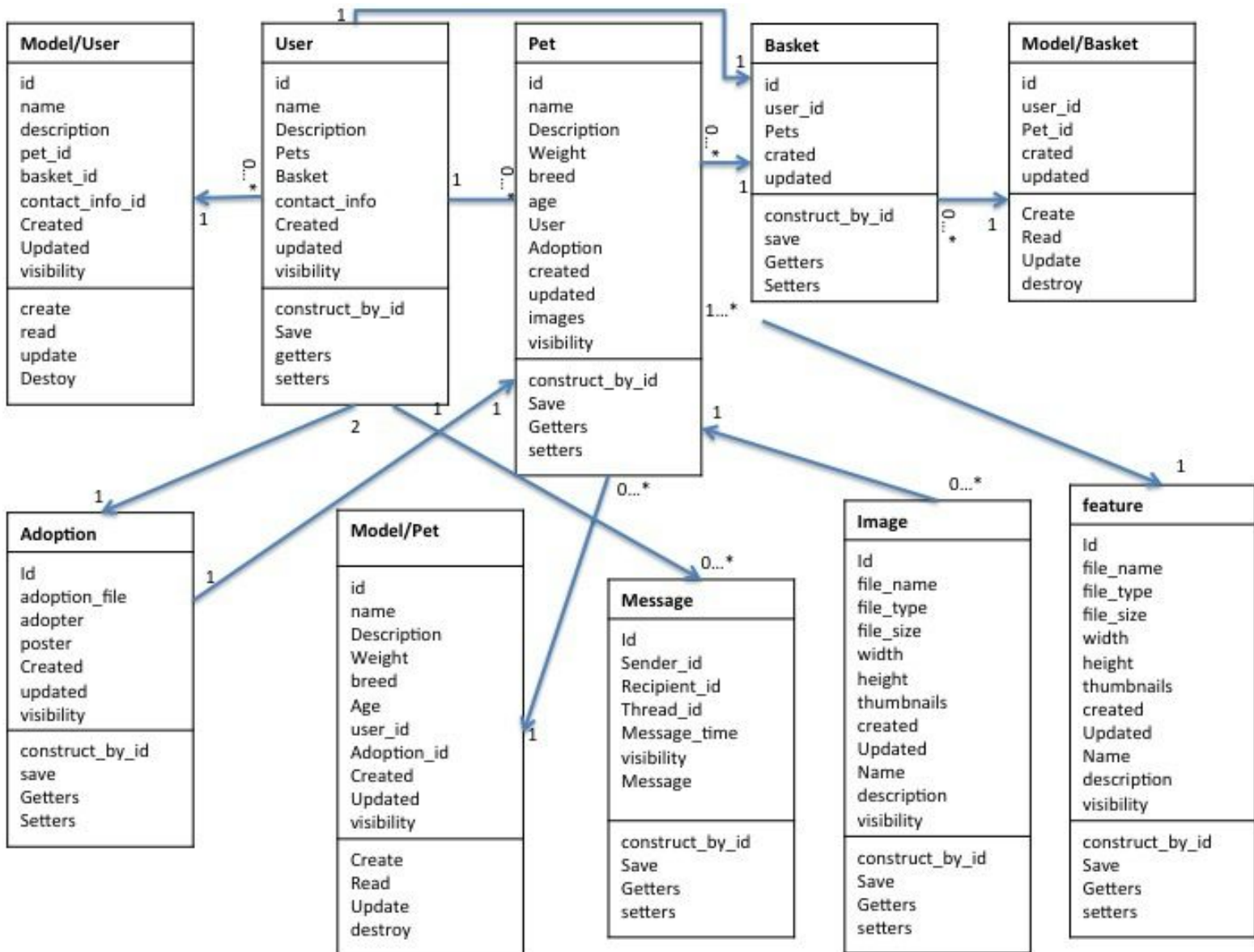
- **User Table:** This table stores meta data about registered users and administrators

id	username	password	email	created	updated
integer auto-increment	varchar name of user	varchar password of user	varchar email of user	timestamp	timestamp
is_admin					
boolean true if admin false if regular user					

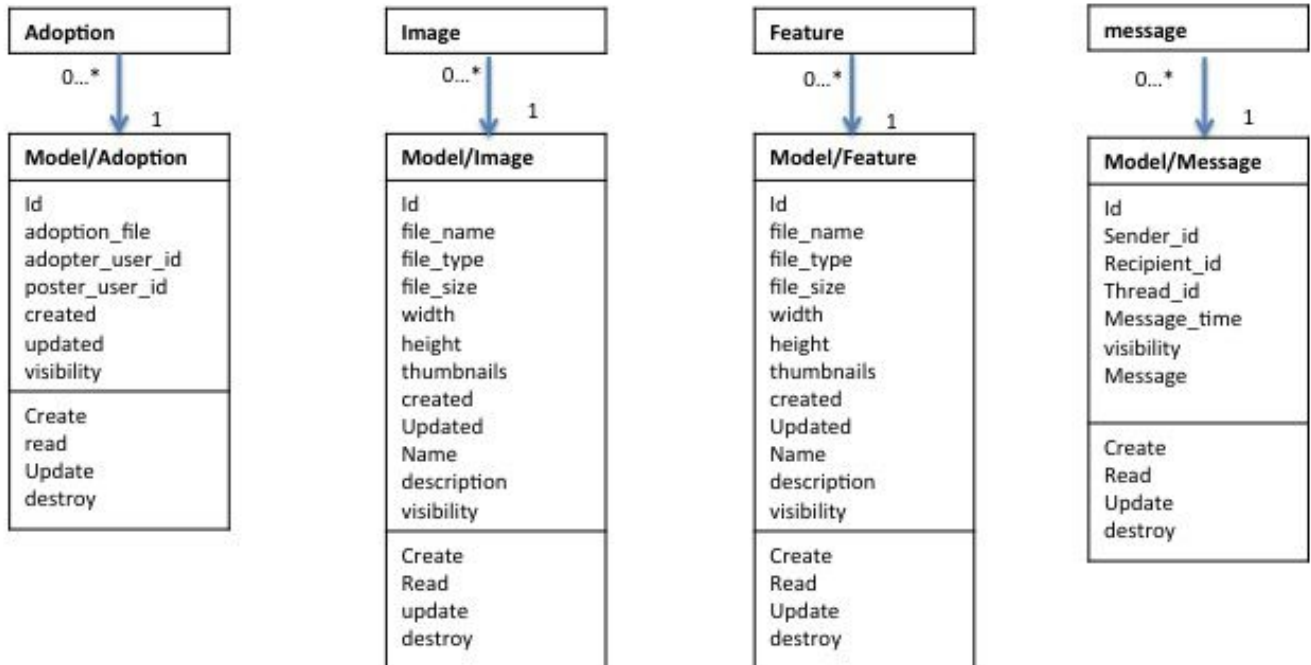
- **Basket Table:** This table stores data about what pets are stored in a user's basket

id	user_id	pet_id	visibility
integer auto-increment	varchar id of user	varchar id of pet in the user's basket	boolean true if in basket false if it gets deleted if row exists and is false, set to true if re-add the pet

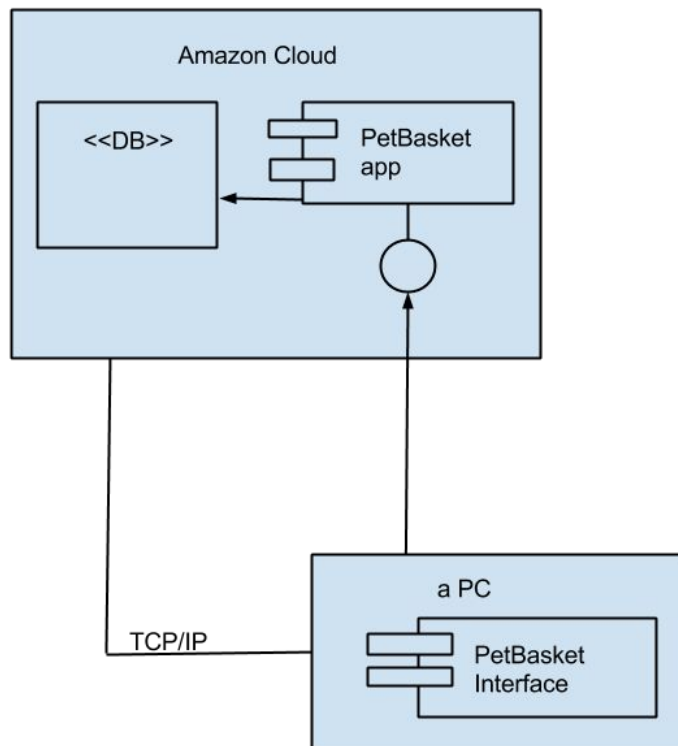
8. UML class diagram



8. UML class diagram (continued)



- Component & Deployment diagram



9. Key Risks

- Skill Risk: There is lack of web development experience for most group members. Milestone 0 has begun to address this risk. Continual work on the programing portions of the following milestones will continue to address this risk. Our tech lead is very knowledgeable in all aspects of web development and is very available when needed to resolve such issues.
- Legal/Content risk: We must obtain image and other related content for our site. This risk will be addressed by continual search of legally obtainable content.
- Organizational Risk: It will be hard to manage a project of this size without prior experience and knowledge. We will need to communicate constantly and keep each other updated on new ideas and keep track of weekly objectives.

10.Team

Allan Gamboa - Group Leader, QA: Organizes group meetings.Distributes tasks among group members.

Christopher Beaton - Tech Lead: Main consultant in structure of software. Main developer of software.

Nathan Lee - Database/SVN Administrator: Manager of proper SVN directory structure. Policies proper submission and retrieval policy.

Sisi Wei - Front End Development, User Experience: UI layout developer.

Stanley Seeto - Back End Development, Documentation Manager: Along with Christopher Beaton development of key classes and functions in software package.