

# SW Engineering CSC 648/848 Fall 2019

## Team Number 5

### CATDOG

Tahar Touati - ttouati@mail.sfsu.edu

Amir Anjomshoaa - Aanjomshoaa@mail.sfsu.edu

Ivan Briseno - ibriseno@mail.sfsu.edu

Meliss Nguyen - mestrada7@mail.sfsu.edu

Xiaopeng Rong - xrong@mail.sfsu.edu

## Milestone 5

URL of the demo: <http://34.67.160.125/>

Date: 12/15/2019

# **Table of Contents:**

- Product Summary
- Milestone 1 V2
- Milestone 2 V2
- Milestone 3 Feedbacks
- Milestone 4 V2
- Screenshots of Product
- Screenshots of Key DB Tables
- Screenshots of Tasks Management Systems
- Team Members Contribution
- Post analysis – lessons learned

# 1. Product Summary

**Product Name:** CatDog

**Product Url:** <http://34.67.160.125/>

## List of Priority 1 functional requirements:

1. Guests shall be able to sign up and create an account on the website:
  - 1.1. Create account as worker.
  - 1.2. Create account as Client.
2. Guests shall be able to send message to support team admin.
3. Guests shall be able to access a demo tour of the website.
4. Guests shall be able to skip the demo.
5. User shall be able to sign into the website.
6. User shall be able to get support from admin.
7. Clients shall be able to create Pet profiles.
8. Clients shall be able to create posts.
9. Client should include the date period for the post
10. Clients shall be able to create Pet profile in sign up.

### All the registered users:

11. User shall be able to log out from the website.
12. User shall be able to get support from admin.
13. User shall be able to change his/her password.
14. Clients shall be able to see who booked the post.
15. Clients shall be able to use search bar:

Filter workers by booking number.

Filter workers by username.

16. Workers shall be able to accept to book a client's post.
17. Workers shall be able to use search bar to search for available jobs in location
18. Worker shall be able to see the available post to book.
19. Administrators shall be able to sign in to the website.
20. Administrators shall be able to log out from the website.
20. Administrators shall be able to delete/ban user.

## **Product Summary:**

CatDog is a website that provides pet sitting by accredited sitters. Our website is targeted to two types of users, the first type are pet owners who frequently travel and need a service that provides pet care during their travel duration, and the second type are accredited sitters and pet-friendly people wanting to earn extra money while taking care of a pet. This is the main focus of our market for catdog.

Users with a client account will be able to create pet profiles for their pets, create a post that seeks a pet sitting service from workers by notifying them and includes the date period and location for the service, select a sitter from those who booked his post and finally confirm the booking.

Users with a sitter account will be able to search clients by location, book a client's post from the available posts or from the post feed, get notified when a client accepts them.

This is the URL of our website: <http://34.67.160.125/>

# SW Engineering CSC 648/848 Fall 2019

## CATDOG

### **Group Number 5**

Tahar Touati - ttouati@mail.sfsu.edu

Amir Anjomshoaa – aanjomshoaa@mail.sfsu.edu

Ivan Briseno – ibriseno@mail.sfsu.edu

Melissa Estrada-mestrada7@mail.sfsu.edu

Xiaopeng Rong-xrong@mail.sfsu.edu

### **“Milestone 1”**

09/30/2019

| History Table     |                 |
|-------------------|-----------------|
| Version           | Submission date |
| Milestone 1 V 1.0 | 10/02/2019      |
| Milestone 2 V 2.0 | 10/30/2019      |

## Table of Contents

|   |    |
|---|----|
| 1-Executive summary: .....                                    | 3  |
| 2-Main Use Cases: .....                                       | 4  |
| 3-List of main data items and entities:.....                  | 9  |
| 4-Initial list of functional requirements:.....               | 11 |
| 5-List of non-functional requirements: .....                  | 13 |
| 6-Competitive Features Analysis: .....                        | 15 |
| 7-High-level system architecture and technologies used: ..... | 16 |
| 8-Team: .....   | 17 |
| 9-Checklist:.....   | 17 |

# **1-Executive summary:**

San Francisco is the Mecca of the tech industry that is full of working-class professionals who have pets as companions. Here at CatDog, we present a service that provides the personal loving pet care that owners themselves would provide in their own home. Our business ensures that a pet owner can take business trips or vacations without the stress of traveling with a pet and feeling secure in knowing their pets are in good hands.

Here at CatDog, we strive to create a network that bridges animal lovers together while providing a service that lacks in the market. As animal lovers ourselves, our headquarters in San Francisco allows us to target the audience and demographics of working-class individuals that are on the go and in need of a personalized and ethic service for their pet companions. We understand the demand for this service in San Francisco.

To emphasize the quality of pet care we will provide and the availability of our service it is important to consider the quality of life San Francisco residents live themselves and share with their pets. Our service is focused on the richness and aid animals provide their owners on a day to day basis, while understanding that our clients have busy schedules but would still revel in knowing their pets are in good hands while they are away. By allowing our clients to continue to invest in their pets with affluent care and service that is provided with a certified staff.

Currently, there are businesses offering pet sitting in San Francisco, but none offer a simple yet trusted service. Most of these pet sitting services care for several pets at once from different owners which causes an issue. Here at Cat Dog, we offer a more compatible and individualized pet care experiences. By relieving our clients of all the middle work, that we take on instead by having insured and bonded employees. We aim to make every process of working with our company as easy as possible starting with our application.

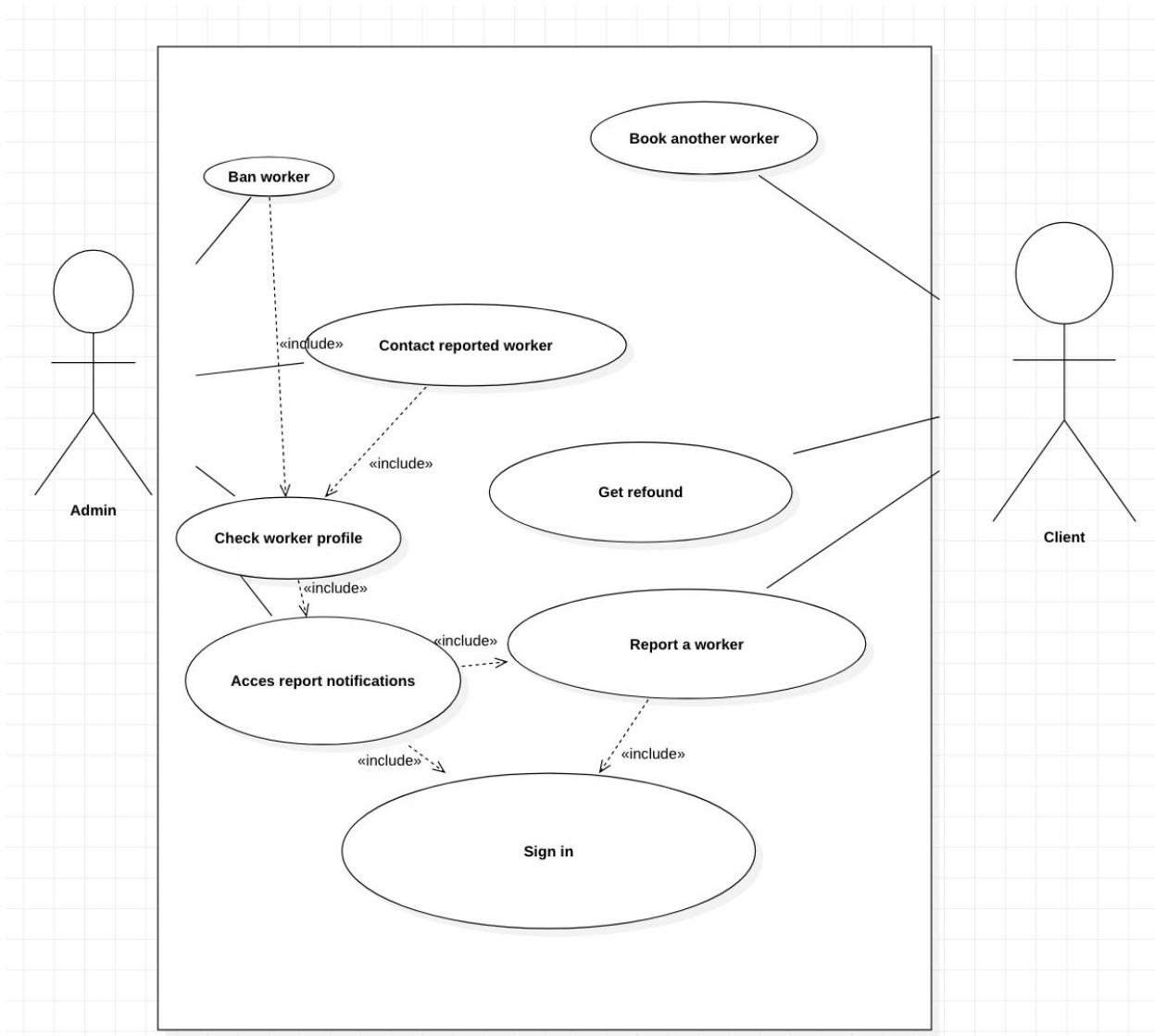
CatDog is hassle-free and simple, we provide thoroughly vetted sitters; making the safety of pets our highest priority. It is slick and streamline with a minimal approach to our user interface. In this interface, we put our focus on pets by building a specified animal profile displaying the needs of the pets. Forming a more personalized pet sitter experience. We are focused on generating a fast, secure, and efficient application.

## 2-Main Use Cases:

### Admin - Privileges/Technical Support/Customer Support:

**Mikael** receives a notification marked as “Urgent”. **Mikael** logs into his internal company email and notices a report message from **Julio** reporting **Joe** for a “No Show at Specified Time”. This is **Joes’** first “No Show” out of two allowed. Were **Joe** to receive a second “no show”

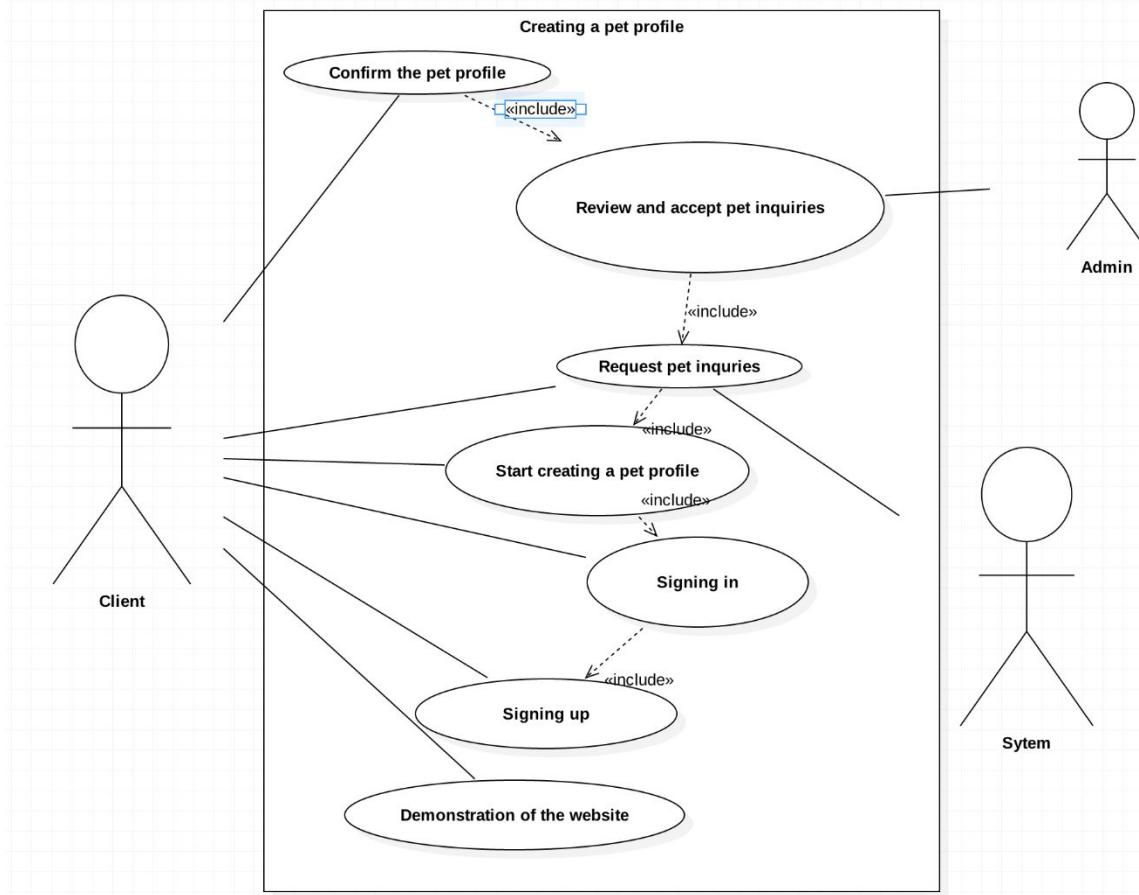
**Mikael** will enforce a ban on the worker and remove his profile. **Mikael** accesses the internal database to retrieve **Joes’** information and proceeds to contact **Joe**. **Mikael** then proceeds to contact **Julio** regarding the situation and gives him the option of cancelling the booking completely in which case the money will be returned, or try to re-book another worker that could be available.



Use case diagram 1: Admin - Customer Support

### **Client (Creating Pet Profile):**

**Julio** has a busy schedule and travels a lot due to work and usually has friends or family care for his cat **Sparkles**. One of his friends mentions this site, CatDog, where one can book pet sitters that work around any schedule. After exploring through the sites main page, accessing a quick demo showing how the sitters services work and reading some of the client reviews on their experiences with the sitter service as well as browsing over some of the accredited pet sitters and their stories, **Julio's** interest has now peaked and he decides to take the next step and create a free client profile registering as a **new user** to find a sitter for his cat **Sparkles**. **Julio** proceeds by creating a **new account**, which will, upon completion, enable him to book a sitter, by filling out all the required contact information. **Julio** has finished signing up as a **new user** and is now redirected unto what will be his **user page (his home page)**. **Julio** can now create his "**Pet Profile**" which will contain a relevant information on his pet. **Julio** proceeds to make a profile for his cat **Sparkles** answering some basic questions. **Julio** has completely finished creating the profile and he is now ready to save it and make it **visible to workers**.



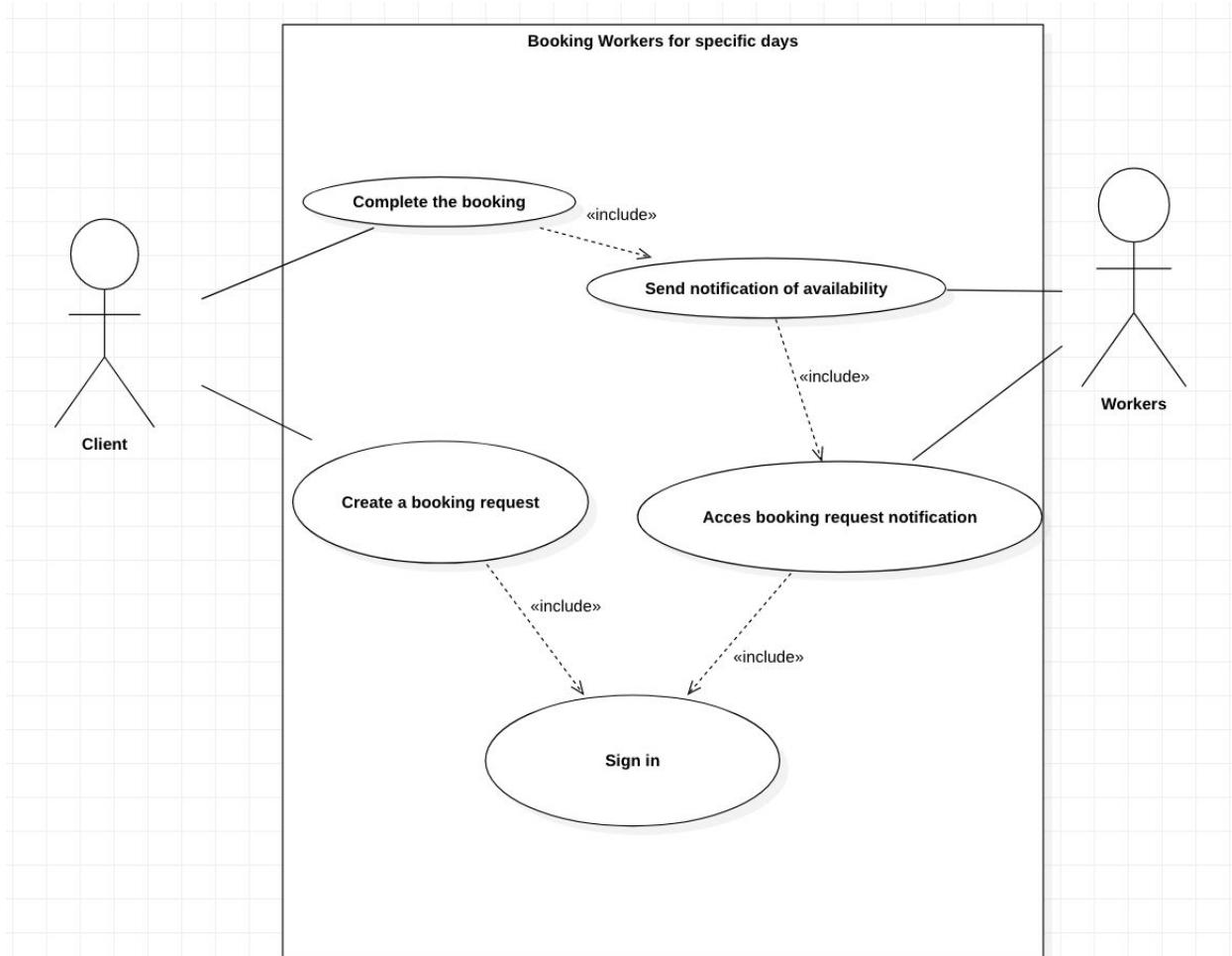
Use case diagram 2: Create a Pet Profile.

### **Client (Booking Worker for Specific Days):**

**Julio** has an important work trip coming up in a week and needs someone to look over his cat **Sparkles** for three days. **Julio** logs into his client account and proceeds to the option of “create a booking request” specifying the three days he will be gone. After completing the required booking information and agreeing to the “Terms & Conditions”, **Julio** can now receive notifications from workers who are interest and available. **Julio** receives a notification from **Joe** saying he are available for the specified dates and proceeds to navigate to **Joe’s** profile to look over his accreditation and read over some of the reviews about **Joe**. After going over **Joe’s** profile, Julio is convinced and decides to complete the booking with **Joe**.

### **Worker (Request Client Booking):**

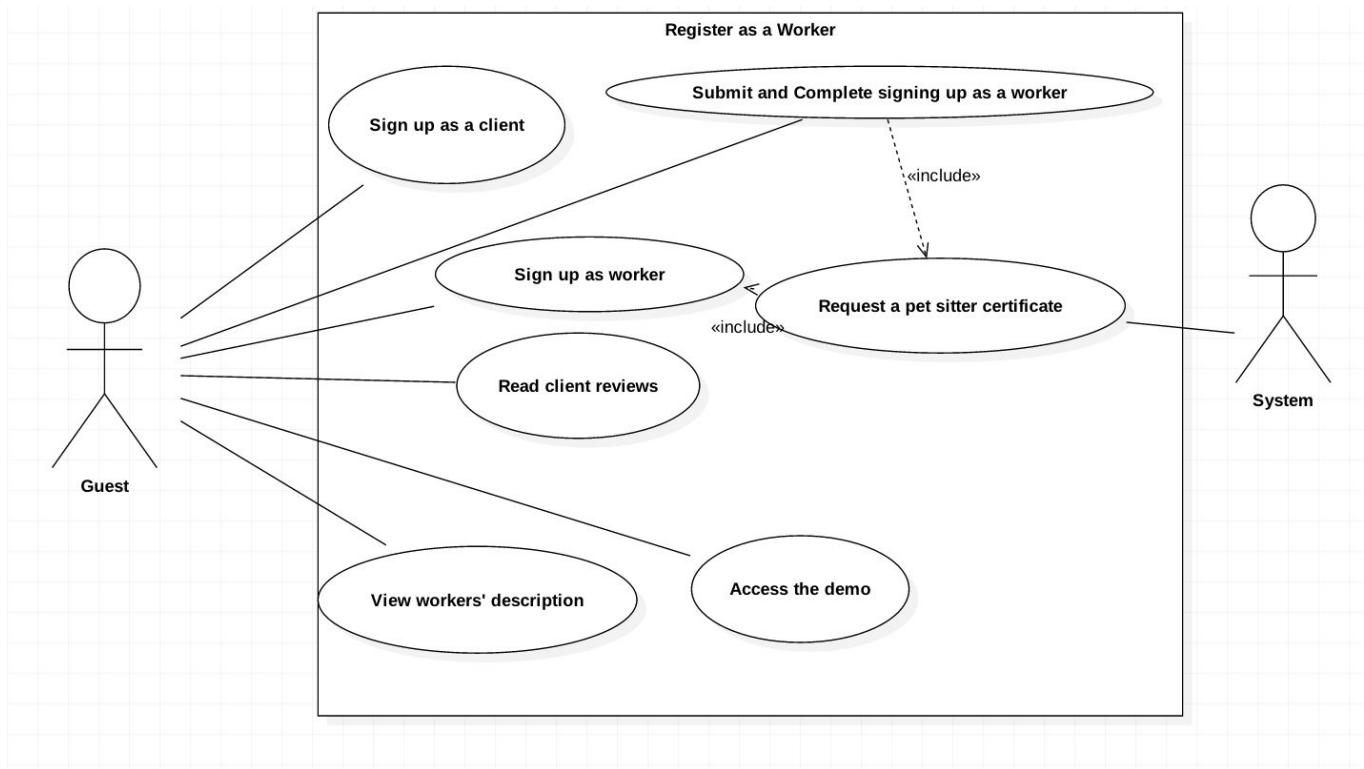
**Joe** has some free time since it’s summer and school’s out so he decides to change his availability on his worker profile to “Open” that will notify him of any booking requests that is coming up. **Joe** receives a notification on a client who will be gone for three days and is requesting a pet sitter for a “feline/cat” for those days. After looking at the clients “Pet Profile” **Joe** decides to “Request This Booking” then sends a message to the client stating that **Joe** is available for the specified dates and is requesting the booking.



Use case diagram 3: Worker (Request Client Booking) / Client: Booking Worker for specific days

### **Guest (Register/Sign Up as Worker):**

**Conny** attends college 4 days out of the week and has afternoons, evenings and nights free so **Conny** concludes that some extra income and working around her own schedule would be nice. **Conny** hears about the CatDog website and decides to browse the CatDog website. After viewing some of the clients reviews and reading about the company benefits at CatDog, **Conny** is convinced and decides to “Sign Up” as a worker. After clicking the “sign up” link/button **Connie** is presented with two options, one is to sign up as a client while the other option is to sign up as a worker. Connie chooses the worker option and is redirected to the worker form, where she fills out all the required information. Before being able to proceed with the application Connie is presented with a message regarding proper certification requirements. Stating that in order to proceed with the worker application the user must have a “pet sitter certificate” from an accredited source. After receiving the certificate Connie is now able to finish registration and submits the form agreeing to a background check, rules & regulations form and signing with an e-signature. With the form submitted all Connie needs to do is wait for a response from CatDog, accepting or denying the request.



Use case diagram 4: Guest Sign Up as a worker

### **3-List of main data items and entities:**

**Admin:** A user chosen by the owners of CatDog team that has access to all users contact information and booking information overseeing and managing the users.

**Registered User:** A Client or Worker who has signed up into our website with proper information and has access to different functionality depending on his role.

**Client:** A registered user who is looking for a worker for a pet sitting service. The Client has access to contact only the worker who requested to book a client.

**Worker:** A registered user who provides pet sitting service. Has access to search for clients based on the availability and location.

**Guest:** An unregistered user who does not have an account on the website. User can view some of the posts on the site and the functionality of the application. But guest does not have access to contact or post anything.

**Pet:** A client's pet that potentially need to be taken care of by the worker.

**Post:** A client's post seeks a pet sitting service from workers by notifying them, and it includes the date period and location for the service.

**Post a review:** Clients evaluate and score the pet sitting provided by workers after the job is done.

#### **Report:**

- **Report a Worker:** Client writing a report to send to an admin about the worker if any misconduct or policy violation happened during the job
- **Report a Client:** Worker writing a report to send to an admin about the Client if any misconduct or policy violation happened during the booking or the job.

**Book a Client:** The worker would book a client's post, and it notifies the client for potential workers.

**Demo:** An interactive tutorial for Guest users to be able to get an understanding of how clients and workers are using the website.

**Support:** Service to help all levels of users on the site in case of any help the user needs or if any problems occurred.

**Profile:** Information about the user, and available to both Client and Worker.

**Client:** Required information needed about the Client's pet and a short bio about the pet itself.

**Worker:** Required information about the Worker itself, and a short bio to inform or explain themselves to the Clients.

**Pet inquiries:** This is mainly used by the worker as help book in case of any problems. Required information for setting up the pet profile for the clients. Client needs to answer some question in regards of the pet's health insurance, diet restriction, pet interactions with other dogs and any other information that the worker needs in case of any problem with the pet.

**Certification:** Workers needs to have valid Pet Sitting Certification which can be acquired by any valid institution in order to sign up and work on our site.

# **4-Initial list of functional requirements:**

## **Guest:**

1. Guests shall be able to sign up and create an account on the website.
2. Guests shall be able to access a demo tour of the website.
3. Guests shall be able to get support from admin.
4. Guests shall be able to read clients reviews.
5. Guests shall be able to skip the demo.
6. Guests shall be able to use search bar.

## **Client:**

7. Clients shall be able to sign in to the website.
8. Clients shall be able to log out from the website.
9. Clients shall be able to get support from admin.
10. Clients shall be able to create posts.
11. Clients shall be able to book a worker.
12. Clients shall be able to review a worker.
13. Clients shall be able to create Pet profiles.
14. Clients shall be able to delete Pet profiles.
15. Clients shall be able to report a worker.
16. Clients shall be able to use search bar.
17. Clients shall be able to pay the worker through the website.
18. Clients shall be able to change their password.
19. Clients shall be able to cancel booking.

## **Workers:**

20. Workers shall be able to sign in to the website.
21. Workers shall be able to log out from the website.
22. Workers shall be able to access notification.
23. Workers shall be able to get support.
24. Workers shall be able to book a client's post.
25. Workers shall be able to report a client.
26. Workers shall be able to use search bar.
27. Workers shall be able to receive their payments through the website.
28. Workers shall be able to change their password.
- 29.

## **Administrator:**

30. Administrators shall be able to sign-in to the website.
31. Administrators shall be able to log out from the website.
32. Administrators shall be able to cancel a booking.
33. Administrators shall be able to ban worker/client users.
34. Administrators shall be able to contact worker/client users.
35. Administrators shall be able to review/accept client's pet inquiries.
36. Administrators shall be able to accept or deny a worker's registration.

# **5-List of non-functional requirements:**

## **Functionality:**

1. The site should be developed and deployed using the stack tools and servers that was approved by the Class CTO.
2. Each WWW page needs to have a functional navbar and search bar with the logo included at the top of the page.
3. Guest user should be able to see the functionality of the website in a demo before signing up.
4. Application UI shall be simple and efficient and easy to use for the user.
5. Application shall be very easy to use and intuitive
6. Application shall be hosted and deployed on Gcloud client server as specified in M0.
7. Application shall be optimized for standard desktop/laptop browsers.
8. All users should be able to contact support.
9. Client should be able to contact Admin or Worker in case of emergency.

## **Compatibility:**

10. Site should have a logo to be displayed next to the title in every browser.
11. The site shall be compatible with the latest version of Chrome browser (77.0.3865.90)
12. The site shall be compatible with the latest version of Firefox browser (68.0.1)
13. The site shall be compatible with the latest version of Safari browser (5.1.7)

## **Performance:**

14. Loading time for site shall be less than 10 seconds for any page.
15. Search bar result should be shown in less than 10 seconds.

## **Security:**

16. Registered user should be able to login with the credentials it made during sign up.
17. Client should input his/her name, address and driver's licence # to be collected as data.
18. Client's pet needs to be California licensed and the ID # needs to be inputted into the data. data.
19. Guest user should not be able to see any live data before signing up.
20. Registered user should be ab
21. Registered user's password should be encrypted and saved in the database.
22. Worker should input his name, address, and driver's licence #, and pet sitting certification to be collected as le to change their password if needed.
23. Admin should be able to ban workers and clients if needed.
24. Admin should be able to modify the bookings if requested by the client.

**Coding Standards:**

- 25. Developer should develop code to be easy to read.
- 26. Methods/functions should have one and only one purpose.
- 27. Developer should be commenting on the specific methods on his/her code.
- 28. Developer should be consistent in the naming convention of variables.
- 29. Back-End developer should develop noncomplex api.
- 30. Developer should write test cases for their functions.

**Layout:**

- 31. Site should have logo in the tab bar.
- 32. Website should be professional looking.
- 33. Website should be user-friendly.

**Data Integrity:**

- 34. User's data shall be inputted into MYSQL database.
- 35. Data in the data base should be back up every week.

**Capacity:**

- 36. The server storage shouldn't exceed 80%.
- 37. No pictures bigger than 2 Mb should be uploaded onto the server.

**Reliability:**

- 38. Inform the users if the website is going to be down for maintenance.
- 39. If the server is down, it should be restarted and back on running in less than 1 hour.
- 40. Pay functionality shall not be implemented.

# 6-Competitive Features Analysis:

| Feature                                     | Rover | TaskRabbit | The animal Nanny | CatDog |
|---|-------|------------|------------------|--------|
| Search Filter                               | +     | -          | -                | ++     |
| Specific for animal sitting (cats and dogs) | -     | -          | +                | ++     |
| security                                    | +     | +          | -                | +      |
| UI Simplicity                               | -     | +          | -                | ++     |
| Map Usage                                   | +     | +          | -                | +      |

+: feature exists

++: feature is superior

-: feature does not exist

## Summary of the competitive analysis:

Our goal for this application is being ahead of our competitors. After doing some research on different websites related to pet care many were lacking a useful search bar. Our goal is to implement an efficient search bar with minimal filters for the client and worker to be able to find his/her need as fast as possible.

Making our search bar simple and minimizing our filters, we allow the client and worker to reach and receive their desired outcome in a more efficient manner. Compared to our competitors who bombard the users with excessive filters. Along with removing unnecessary specifications (and replacing them with more general requirements /needs). And lastly our goal in our search engine is to represent the best result for the client or worker depending on their needs. As an example, clients who look for a worker in their area can use our search engine and view our best rated workers available in their area.

Related to the rest of the websites in our research we discovered the lack of pet compatibility between sitters, it seemed to be more of a match between the sitter and client rather than what would be better for their animal. Our goal is to service mainly the pet by creating an animal profile instead of having the focus on the client, but still having the client needs in mind.

With security, we will have the sitters upload their California ID information in our database. And, to have a good and reliable worker in our website we are requiring mandatory certification from the workers during the registration and sign up.

Lastly, we plan to implement a map function for the worker to show listings that are available to clients in their area.

## 7-High-level system architecture and technologies used:

- Server Host: Google Compute Engine
- Operating System: Ubuntu 16.04 Server
- Database: MySQL 8.0.17
- Web Server: Apache
- Server-Side Language: JavaScript
- Additional Technologies: Web Framework: React, Node.JS, Express
- IDE: WebStorm Jetbrains/ Visual Studio Code
- Web Analytics: Google Analytics

## 8-Team:

| Team member     | Position                                |
|-----------------|---|
| Tahar Touati    | Team lead / Github Master               |
| Amir Anjomshoaa | Back-end Lead / Database master         |
| Melissa Estrada | Front-end developer                     |
| Ivan Briseno    | Front-end Lead                          |
| Xiaopeng Rong   | Back-end developer / Database developer |

## 9-Checklist:

| Checklist Item   | Status   |
|--|----------|
| Team found a time slot to meet outside of the class  | Done     |
| Github master chosen   | Done     |
| Team decided and agreed together on using the listed SW tools and deployment server  | Done     |
| Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing | On Track |
| Team lead ensured that all team members read the final M1 and agree/ understand it before submission                                   | Done     |
| • Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)                 | Done     |

# SW Engineering CSC 648/848 Fall 2019

## CATDOG

### **Group Number 5**

Tahar Touati - ttouati@mail.sfsu.edu

Amir Anjomshoaa - Aanjomshoaa@mail.sfsu.edu

Ivan Briseno - ibriseno@mail.sfsu.edu

Melissa Estrada - mestrada7@mail.sfsu.edu

Xiaopeng Rong - xrong@mail.sfsu.edu

### **“Milestone 3”**

11/21/2019

| History Table     |                 |
|-------------------|-----------------|
| Version           | Submission date |
| Milestone 1 V 1.0 | 10/02/2019      |
| Milestone 1 V 2.0 | 10/30/2019      |
| Milestone 2 V 1.0 | 10/30/2019      |
| Milestone 2 V 2.0 | 16/11/2019      |
| Milestone 3 V 1.0 | 12/03/2019      |

## **Table of Contents**

- 1-Executive summary
- 2-Main Use Cases
- 3-List of main data items and entities
- 4-Initial list of functional requirements
- 5-List of non-functional requirements
- 6-Competitive Features Analysis
- 7-High-level system architecture and technologies used
- 8-Team
- 9-Checklist

# 1- Data Definitions V2

**Admin:** A user chosen by the leaders of CatDog team that has access to all the client and worker's contact information and booking information in case of any problems or emergencies. Admin's main job would be overseeing all the posts and bookings that is happening on the website, and in case of any misconduct or policy violation admin has the right to ban the user till further notice

**Registered User:** A Client or Worker who has signed up into our website with proper information and necessary documentations and has access to different functionality depending on his role.

- **Client:** A registered user who is looking for a worker for a pet sitting service. The Client has access to contact the available workers or wait for them to request and lastly to book a worker for the job. Client has access to the support also in case of any problems or change of plans.
- **Worker:** A registered user who provides pet sitting service. Worker needs to be a certified pet sitter to be able to accept jobs on our website. Worker has access to search for clients based on the availability and location, and also the worker has access to our support in case of any problems.

**Guest:** An unregistered user who does not have an account on the website. User can view some of the posts on the site and the functionality of the application. Guest user would also be able to read some reviews from former or current users on the website. But guest does not have access to contact or post anything till signing up on the site.

**Pet:** A client's pet cat or dog that potentially needs to be taken care of by the worker.

**Post:** A client's post seeks a pet sitting service from workers by notifying them, and it includes the date period and location for the service.

**Post a review:** Clients evaluate and score the pet sitting provided by workers after the job is done.

## Report:

- **Report a Worker:** Client writing a report to send to an admin about the worker if any misconduct or policy violation happened during the job
- **Report a Client:** Worker writing a report to send to an admin about the Client if any misconduct or policy violation happened during the booking or the job.

**Book a Client:** The worker would book a client's post, and it notifies the client for potential workers. Client can look through the available workers who booked the post and chooses one.

**Accept a Worker:** The client can choose from a list of potential workers to choose the best option available to them from the list.

**Demo:** An interactive tutorial for Guest users to be able to get an understanding of how clients and workers are using the website and the functionality of the site.

**Support:** Service to help all levels of users on the site in case of any help the user needs or if any problems occurred. Support is also available to unregistered user in case of any questions.

**Profile:** Information about the user such as the Client or the Worker or the pet.

- **Client:** Required information needed about the Client itself and some information in regards of their pet for the pet profile.
- **Worker:** Required information about the Worker itself, and a short bio to inform or explain themselves to the Clients. And also any necessary information for the clients to read such as job experiences.
- **Pet:** Information about the pet provided by the client in regards of the personality of the dog. Such as likes and dislikes to make the pet sitting for the worker easier.

**Pet inquiries:** This is mainly used by the worker as help book in case of any problems. Required information for setting up the pet profile for the clients. Client needs to answer some question in regards of the pet's health insurance, diet restriction, pet interactions with other dogs and any other information that the worker needs in case of any problem with the pet.

**Certification:** Workers needs to have valid Pet Sitting Certification which can be acquired by any valid institution in order to sign up and work on our site.

## 2- Functional Requirements V2

### Priority 1:

#### **Guest:**

1. Guests shall be able to sign up and create an account on the website:
  - 1.1. Guests shall be able to sign up as a client.
  - 1.2. Guests shall be able to sign up as a worker.
    2. Guests shall be able to get support from admin.
    3. Guests shall be able to access a demo tour of the website.
    4. Guests shall be able to skip the demo.
    5. Guests shall be able to use search bar:
      - 6.1. Filter workers reviews by location.
      - 6.2. Filter workers by price range.
      - 6.3. Filter workers by pet preference (cat/dog).
      - 6.4 Filter workers by numbers of star reviews.

#### **Client:**

6. Clients shall be able to sign into the website.
7. Clients shall be able to log out from the website.
8. Clients shall be able to get support from admin.
9. Clients shall be able to create posts.
  - 10.1 Client should include the date period for the post.
10. Clients shall be able to book a worker.
11. Clients shall be able to create Pet profiles.
  - 12.1 Client should answer the pet inquiry questions.
12. Clients shall be able to delete Pet profile.
13. Clients shall be able to use search bar:
  - 15.1. Filter workers by booking number.
  - 15.2. Filter workers by username.
14. Clients shall be able to change their password.
15. Clients shall be able to cancel booking.

#### **Workers:**

16. Workers shall be able to sign into the website.
17. Workers shall be able to log out from the website.
18. Workers shall be able to get support.
19. Workers shall be able to book a client's post.
20. Workers shall be able to report a client.
21. Workers shall be able to use search bar:
  - 21.1. Filter clients' posts by location.
22. Workers shall be able to change their password.

**Administrator:**

23. Administrators shall be able to sign in to the website.
24. Administrators shall be able to log out from the website.
25. Administrators shall be able to cancel a booking.

**Priority 2:**

**Guest:**

26. Guest can inquire about a sitter to customer support
27. Guest can report a sitter to customer support

**Client:**

28. Clients shall be able to review a worker.
29. Clients shall be able to report a worker.
30. Clients shall be able to pay the worker through the website.
31. Clients shall be able to sort their searching results.
32. Clients shall be able to filter workers by payment preference.
33. Clients
34. Client shall be able to receive picture updates from sitter
35. Client shall be able to have text notifications from sitter

**Workers:**

36. Workers shall be able to receive their payments through the website.
37. Workers shall be able to sort their searching results.
38. Workers shall be able to access notification.
39. Worker shall be able to access a notification system to communicate with Clients.
40. Worker shall be able to narrow down their preferences by animal size and breed.
41. Workers shall have a system that guarantees CatDog find a new sitter in case of emergency

**Administrator:**

42. Administrator shall be able to reimburse Clients
43. Administrator shall be able to demonetize workers (lower rates)
44. Administrator shall be able to generate earnings estimate for the worker
45. Administrator shall be able to view direct messages between clients and workers
46. Administrators shall be able to ban worker/client users.
47. Administrators shall be able to contact worker/client users.
48. Administrators shall be able to review/accept client's pet inquiries.
49. Administrators shall be able to accept or deny a worker's registration.

## Priority 3:

### Client:

- 50. Clients shall be able to have service history.
- 51. Client shall be able to view previous sitters
- 52. Client shall be able to be notified of new reviews from past sitters
- 53. Client shall be able to check in on pet via video chat
- 54. Client shall be able to receive end of day summary from Worker via CatDog inbox
- 55. Workers shall be able to have service history.

### Workers:

- 56. Worker shall be able to view rates of other sitters
- 57. Worker shall be able to raise rates
- 58. Worker shall be able to have live video chat with Client
- 59. Worker shall be able to send end of day summary to Client via CatDog inbox

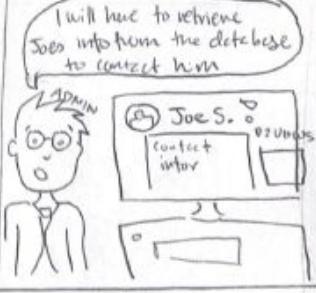
# 3- UI Mockups and Storyboards

## Admin - Privileges/Technical Support/Customer Support:

• Michael receives a notification marked as urgent. Michael is the Admin, logs into internal email

• The message is from Julio (client) reporting Joe (sitter) for a no show.

• If Joe (sitter) does this again Michael (Admin) will enforce a BAN and remove profile



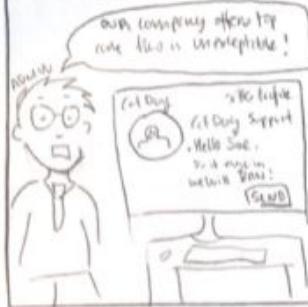
↳ We will flag Joe now we have a bad review

↳ Contact preference for sitter is email.

• Michael (Admin) contacts Joe (sitter) to inform him and send a warning

• Michael (Admin) now has to contact Julio (client) to fix the situation and give him options

• Michael offers to give Julio the option to cancel w/t full refund or rebook someone else.



↳ fix the issue with

client is important  
Julio's preferred  
method of contact  
is email.

↳ support offered

↳ refund  
↳ cancellation  
↳ rebook

But we were able to respond fast we didn't lose a client or received an offical bad review.

## Client (Creating Pet Profile):

### Client (Creating Pet Profile)



- After deciding to check out the website to find the perfect care for his pet.

**[LANDING PAGE]**

Welcome To CatDog!  
Find The Care Your Pet Deserves.

- How it works
  - Our Services
  - Reviews
  - Picture
  - Services
- GET STARTED!
  - Reviews
  - Testimonials
- BROWZ A SITTERS
  - Benefits
  - Blu

Logo / Pet Pic

continue as guest

**[DEMO PAGE]**

Service for every Dog and Cat

- Dog and cat Boarding
- House sitting
- Dog care
- Drop in Visits

some w/ pet sitters  
✓ you control  
✓ background checks  
✓ certified pet sitters  
✓ approved by our team

All services are backed by CatDog 24/7 support  
Reservaton protection

Browse our Best sitters

#### [LANDING PAGE]

- Julio decides to go through the Demo
- Where we explain how our service works
- How we bridge pet lovers to their pet
- The security we offer.
- He is able to view our best reviews
- He is able to browse our best sitters.

#### [DEMO PAGE]

- After viewing the services & reviews he continues to Browse the sitters profile.

**Sitter Profiles**

- (1) Cindy T. Bio: reviews
- (2) Tom S. Bio: reviews
- (3) Sam C. Bio: reviews

trusted sitters

Babysit relax 24/7 support

**Sign Up**

Logo  
First name  
Last name  
Email  
Phone  
Pet pic  
Sign up

Learn More About-fatdog Top Sitters

**User Page [Private]**

Julia's Profile

Julia's Bio Edit Contact

Inbox Requests Bookings History

your Pet's BioProfile

Pet Bio Edit profile

Welcome! [fatdog.com](http://fatdog.com)

Search Sitters Customer Support Help Create Request / Post

Finally he sees the reviews and decides to sign up.

- He will have to create a client account

\* After filling out contact info he is directed to his user page

[USER PAGE] [Private]

where he can create the pet profile.

**PET**

CHOOSE CAT OR DOG ICON

PET PERSONALITY Q&A [Pet Started]

**Needs Services** [Edit] [Save]

Behavior Medical

**Pet Bio** [Edit] [Save]

**PETS** STARKLES

**PET BIO**

**PET PERSONALITY Q&A**

Julia's Profile Edit Contact

Needs Services Behavior Medical

Julia's Bio Edit

Julia's Pet Profile

Julia's Bio Edit Contact

Julia's Pet Profile

- Now Julio can create his pet profile and add his pet's w/bio and needs.
  - also answering some basic pet questions.
- [PET PROFILE] is public after completion.

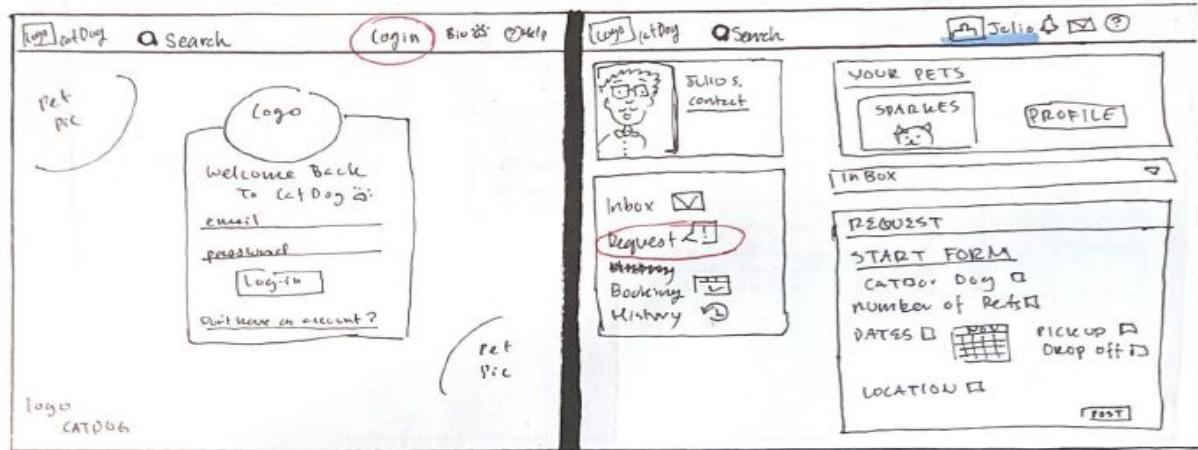
- Julio is finally done.
- fatdog has approved his pet profile and it is now visible to workers.
- Julio now will be able to sign in & out of catalog reviews and accept pet inquiries and requests.

The final demonstration.

[Public Profile]

## Client (Booking Worker for Specific Days):

### Client (Booking Worker for Specific Days)



- Julio (client) has a trip coming up that will be 3 days. He decides to log in and create a book request.
- [Log-in Page]

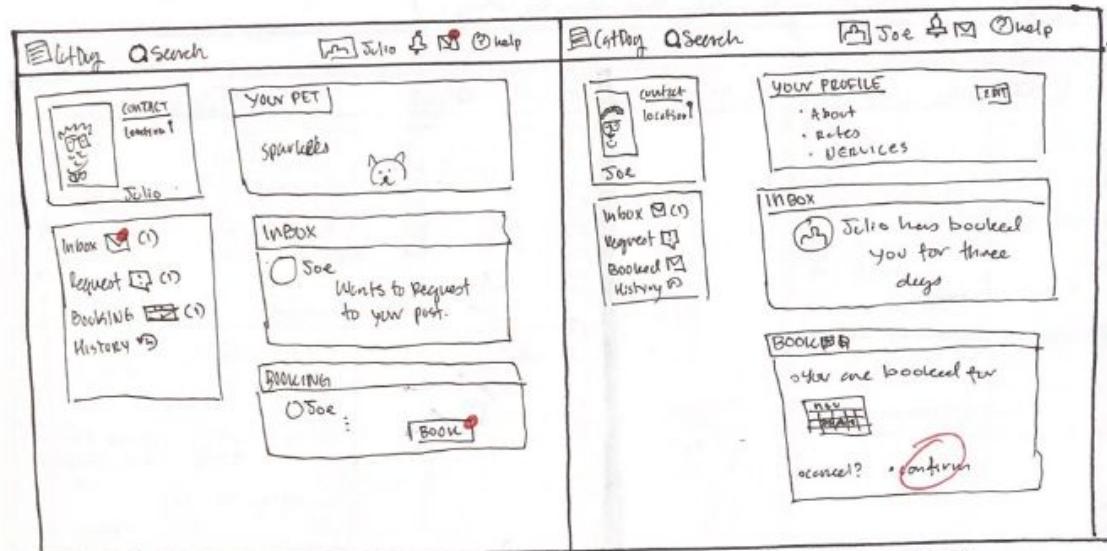
- Julio (client) is logged in, now he can continue to work towards creating a booking request.
- USER PROFILE PRIVATE
- fills out Request Form /completes Request ~~Booking~~
- Posts Request Form



- His post is now available for workers/sitters to view
  - with the pet profile attached.
- [POST PAGE] Public

- Joe (worker/sitter) decides he has free time and decides to look at the post. He decides that he is available to help Julio out and request to book [JOE PROFILE]

- Julio receives a notification that someone (Joe) has requested to be considered.
- Julio goes on his profile to view reviews.

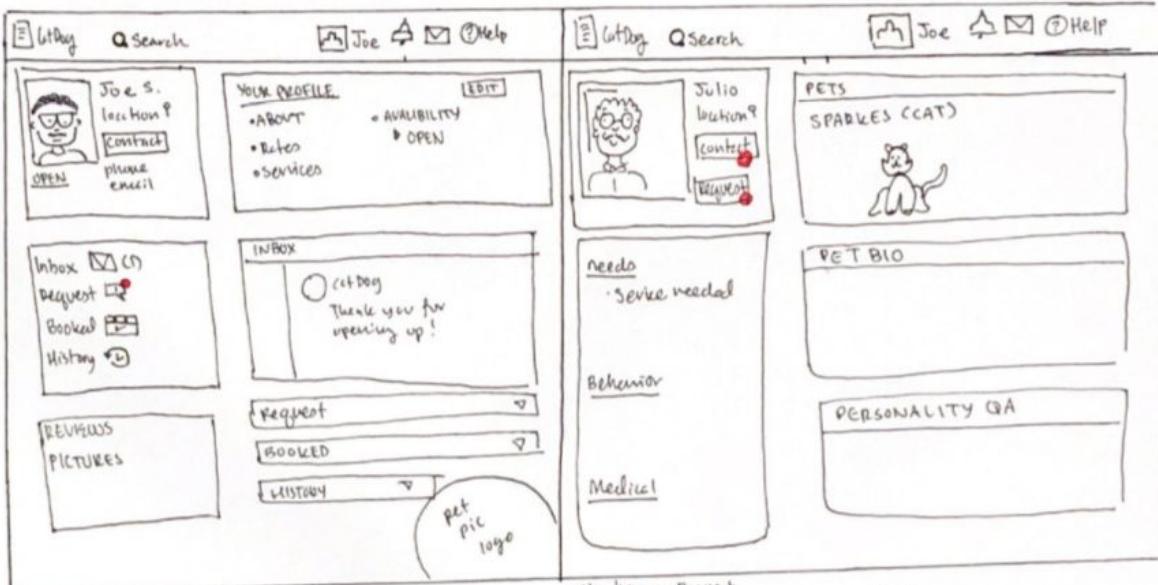


• Julio Books Joe.

• Joe confirms the booking on his worker/sitter profile.

## Worker (Request Client Booking):

### Worker (Request Client Booking)



- Workers/sitter profile
- Joe changes his availability to OPEN now he can receive notifications.
- He has a request
- So he decides to view the Pet Profile
- ADD decides to request the booking



- Joe has been booked and can now approve / confirm
- Julio receives a message that Joe has confirmed.

## Guest (Register/Sign Up as Worker):

Guest (Registered/Sign Up)

Connie decides to check out CatDog and see if she would sign up.

Connie decides to first browse the website as a guest. She reads about the company and the benefits of CatDog.

Now we will view the website in Connie's perspective.  
She decides to sign up.

Get Paid to Play with furry animals  
CatDog makes it easy by creating you animal lover spirit to the best pet parents.

make your own money & schedule

Become a Sitter

Flexibility puts you in control  
✓ set your own schedule  
✓ Create your own offers  
✓ Set pet preference that works for you

The tools you need to succeed  
✓ Your services are protected  
✓ Manage your own business  
✓ 24/7 support

How it works

① Create your profile → ② Verifying → ③ Accept request method

Softkey first Always!  
We work to ensure kids keep smiling

Start creating your profile

\* Connie sees the benefits  
\* Connie sees the reviews.

|  |  |   |
|--|--|---|
| <b>Log In</b>   CutDog   Search   Login Bio &  | <b>Log In</b>   CutDog   Search   Login Bio &  | <b>Log In</b>   CutDog   Search   Login Bio & <b>RATES</b>  |
| Account Info   Service & Roles   Silver Profile   Calendar<br>Tell us about yourself?<br>Thank you for your interest in CutDog!<br>Why are you becoming a Sitter?<br><input type="button" value="Select Answer"/><br>Do you have Pets?<br><input type="button" value="Select Answer"/><br>Let's start with the Basics<br>Add address<br><input type="text" value="Address"/><br>↓<br><input type="text" value="Email"/><br><b>Step 2</b> | Require a Photo for Profile<br><input type="button" value="Upload"/><br><br>Birthday _____<br>Phone number _____<br>Emergency contact name _____<br>number _____<br>Member of Horse Hotel _____<br>What services would you like to offer?<br>↓<br><input type="button" value="SAVE"/><br><b>Step 3</b> | <br>CutDog Walkers <b>\$5</b><br><br>CutDog Boarding <b>\$35</b><br><br>CutDog Horse Sitting <b>\$35</b><br><br>CutDog Visits <b>\$25</b><br><br>CutDog Daycare <b>\$15</b><br>need Help?<br><input type="button" value="SAVE"/><br><b>Step 3</b> |

Lorraine begins the registration process

Lorraine can specify what she would like to be offered for services.

|   |   |  |
|---|---|--|
| <b>Log In</b>   CutDog   Search   Login Bio &   | <b>Log In</b>   CutDog   Search   Login Bio &   | <b>Log In</b>   CutDog   Search   Login Bio & <b>(Logo)</b>  |
| Do you have Preferences?<br><input checked="" type="checkbox"/> Cats <input checked="" type="checkbox"/> Dogs<br>Create "About Me"<br><input type="text"/><br>↓<br>Let's Get Verified!<br>In order to continue on with creating a profile you must upload a verification certificate<br><input type="button" value="Upload"/> <input type="button" value="CONTINUE"/> | Background Check<br>license # _____<br><input type="button" value="SUBMIT"/><br>YAY! Almost there<br>TERMS & conditions <b>OR</b><br>:<br>Rules & Regulations <b>OR</b><br>E-Signature <u>Lorraine Sargent</u><br><input type="button" value="DONE"/> | <br><b>Thank You</b><br>Please check your email for approval and verification to continue.<br>What's Next?<br>After being approved you'll be able to come back and sign in to info to your profile that was generated from your response.<br>See You Soon<br><b>(Logo)</b> |

- Lorraine checks her preferences
- creates an about me
- gets a message she needs to upload a certificate or verification.
- proof of pet lover / pics or testimony

- Does Background check
- Agreed to rules
- E-signature
- Message sent to Admin

- Lorraine is all done
- now she waits for a confirmation email
- After that all her images will be on CutDog.

# 4- High level database architecture and organization

## Business Rules

Admin:

1. Admin can have access to Client full information.
2. Admin can have access to Worker full information.

Registered User:

3. Client can have multiple pet profiles.
4. Client can have only one post at a time.
5. Client can accept only one worker for a job.
6. Client can have only one profile.
7. Worker can request multiple bookings.
8. Worker can be accepted for only one booking.
9. Worker can have only one profile

Important/Weak Data:

10. There could be multiple post available from different clients.
11. There can multiple reports from each client and worker. (support)
12. There can only be one pet inquiries per pet.
13. There can be only one certification per worker.

## Entities

**Client:**

Attributes:

1. ClientUser **PK**
2. Name: (FirstName + LastName)
3. Address: (Street + State + ZipCode)
4. Email
5. petId **FK**
6. PetQuantity

**Worker:**

Attributes:

1. WorkerUser **PK**
2. Name: (FirstName + LastName)
3. Address: (Street + State + ZipCode)
4. Email
3. Certification

**Pet:**

Attributes:

1. petId **PK**
2. clientUser **FK**
3. Name:

4. petInquiries
5. bio

**Post:**

Attributes:

1. postId **PK**
2. clientUser **FK**
3. datePosted
4. dateRequested
5. lengthRequested

**Booking:**

Attributes:

1. bookingId **PK**
2. workerUser **FK**
3. clientUser **FK**
4. petId **FK**
5. postId **FK**
6. dateBooked

**ClientProfile:**

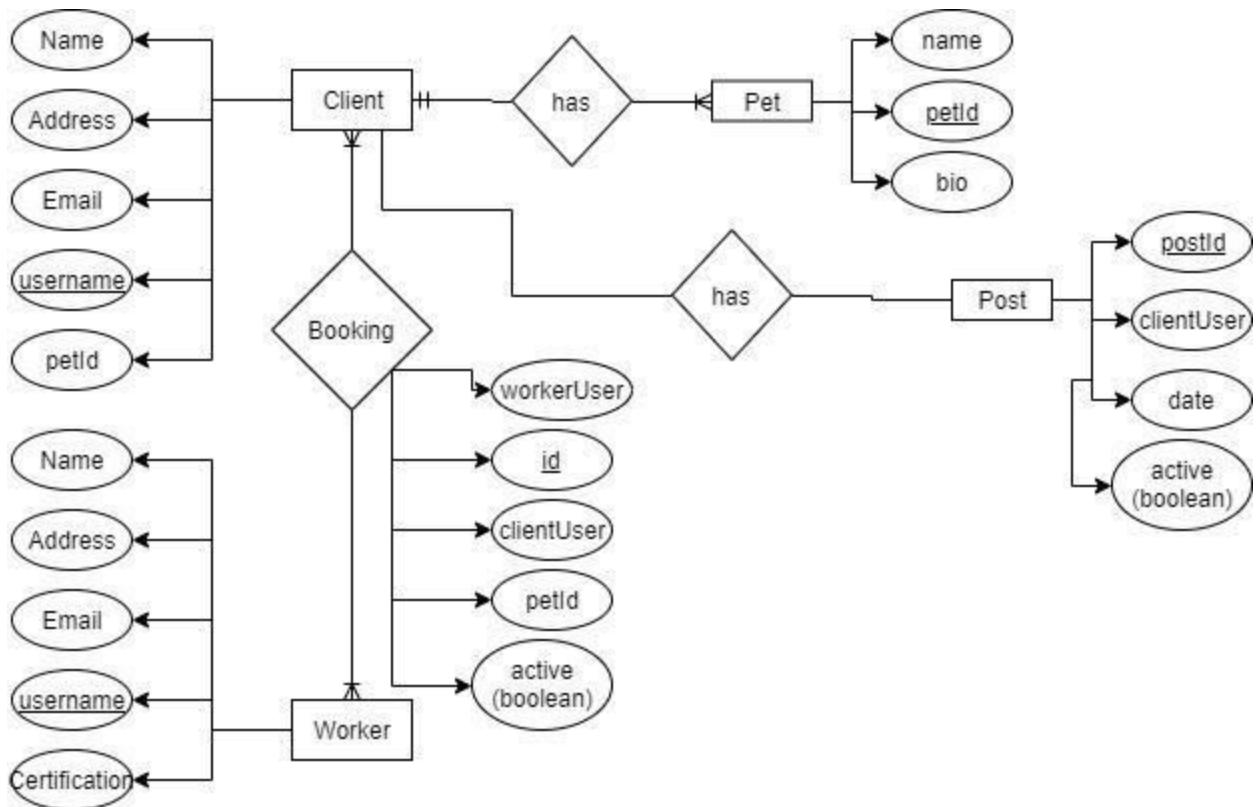
Attributes:

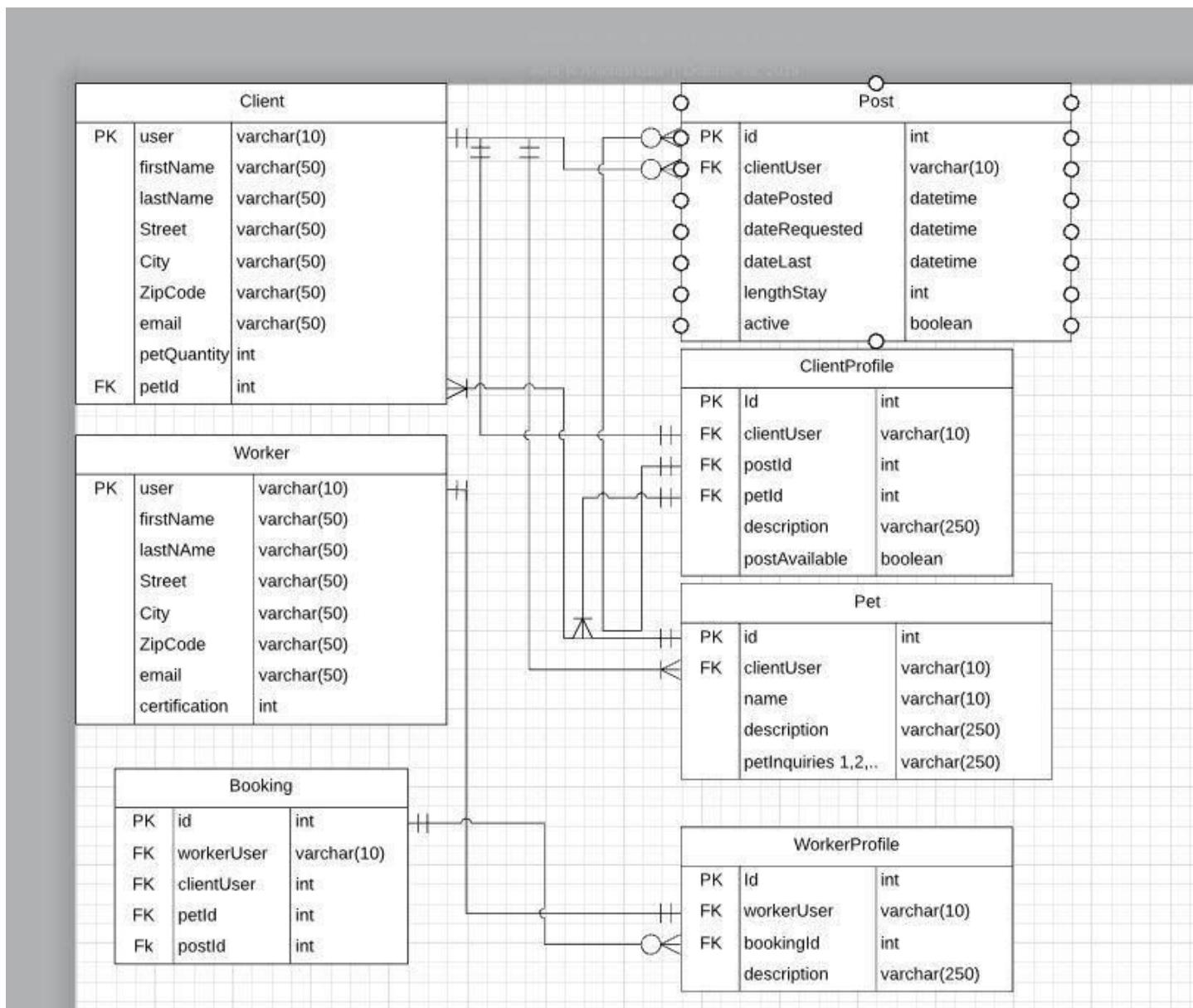
1. clientUser **PK**
2. postId
3. postAvailable
4. bio

**WorkerProfile:**

Attributes:

1. workerUser **PK**
2. workHistory [booking] **FK**
3. bio





**Media Storage:** Saving users pictures are a functional priority 2. If we get a chance we would be implementing it, and we would be saving the user's uploaded picture and saving it on the server as static file and just save the path directory inside the table for specified user.

**Search Filter:** Search in our website is working in two ways. One is the client looking for worker by username or booking number and this would allow back end to grab the data from the worker table or booking table depending on the username or booking number. (We are thinking of implementing also search by name for clients to look for worker). Next would be the worker to search and he is only allowed to search by zipcode or name of the city he/she is looking for job to bring up all the results depending on that city from the post table.

# 5- High Level APIs and Main Algorithms

## 5.1 APIs:

Below are some general application programming interface (API) that will be implemented by back-end engineers. Some general features can be explained as follows:

- The “xxx” ahead of each API notifies the corresponding database model, and therefore will be replaced by real model name (e.g., client, worker, etc.) during the development process.

- Each API is completely in lower-case format, without any white spaces in between.

- “/” separates the name of API and the required parameter.

- “id” is always numeric value, to reduce the time complexity of searching algorithm within database.

The back-end team is developing the end points for the front-end to communicate with the client and send the script over to back-end. All the api's would be path toward /api/

### high-level explanation for the APIs:

#### 1) /api/search

- This post request to the api, has a body from the front-end which sends over the user input for the search and also lets the back end know if they are looking for clients post or workers availability.

#### 2) /api/createClient

- This post request to the back end sends over a body from the front end containing all the information for signing up a client to our website.

#### 3) /api/clientProfile

- This post request is when the client is filling out the information for their profile, also contains information about the client.

#### 4) /api/createWorker

- This post request to the back end sends over a body from the front end containing all the information for signing up a worker to our website.

#### 5) /api/workerProfile

- This post request is when the worker is filling out the information about themselves for their profile. And contains any experiences and certification.

#### 6) /profile?user=xxx

- This get request sends the username of the client or worker and depending on the role back end would generate all the necessary data the front end needs to show their profile. (Both client and worker)

7) /api/deletePost 8) /api/cancelBooking

- Depending on the client or worker, they can cancel a post or booking and these post request would let the back end know to remove the data from the live site and just recorded them.

Search Algorithm and our goal for it:

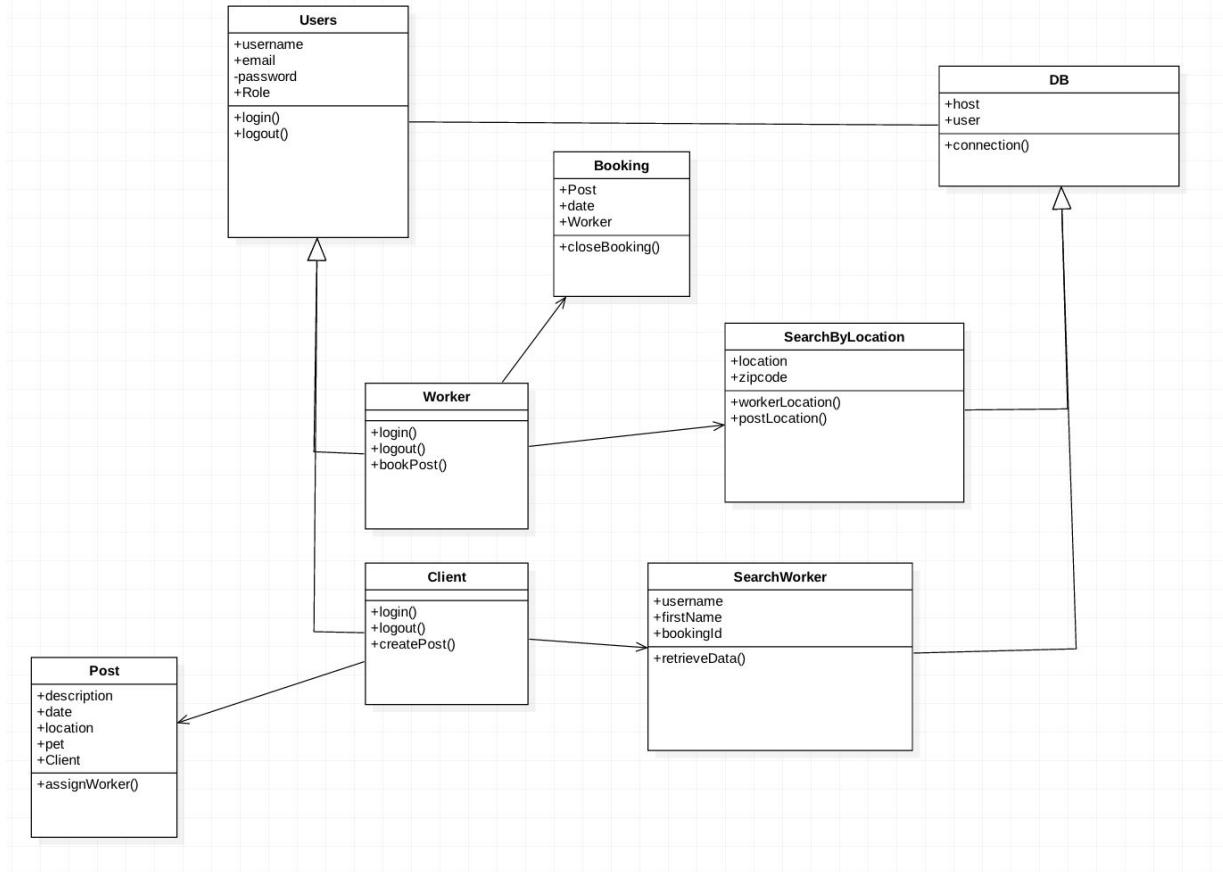
1) Simple Search from the worker point of view:

- The worker has a search bar available to them to be able to search for available post by clients near to them. The user can search by location or zip code and depending on the available posts inside the data base table posts I could check the zip code or location and show the worker the list of post available to book.

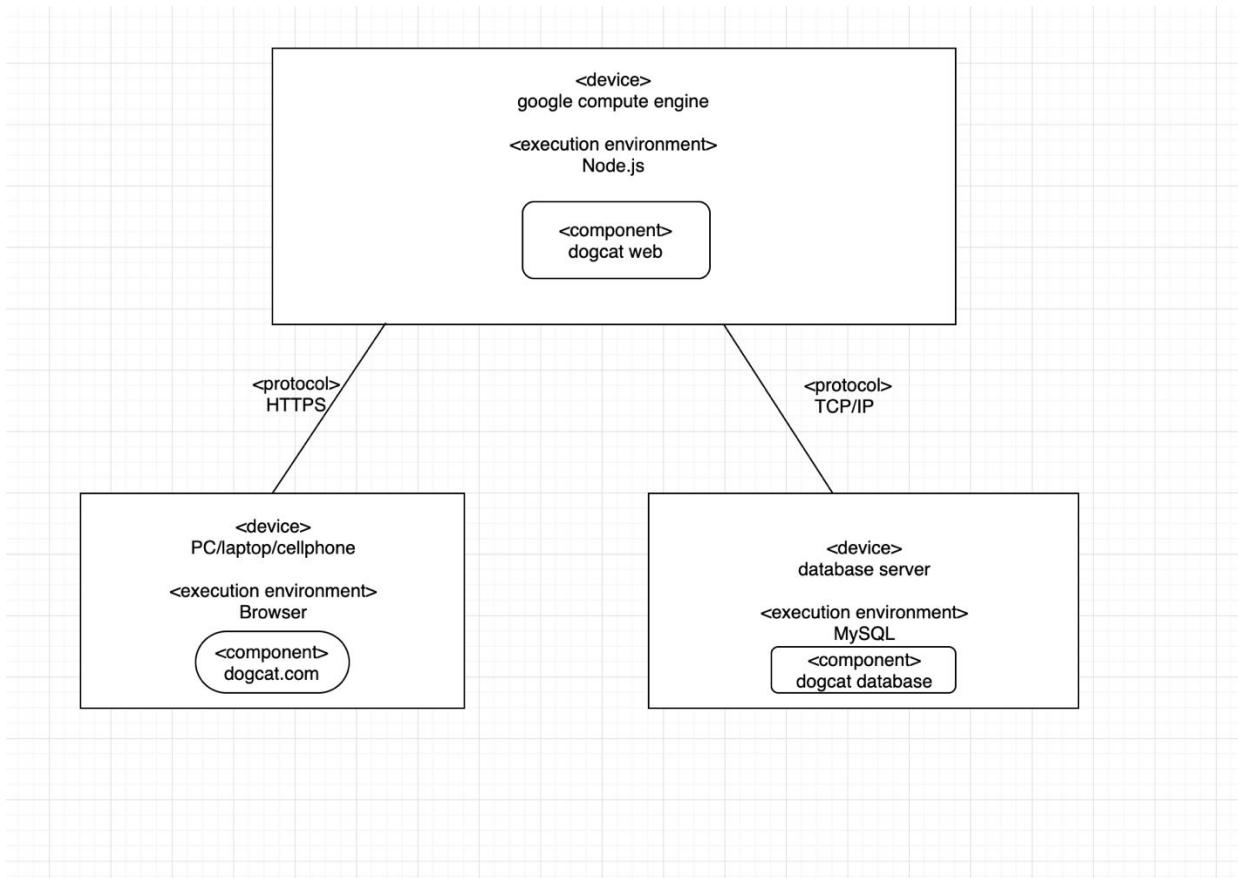
2) Simple Search from the client point of view:

- The client can search in the search bar for workers by searching their name or username or booking id. The result would be back end checking if the user input a name, it would grab all the available names matching to the search and presenting them. If user searches by username it would be easier and faster response because it can be only one result inside the database containing the username cause it's the primary key. And if the user searches for a booking id, back end would just check into the booking table and find the worker from that table and grab all the necessary information and present them.

# 6- High Level UML Diagrams

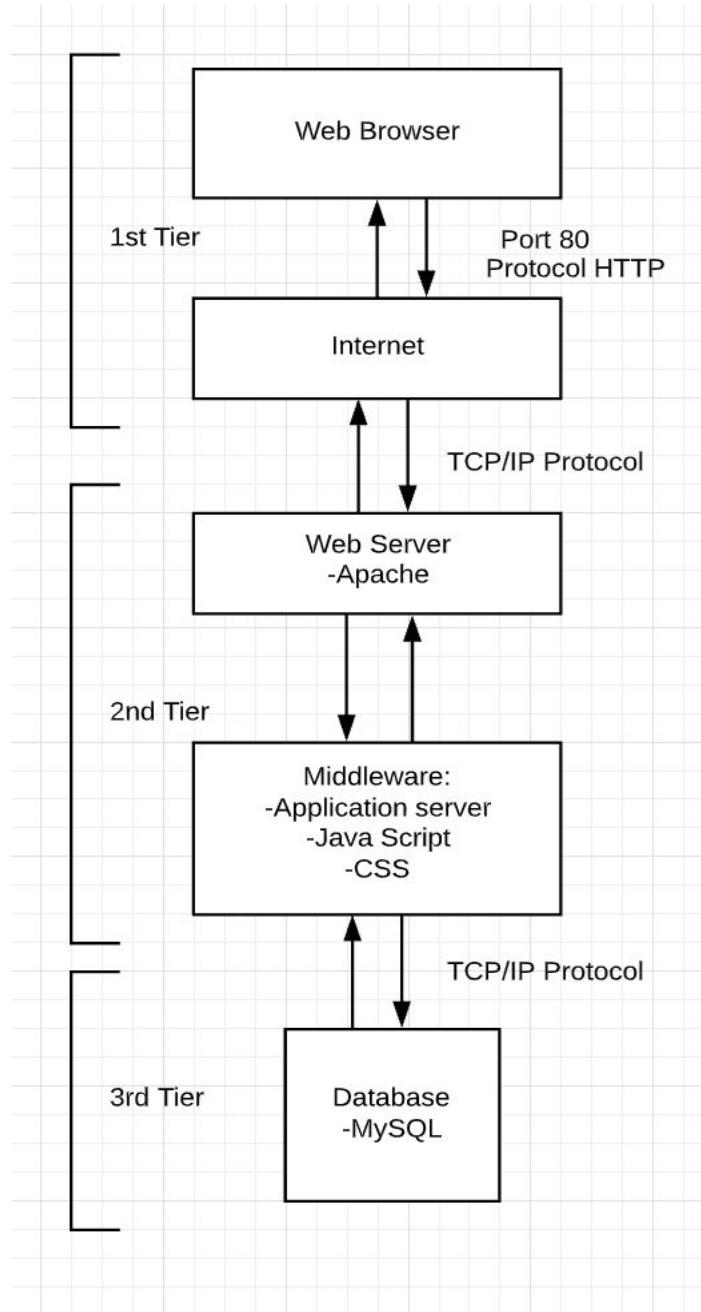


-Class Diagram



Deployment Diagram

# 7- High Level Application Network and Deployment Diagrams



-Application network Diagram

## 8- key risks

Currently for our project there are a few risks we have encountered, and we have also analyzed future concerns we may come across during our development. For our skills risk our team has been working hard on developing a strong foundation in understanding the core of our application because we all did start at a more intermediate level. We have pushed ourselves to gain the proper skills by watching tutorials and helping each other. The risk we run is not everyone being at the same level technically and it may lead to confusion or problems when merging. To be able to cope with our different skill levels we have created a system of conference calls that each one of us can take advantage of and ask for help or clarification on any topic discussed so far and on the future teamwork.

As for schedule risks we have really committed ourselves to certain days and times dedicated to the sole focus of our project. The risk we run with our schedules for the future, is related to other schoolwork and test that may interfere with the times we have selected. In order to stay on top of our schedules we have created a time sheet that we keep updated that way as a team we are able to

Our technical risks are communicating properly with our database because of the large amount of data we require from the user. Another issue we may run into is our backend connection to the sever we may run into issues and need to work and ask for help from our backend leader to help the front end create a more streamline connection. This is currently being resolved by having the front end and backend working closely together.

For teamwork risks we did have some issues in the beginning, but we have resolved them and are working great together. With the use of Trello we have been able to keep all our tasks organized and maintained this has allowed us all to be held accountable as well as on track. We will continue to take advantage of Trello, hangout, and slack to keep our team on track to deliver.

Legal content risks include giving the correct copy right to ownership for any software we use. We are avoiding this by making sure to give credit where it is due. As a team we have already come a long way and are excited to continue working alongside to work and overcome any obstacle as a team.

## **9- Project management:**

This milestone was managed in a much more efficient and collaborative manner than our previous one. We achieved a harmonies distribution of task by having our team use Trello to help organize our project. This tool allowed us to set up tasks with deadlines that we all were able to agree on. Another tool we've been utilizing the most is Google's Hangout. During these calls we update the team on where each individual is in regard to their progress and offering help in any aspect of production. The school library has become our headquarters where we conduct weekly in person meetings. During these in person meeting the team sets the groundwork and brainstorms on the ideas. We also work out problems that have been encountered by working together to generate a solution. By having these meetings, we can clarify any confusion and bring our ideas together and create a game plan for the week. Also, our main source of communication is done on a regular basis using our Slack channel where our team communicates daily. For the future we will continue to use all these tools to tackle task and conduct our communication. These tools have allowed our team to stay on track and have a streamline process thus far.

# 10- Project Feedbacks:

Here is a list of the feedbacks we received for our frontend and UI:

- Fixe the empty space in the home page.
- Make the search bar more noticeable.
- Make each parallel boxes (reviews etc) on the clientApp page the same size.
- Make the website reviews bigger / more emphasized.
- Make the names of client reviews of the sitters bigger.
- Make the background color less attention taking.
- The client subscription form should be smaller.
- The Pet profile bouton shouldn't be on the top of the client profile.
- Make the procedure to create a post as a client more obvious.
- Everything that explains how the website works needs to be bigger.
- Emphasize what the user should focus on when he comes to the website, make important things more obvious.

# SW Engineering CSC 648/848 Fall 2019

## CATDOG

### **Team Number 5**

Tahar Touati - ttouati@mail.sfsu.edu

Amir Anjomshoaa - Aanjomshoaa@mail.sfsu.edu

Ivan Briseno - ibriseno@mail.sfsu.edu

Melissa Estrada - mestrada7@mail.sfsu.edu

Xiaopeng Rong - xrong@mail.sfsu.edu

### **“Milestone 4”**

12/19/2019

| History Table     |                 |
|-------------------|-----------------|
| Version           | Submission date |
| Milestone 1 V 1.0 | 10/02/2019      |
| Milestone 1 V 2.0 | 10/30/2019      |
| Milestone 2 V 1.0 | 10/30/2019      |
| Milestone 2 V 2.0 | 11/18/2019      |
| Milestone 3 V 1.0 | 12/03/2019      |
| Milestone 4 V 1.0 | 12/5/2019       |
| Milestone 4 V 2.0 | 12/19/2019      |

# **Table of Contents**

- 1- Product Summary. 3
- 2- Usability Test Plan. 5
- 3- QA Test Plan. 7
- 4- Code Review.. 9
- 5- Self-Check: Adherence to Original Non-Functional Specs. 15

# Product Summary

**Product Name:** CatDog

**Product Url:** <http://34.67.160.125/>

## List of Priority 1 functional requirements:

1. Guests shall be able to sign up and create an account on the website:
  - 1.1. Create account as worker.
  - 1.2. Create account as Client.
2. Guests shall be able to send message to support team admin.
3. Guests shall be able to access a demo tour of the website.
4. Guests shall be able to skip the demo.
5. User shall be able to sign into the website.
6. User shall be able to get support from admin.
7. Clients shall be able to create Pet profiles.
8. Clients shall be able to create posts.
9. Client should include the date period for the post
10. Clients shall be able to create Pet profile in sign up.

### All the registered users:

11. User shall be able to log out from the website.
12. User shall be able to get support from admin.
13. User shall be able to change his/her password.
14. Clients shall be able to see who booked the post.
15. Clients shall be able to use search bar:

Filter workers by booking number.

Filter workers by username.

16. Workers shall be able to accept to book a client's post.
17. Workers shall be able to use search bar to search for available jobs in location
18. Worker shall be able to see the available post to book.
19. Administrators shall be able to sign in to the website.
20. Administrators shall be able to log out from the website.
20. Administrators shall be able to delete/ban user.

**Product Summary:**

CatDog is a website that provides pet sitting by accredited sitters. Our website is targeted to two types of users, the first type are pet owners who frequently travel and need a service that provides pet care during their travel duration, and the second type are accredited sitters and pet-friendly people wanting to earn extra money while taking care of a pet. This is the main focus of our market for catdog.

Users with a client account will be able to create pet profiles for their pets, create a post that seeks a pet sitting service from workers by notifying them and includes the date period and location for the service, select a sitter from those who booked his post and finally confirm the booking.

Users with a sitter account will be able to search clients by location, book a client's post from the available posts or from the post feed, get notified when a client accepts them.

This is the URL of our website: <http://34.67.160.125/>

# Usability Test Plan

## Test Objectives:

The feature being tested is for the client side of our website. The feature is going to let the client to be able to create a post. The feature will be tested with making sure of the effectiveness and accuracy due to being one of the main functionalities of our website. The test is to make sure the Client side of our website is able to create a post for their pet. This process contains updating the database so the Worker side of our website is able to see the post created by the client moments after.

## Test Description:

### · System Setup:

The user will be provided a laptop with windows 10 OS running on it, and also with a working internet connection and a chrome installed on it with the version 78.0.39 (latest version for windows).

We also going to guide the user through the sign-up process as a client so the user is able to test the functionality on their own profile.

### · Starting Point:

After making sure user has a working account we could start the test by the login page which would be [/login](#) which would redirect the user after login into the page [/clientprof?user=\[\]](#).

### · Intended Users:

For this test we are looking for people who never used our website, and also are cat or dog pet owners who are adapted to the lifestyle that would cause them to leave the city often a lot and can't take the responsibility of taking their pet with them on the trips. This group of people is our main focus of marketing team.

### · Url to be tested and measured:

The first Url user would see is the [/clientprof?user=\[\]](#) which is query string `user=[username]` and this would allow the system to make sure with the cookie and authenticate the user to be on that page. And every api call is communicating with the back-end through the request body. And user satisfaction would be measured at last by the Likert scale questionnaire.

## Usability Task Description:

- User first needs to sign up with the help of us as a client in our website.
- After the user creates the account we would redirect them to login page and from their they should be able to navigate forward
- After the user logs into the website it would be redirected to the profile page and from their the tab for creating a post is visible to them and easy to navigate.
- And lastly the form is easy to use to fill up and choose for what pet they are choosing and what range of date.

**Questionnaire:**

- The post creating feature was easy to find and navigate to (Circle one):  
Strongly Disagree      Disagree      Neutral      Agree      Strongly Agree
  
- The form for creating the post was simple and easy to fill out (Circle one):  
Strongly Disagree      Disagree      Neutral      Agree      Strongly Agree
  
- Choosing a date and time for the post was straight forward (Circle one):  
Strongly Disagree      Disagree      Neutral      Agree      Strongly Agree
  
- I am satisfied with the post creation feature (Circle one):  
Strongly Disagree      Disagree      Neutral      Agree      Strongly Agree

# QA Test Plan

## Test Objectives:

The objective of the test is to explore the possibilities offered by the post feature, try all the error cases and make sure that the posts gets published in the workers posts page when posted, and also in the active posts when they are booked by a worker.

## Hardware and Software setup:

Hardware Setup: a laptop connected to the internet connection and running windows 10.

Software Setup: google chrome installed and open the website's signup page which URL is <http://34.67.160.125/Signup>

## Features to be tested:

Client creating a post and publishing the post for the workers to request to book.

## QA Test Table:

### For Chrome and Firefox:

| # | Test                    | Test Description   | Input   | Expected output   | PASS/FAIL |
|---|-------------------------|--|---|---|-----------|
| 1 | The Calendar Date Range | Client X creates a post with a starting date of yesterday. | Start date: 12/4/2019<br>End date: 12/09/2019 | Display an error with a message “please check the dates of your post” | FAIL      |

|   |  |  |   |  |             |
|---|--|--|---|--|-------------|
| 2 | Post visibility in the client page as a the post gets created inside the tab pending post. | Client X creates a valid post with correct pet name and date range and time for pickUp | petName: [userInput]<br>Date Start: [userInput]<br>Date End: [userInput]<br>Location: [userInput] | The post appears inside the clients page in the pending post tab.  | <u>PASS</u> |
| 3 | Post being request to be booked by worker  | Worker Y requests to book the post that was created by the client X                    | Clicking on the booking button accept on the worker profile page                                  | Client receives an email that his post was requested to booked and the posts shows up in the request post page on the client profile | <u>FAIL</u> |

# Code Review

Other team members reviewed my code for my routes which contains my api routes and post routes:

I received comments in my code with his name next to it:

```
var express = require('express');

const router = express()

const session = require('express-session')
const search = require('./data/db').search

const client = require('./data/clientDb')

const addSub = require('./data/db').addSub

const dbConnect = require('./data/db').connect
const dbDisconnect = require('./data/db').disconnect

const bcrypt = require('bcryptjs')

const uuid = require('uuid/v4')

//Jose Castanon -- more descriptions as to what each route does would be nice.

dbConnect()

const auth = (req,res,next) => {
  console.log(req.session)
  if(req.session.loggedIn){
    next()
  }
  //res.status(403).send("Access not allowed")
}

router.get('/', function(req, res, next) {

  req.session.loggedin = true
  res.render('index', { title: uniqId });
});

router.get('/test', auth ,(req,res) => {
  console.log(req.session.loggedin)
  res.send('Hi')
})

router.post('/db/search', (req,res) => {
  console.log(req.body.searchValue)
```

```

search(req.body.searchValue).then((result)=>{
    res.json(result)
}).catch((e) => {
    console.log(e)
})
})

router.post('/db/addSub', (req,res) => {
    console.log(req.body)
    addSub(req.body.firstName,req.body.lastName,req.body.email).then((result) => {
        res.send(result)
    }).catch((e) => {
        res.send("Already added")
    })
})

router.post('/db/addClient', async (req,res) => {

    console.log(req.body)

    try{
        //Jose Castanon -- request body can be passed to add client, that way you can have less variables and cleaner code.
        const user = req.body.userName;
        const firstName = req.body.firstName
        const lastName = req.body.lastName
        const street = req.body.street
        const city = req.body.city
        const zipCode = req.body.zipCode
        const email = req.body.email
        const petQuantity = 0
        const petId = 0

        const password = req.body.password

        const hashPassword = await bcrypt.hash(password, 3)

        console.log(hashPassword)

        client.addClient(user,firstName,lastName,street,city,zipCode,email,hashPassword).then((result)=>{

            req.session.loggedin = true;
            console.log(req.session)

            res.status(201).send("Created")
        }).catch((e) => {
            res.send(e)
        })
    }catch(error){

        console.log(error)
    }
})

```

```

})

router.post('/db/addPet', auth ,async(req,res) => {
  try{
    const result = await client.addPet(req.body.user,req.body.name,req.body.description)
    res.status(201).send('All good')
  }catch(error){
    res.status(200).send('User Taken')
  //Jose Castanon -- errors could be more descriptive. Error says user taken on an add pet route. Does this mean that the
  //add pet route also checks if the user has been registered already?
  }
}

router.post('/login', async (req,res) => {
  try{
    const user = req.body.user
    const password = req.body.password

    // const isMatch = await bcrypt.compare(password)
    client.getPassword(user).then(async (result) => {
      //console.log
      const hashPassword = result;
      const isMatch = await bcrypt.compare(password,hashPassword)

      if(isMatch){
        console.log("Matched")
      }else{
        res.json({error: "Incorrect Password"})
      }
    }).catch((e) => {
      res.status(406).send('Username Already Taken') //Jose Castanon -- username check should be done when
      registering a user
    })
  }catch(error){
    console.log(error)
  }
}

module.exports = router;

```

#### **Internal Team Member Melissa Code review:**

The review is done by Amir on Melissa's code for the clientProfile page

The code is available on the github branch in  
 "/application/client/component/creatingClient/clientprof.js"

Too big to post here the whole code.

#### **Comments for her code from Amir:**

```
/*
```

- There are some packages being imported into the project which are not being used for best practice I think its better to take them off.
- Some comments available inside the code makes the just for the back-end team easier to navigate through your code for fixing the states of our website.
- Overall great code and usage of css and thank you for making our website look great Melissa

\*/

# Self-Check on Best Practice for Security

The assets being protected in our websites are:

- passwords are encrypted and also salted when placed inside the database.
- Search bar inputs are being secured for any SQL injection.
- Client and profile pages are secured and also visible to the user with correct username and password.
- Data is being communicated to the back-end with the help of proxy for more security

Proof for salting and encrypting password:

Figure 1 - Password encryption

Figure 2 – Adding the Hashed password

Figure 3 – Checking the Hashed password

## Data Validation:

For the forms validation we are implementing errors checkers for such as correct email was input or the user does not enter the primary user name key. Also for the search we're validating the input depending on the side of the service. If the client is searching for a worker, they are only allowed to search by booking # or username of the worker. Bookings are only numbers so if the user input is only number gets validated for the booking id and if it contains letter it would be username. The search for client is designed to not search for any special character in their input string. This is also being offered to the worker side of our website for safety of our database and the user information

- **Password is being validated and hashed**
- **Search bar input is being validated for any mysql injection and detecting ;**

# Self-Check: Adherence to Original Non-Functional Specs

Copy all original non-functional specs as in high level application document published at the very beginning of the class. Then for each say either:

DONE if it is done;

ON TRACK if it is in the process of being done and you are sure it will be completed on time;

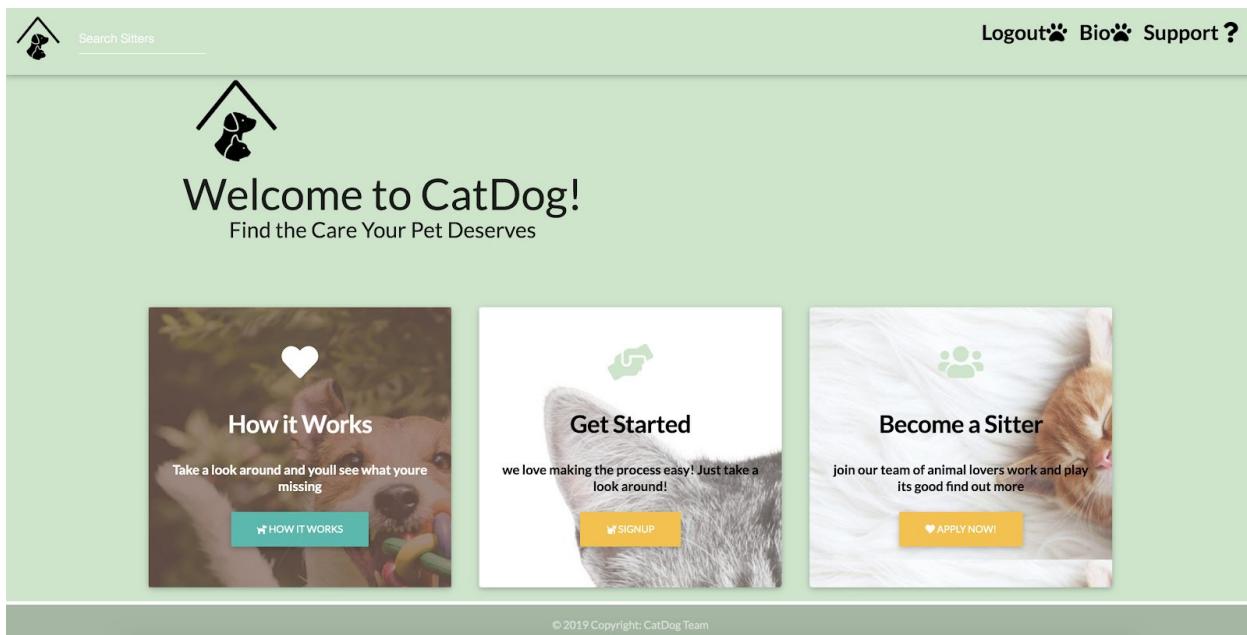
or ISSUE meaning you have some problems and then

| Non- functional Requirements<br><b>Functionality</b>  | Status |
|---|--------|
| 1. The site should be developed and deployed using the stack tools and servers that was approved by the Class CTO.      | Done   |
| 2. Each WWW page needs to have a functional navbar and search bar with the logo included at the top of the page.        | Done   |
| 3. Guest user should be able to see the functionality of the website in a demo before signing up.                       | Done   |
| 4. Application shall be very easy to use and intuitive  | Done   |
| 5. Application shall be hosted and deployed on Gcloud client server as specified in M0.                                 | Done   |
| 6. Application shall be optimized for standard desktop/laptop browsers.<br>All users should be able to contact support. | Done   |
| 7. Client should be able to contact Admin in case of emergency.   | Done   |
| <b>Compatibility:</b>   |        |
| 8. Site should have a logo to be displayed next to the title in every browser.  | Done   |

|  |      |
|--|------|
| 9. The site shall be compatible with the latest version of Chrome browser (77.0.3865.90)   | Done |
| 10. The site shall be compatible with the latest version of Firefox browser (68.0.1)   | Done |
| 11. The site shall be compatible with the latest version of Safari browser (5.1.7)   | Done |
| <b>Performance</b>   |      |
| 12. Loading time for site shall be less than 10 seconds for any page.  | Done |
| 13. Search bar result should be shown in less than 10 seconds.   | Done |
| <b>Security</b>  |      |
| 14. Registered user should be able to login with the credentials it made during sign up.   | Done |
| 15. Client should input his/her name, address and driver's licence # to be collected as data.  | Done |
| 16. Client's pet needs to be California licensed and the ID # needs to be inputted into the data. Data.  | Done |
| 17. Guest user should not be able to see any live data before signing up.  | Done |
| 18. Registered user's password should be encrypted and saved in the database.  | Done |
| 19. Worker should input his name, address, and driver's licence #, and pet sitting certification to be collected as le to change their password if needed. | Done |
| <b>Coding Standards</b>  |      |
| 20. Admin should be able to ban workers and clients if needed  | Done |
| 21. Developer should develop code to be easy to read.  | Done |

|   |      |
|---|------|
| 22. Methods/functions should have one and only one purpose.   | Done |
| 23. Developer should be commenting on the specific methods on his/her code.                           | Done |
| 24. Developer should be consistent in the naming convention of variables.                             | Done |
| 25. Back-End developer should develop noncomplex api.   | Done |
| 26. Developer should write test cases for their functions.  | Done |
| <b>Layout</b>   | Done |
| 27. Website should be professional looking  | Done |
| 28. Website should be user-friendly   | Done |
| <b>Data Integrity &amp; Capacity &amp; Reliability</b>  |      |
| 29. User's data shall be inputted into MYSQL database.  | Done |
| 30. Data in the data base should be back up every week.   | Done |
| 31. The server storage shouldn't exceed 80%   | Done |
| 32. Inform the users if the website is going to be down for maintenance.                              | Done |
| 33. If the server is down, it should be restarted and back on running in less than 1 hour in a month. | Done |
| 34. Pay functionality shall not be implemented.   | Done |

# Screen Shots Of Product



This screenshot shows the "How It Works" section of the CatDog website, viewed in a browser window titled "CatDog". The top navigation bar includes "Logout", "Bio", "Support", and a question mark. The main heading "How It Works!" is centered above two columns. The left column, titled "I am a Client", lists five steps: 1. Create a Client account (pencil icon), 2. Create a Profile for your Pet (paw print icon), 3. Create a Sitting request (hand icon), 4. Select a sitter from the booking requests (person icon), and 5. meet your sitter the D-Day (camera icon). The right column, titled "I am a Sitter", lists four steps: 1. Start by creating a Sitter account (pencil icon), 2. Provide a Pet Sitter Credentials (person icon), 3. Book a client post from the post feed (calendar icon), and 4. Confirm the booking (calendar icon). The bottom of the screen shows a Mac OS X dock with various application icons.

**I am a Client**

- First: Create a Client account**  
Create an account in our website and provide address and other informations
- Second: Create a Profile for your Pet**  
Create a Profile for each of your pets and answer some inquiries that will help our sitters take care of them
- Third: Create a Sitting request**  
Create a booking/sitting request with the date you want your pet to be sitted
- Fourth: Select a sitter from the booking requests**  
Review and Select a sitter from the list of sitters that booked your posts
- Fifth: meet your sitter the D-Day**  
meet your sitter at a specific location and hand him your pet

**SIGNUP**

**I am a Sitter**

- First: Start by creating a Sitter account**  
Create a Sitter account in our website and provide address and other information.
- Second: Provide a Pet Sitter Credentials**  
Our sitters offer a secure boarding services making sure your pet receives the proper treatment
- Third: Book a client post from the post feed**  
Access your post-feed and book a client post that suits your work schedule
- Fourth: Confirm the booking**  
Access notifications to have to confirm that the clients that booked you and confirm the booking
- Fifth: meet the client the D-Day**  
Our sitters offer a secure boarding services to make you happy

**APPLY NOW!**

A screenshot of a web browser window showing the homepage of the website "CatDog". The title bar reads "Chrome File Edit View History Bookmarks People Tab Window Help" and "CatDog Not Secure | 34.67.160.125/clientApp". The page features a large green header with the text "Services for Every Cat and Dog" and "WE KNOW THAT YOUR FUR BABIES ARE MORE THEN JUST PETS". It includes a yellow "START" button and two service sections: "Our Services" (with icons for dog and cat boarding, house sitting, and pet sitting) and "Book with our trusted Sitters" (with a shield icon and a list of four checked-off services: Background Checks, Certified Pet Sitters, Approved by our Team, and Reservation Protected). The footer shows a standard Mac OS X dock with various application icons.

Chrome File Edit View History Bookmarks People Tab Window Help

Not Secure | 34.67.160.125/clientApp

Incognito

Login Bio Support ?

So many have trusted our services

**Susan Smith**  
Joined in 2016  
"The CatDog Sitters sent me updates throughout my cats stay! I would highly recommend!"

**Maria Smith**  
Joined in 2017  
"The CatDog Sitters made sure my dog receives exercise on time! I told all my friends about it and now they've signed up as well. I was so happy with everything I became a regular! Now we're super close to our sitter."

**Susan Smith**  
Joined in 2016  
"CatDog has everything I have always looked for! I used to be so hesitant but the service was so quick and streamline. I would highly recommend!"

Mac OS X Dock: iWork, Utilities, Mail, Safari, Google Chrome, etc.

Chrome File Edit View History Bookmarks People Tab Window Help

Not Secure | 34.67.160.125/clientApp

Incognito

Login Bio Support ?

Services for Every Cat and Dog

Everything your best FurFriend needs! Just check out our highly rated sitters!

**Megan Smith**   
Joined in 2019  
Megan is a comedian living in Nashville

Mac OS X Dock: iWork, Utilities, Mail, Safari, Google Chrome, etc.

A screenshot of a web browser window titled "CatDog" on a non-secure connection. The page features a light green header with a small icon of a person holding a dog. On the right side of the header are links for "Login", "Bio", and "Support". Below the header, there's a user profile section with a thumbnail of a man holding a cat, followed by a status message: "Joined in 2019" and "John is a student living in the Tenderloin SF". To the right of this is a large rectangular area with several horizontal gray bars of varying lengths. In the center of the page, under the heading "So what are you waiting For?", is the text "Its Time to Take your Pets Care To the Next Level" and a prominent yellow "START" button.

A screenshot of a web browser window titled "CatDog" on a non-secure connection. The page has a light green header with a small icon of a person holding a dog. On the right side of the header are links for "Login", "Bio", and "Support". Below the header, there's a large circular icon containing a smaller icon of a dog. Underneath it, the text "Let's Sign Up!" is displayed. Below this, the text "As Our Client We Strive to Make the Registration Processes Easy Peasy" is followed by the subtext "First the Basics". The main content area contains five input fields labeled "userNmae", "First Name", "Last name", "Email", and "Password". Below these fields is a label "Address 1" with a corresponding input field containing the placeholder "1234 Main St, Apartment, studio, or floor".

Chrome File Edit View History Bookmarks People Tab Window Help

Not Secure | 34.67.160.125/SignUp

Login Bio Support ?

userN<sup>ame</sup>

First Name

Last name

Email

Password

Address 1  
1234 Main St, Apartment, studio, or floor

City Choose a Option Zip

Agree to the Terms and Conditions

« CONTINUE

Container

© 2019 Copyright: CatDog Team

Chrome File Edit View History Bookmarks People Tab Window Help

Not Secure | 34.67.160.125/petinfo?user=hakim

Login Bio Support ?

Tell us About Your Furry Friend!

The Main Event

Furry Friend Name Gender

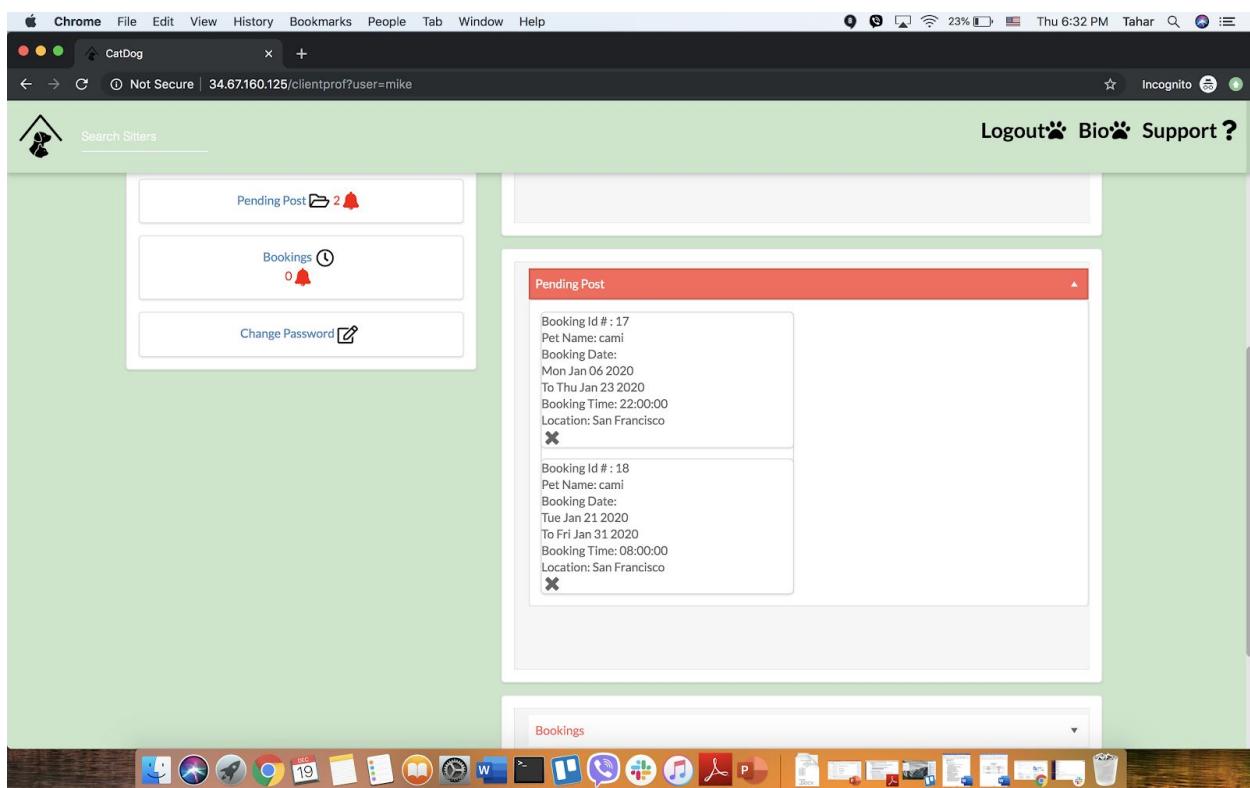
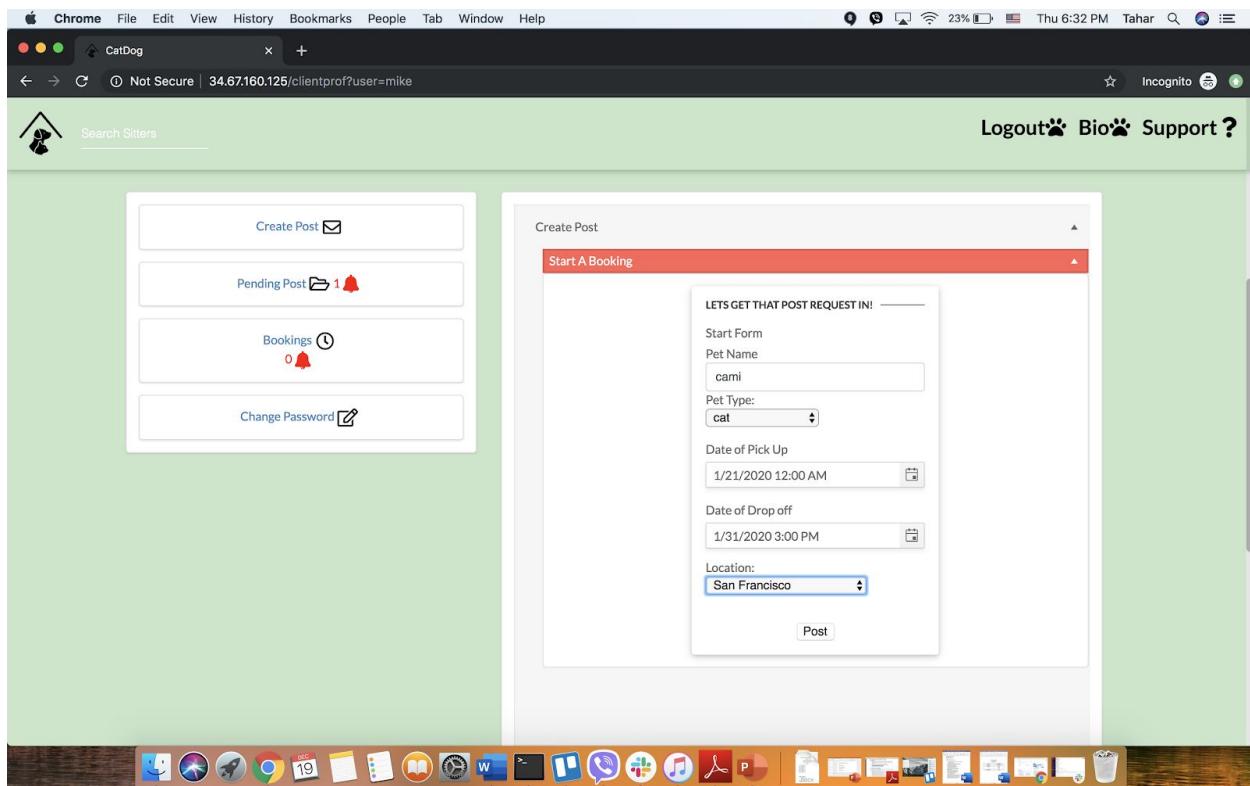
blue Male

Let Us Know Anything Else

Blue is a beautiful dog with blue eyes

« SUBMIT

A screenshot of a Chrome browser window displaying a user profile and a pet profile on the CatDog website. The browser's top bar shows the title "CatDog" and the URL "Not Secure | 34.67.160.125/clientprof?user=mike". The page itself has a light green header with a navigation bar and search bar. On the left, there's a user profile section for "Mike Trump" with a placeholder photo, contact information (Email: mike@trump.com, City: San Francisco), and a "Create Post" button. On the right, there's a "Your Pet Profile" section for a dog named "cami" with a placeholder photo, bio (Bio: cami is a beautiful dog), and a "Create Post" button. A "Pending Post" notification is visible. The bottom of the screen shows the Mac OS X dock with various application icons.



MDBCard title

Get Paid to Play with Furry Animals!

CatDog Makes it Easy By Connecting Your Animal Lover Spirit To The Best Pet Parents!

Work Around Your Own Schedule!

FIND OUT MORE!

## Why is it so great?

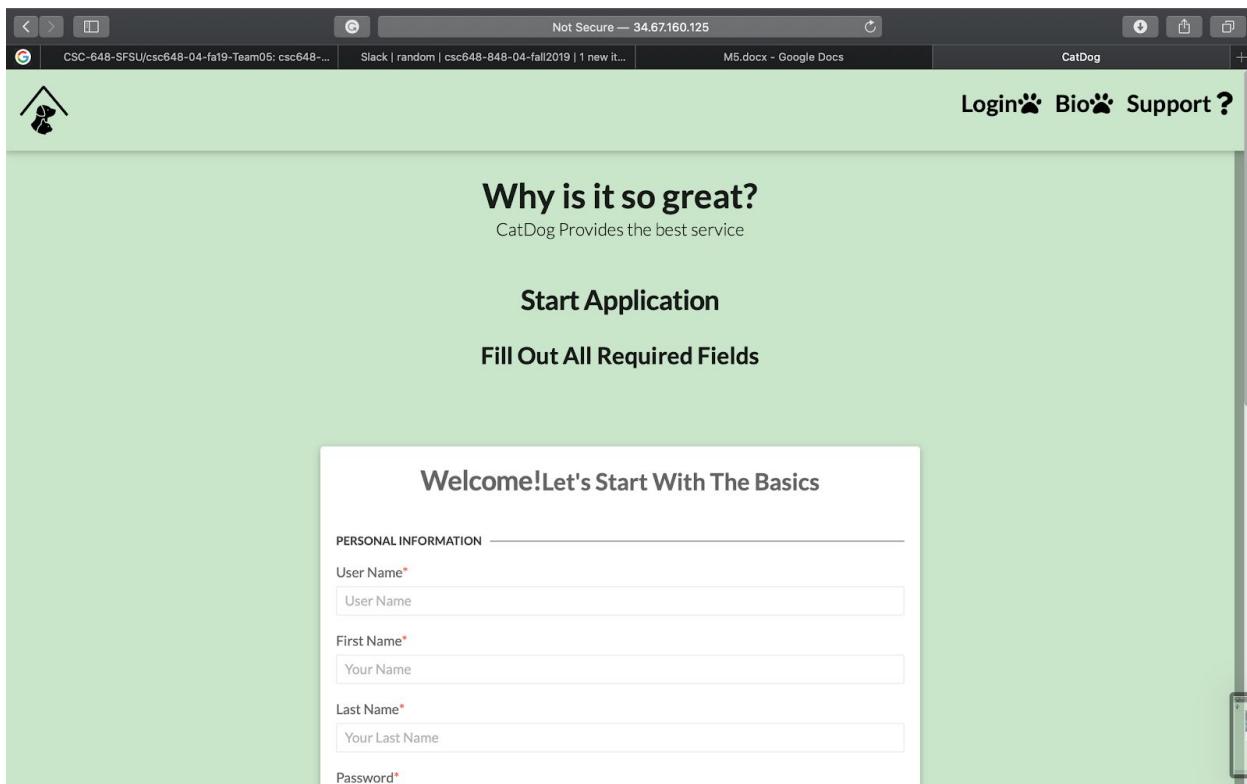
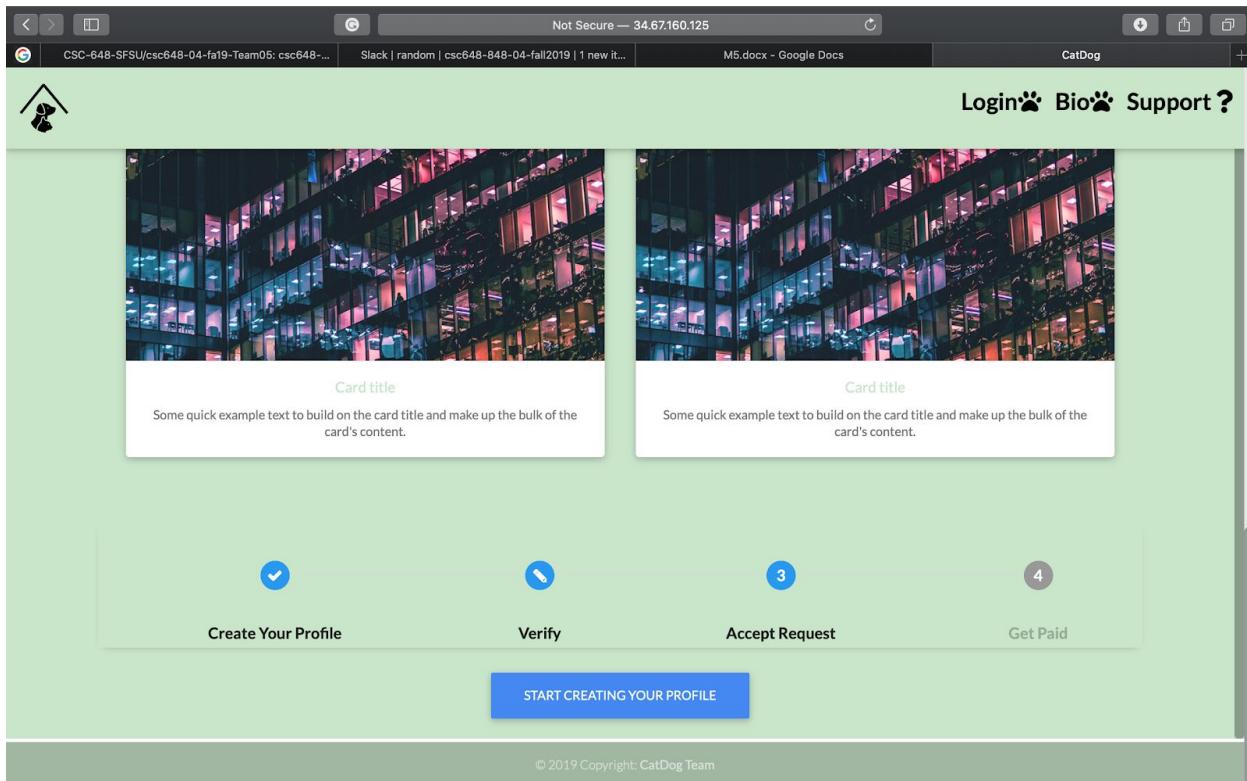
Lore ipsum dolor sit amet, consectetur adipisciing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam.

Flexibility Puts You In Control

- ✓ Lore ipsum dolor sit amet, consectetur adipisciing elit. Reprehenderit maiores aperiam minima assumenda deleniti hic.
- ✓ Lore ipsum dolor sit amet, consectetur adipisciing elit. Reprehenderit maiores aperiam minima assumenda deleniti hic.
- ✓ Lore ipsum dolor sit amet, consectetur adipisciing elit. Reprehenderit maiores aperiam minima assumenda deleniti hic.

Tools To Succeed

- ✓ Lore ipsum dolor sit amet, consectetur adipisciing elit. Reprehenderit maiores aperiam minima assumenda deleniti hic.
- ✓ Lore ipsum dolor sit amet, consectetur adipisciing elit. Reprehenderit maiores aperiam minima assumenda deleniti hic.
- ✓ Lore ipsum dolor sit amet, consectetur adipisciing elit. Reprehenderit maiores aperiam minima assumenda deleniti hic.



Not Secure — 34.67.160.125

CSC-648-SFSU/csc648-04-fa19-Team05: csc648-... Slack | random | csc648-848-04-fall2019 | 1 new item... M5.docx - Google Docs CatDog +

The screenshot shows a web-based application for pet sitters. At the top right are links for 'Login', 'Bio', and 'Support'. On the left, there's a small icon of a dog in a house. The main form area contains fields for 'Email \*' (with placeholder 'Your Email'), 'Enter phone number' (with placeholder '(359) 884-1233'), and 'Please Enter Your Birthday' (with placeholder 'YYYY-M-D'). Below these are fields for 'Phone Number \*' (placeholder 'Phone Number') and 'Let's Get You Verified' (with a sub-instruction 'Input Pet Sitter Credential Number'). A note states 'In Order to Complete the Application Process A Pet Sitter Certificate is Required'. There's a field for 'Credential Number \*' (placeholder 'Credential Number'). The section 'Background Check Information' includes a 'Drivers License' field with a placeholder 'Driver Licence \*'.

Not Secure — 34.67.160.125

CSC-648-SFSU/csc648-04-fa19-Team05: csc648-... Slack | random | csc648-848-04-fall2019 | 1 new item... M5.docx - Google Docs CatDog +

The screenshot shows a user's profile page. At the top right are links for 'Logout', 'Bio', and 'Support'. On the left, there's a small icon of a dog in a house. The main area starts with 'Welcome to your Sitter Profile'. It features a large profile picture of a man with brown hair and a beard. To the right, there are two boxes: 'Your Profile' containing 'Private Info' (Email: melissaestrada101009@gmail.com, Address: 99 Caine Avenue San Francisco 94112) and 'Public Info' (Certification Number: 44444, Phone Number: 7142034135). Below this are sections for 'Available Jobs' (6 notifications), 'Accepted Jobs' (0 notifications), and a 'Change Password' button. To the right are two large, empty scrollable boxes labeled 'Available Jobs' and 'Accepted Jobs'.

The screenshot shows a web browser window with a light green header bar. The header contains the text "Not Secure — 34.67.160.125", the URL "CSC-648-SFSU/csc648-04-fa19-Team05: csc648...", "Slack | random | csc648-848-04-fall2019 | 1 new item...", "M5.docx - Google Docs", and "CatDog". On the far right of the header is a "+" button.

The main content area has a white background. At the top left is a small black icon of a dog's head. To its right is a navigation menu with three items: "Available Jobs" (with a red bell icon), "Accepted Jobs" (with a red bell icon), and "Change Password".

The central part of the screen displays two job entries under the heading "Available Jobs".

**Job 1:**

- Pet Name: Sadada
- Pet Type: Cat
- Owner: Ben
- Booking Date:  
Tue Jan 28 2020  
To  
Wed Jan 29 2020
- Booking Time: 08:00:00
- Location: San Francisco

**Job 2:**

- Pet Name: Tommy
- Pet Type: Dog
- Owner: Amir
- Booking Date:  
Thu Jan 30 2020  
To  
Wed Jan 29 2020
- Booking Time: 08:00:00
- Location: San Francisco

Below each job entry is a red "Accept Job" button with a white icon.

On the right side of the screen, there is a small inset image showing a mobile device displaying the same application interface.

This screenshot shows the same web browser window and header as the previous one.

The main content area now displays a single job entry under the heading "Accepted Jobs".

**Accepted Job:**

- Pet Name: Cami
- Pet Type: Cat
- Owner: Mike
- Booking Date:  
Tue Jan 21 2020  
To  
Fri Jan 31 2020
- Booking Time: 08:00:00
- Location: San Francisco

Below the job entry is a red "Accept Job" button with a white icon.

Welcome to your Sitter Profile

Your Profile

Private Info

Email: melissaestrada101009@gmail.com  
Address: 99 Caine Avenue San Francisco 94112

Public Info

Certification Number: 44444  
Phone Number: 7142034135

Available Jobs 6

Accepted Jobs 0

Change Password

Available Jobs

Pet Name: Sadada  
Pet Type: Cat  
Owner: Ben  
Booking Date:  
Tue Jan 28 2020  
To  
Wed Jan 29 2020  
Booking Time: 08:00:00  
Location: San Francisco

Pet Name: Tommy  
Pet Type: Dog  
Owner: Amir  
Booking Date:  
Thu Jan 30 2020  
To  
Wed Jan 29 2020  
Booking Time: 08:00:00  
Location: San Francisco

Pet Name: Tommy

Accept Job

The screenshot shows a web browser window with the following details:

- Address Bar:** Not Secure — 34.67.160.125
- Open Tabs:** CSC-648-SFSU/csc648-04-fa19-Team05; csc648-... (highlighted), Slack | random | csc648-848-04-fall2019 | 1 new item..., M5.docx - Google Docs, CatDog

The main content area displays a booking interface for a pet named Cami:

- Pet Details:** Pet Name: Cami, Pet Type: Cat, Owner: Mike, Booking Date: Tue Jan 21 2020, To: Fri Jan 31 2020, Booking Time: 08:00:00, Location: San Francisco.
- Action Button:** A red button labeled "Accept Job" with a dog icon.

Below this, there is a sidebar titled "Accepted Jobs" which is currently empty.

# Screen shots of key DB table

This screenshot shows the MySQL Workbench interface with the 'Post' table selected in the Navigator pane. The table structure includes columns: id, clientUser, datePosted, dateRequested, location, active, type, name, and worker. The data grid displays 10 rows of sample data. The right-hand panel contains various tools like SQLAdditions, Form Editor, Field Types, Query Stats, and Execution Plan.

| id | clientUser | datePosted               | dateRequested            | location | active | type | name   | worker |
|----|------------|--------------------------|--------------------------|----------|--------|------|--------|--------|
| 3  | amir       | 2019-02-28T08:00:00.000Z | 2019-02-23T08:00:00.000Z | Ca       | 1      | cat  | Tommy  | amir   |
| 4  | swade      | 2019-12-05T19:57:17.946Z | 2019-02-28T19:57:25.271Z | Wa       | 0      | cat  | ff     | swade  |
| 5  | swade      | 2019-12-05T19:57:17.946Z | 2019-02-28T19:57:25.271Z | Wa       | 0      | cat  | ff     | swade  |
| 6  | Johnn      | 2019-02-20T08:00:00.000Z | 2019-04-10T07:03:00.000Z | Ca       | 0      | dog  | Cooper | Johnn  |
| 7  | Johnn      | 2019-02-27T08:03:00.000Z | 2019-03-17T07:00:00.000Z | Ca       | 0      | cat  | Cooper | Johnn  |
| 8  | amir       | 2020-01-08T08:00:00.000Z | 2020-01-24T20:00:00.000Z | Ca       | 0      |      |        | amir   |
| 9  | amir       | 2020-01-31T08:00:00.000Z | 2020-01-31T08:00:00.000Z | Ca       | 0      |      |        | amir   |
| 10 | amir       | 2020-01-31T08:00:00.000Z | 2020-01-31T08:00:00.000Z | Ca       | 0      |      |        | amir   |

This screenshot shows the MySQL Workbench interface with the 'Client' table selected in the Navigator pane. The table structure includes columns: user, firstname, lastname, street, city, zipCode, email, and password. The data grid displays 10 rows of sample data. The right-hand panel contains various tools like SQLAdditions, Form Editor, Field Types, Query Stats, and Execution Plan.

| user       | firstname | lastname  | street                    | city          | zipCode | email                         | password                                   |
|------------|-----------|-----------|---------------------------|---------------|---------|-------------------------------|--|
| amir       | Amir      | Amorphous | asdas                     | asdas         | 95538   | ar.amorphous@gmail.com        | \$2a\$10\$UUE1A6jWV02z8C0K03Q4fjQ3u4P...   |
| ane        | ane       | ane       | asdas                     | asdas         | 95538   | ar.amorphous@gmail.com        | \$2a\$10\$GQ7vWNLSQLbdot1L5eketad025Cb...  |
| amrjanovic | Amir      | Amorphous | 995 John Muir Dr, Apt 618 | san francisco | 94132   | ar.amrjanovic@gmail.com       | \$2a\$10\$EQAQd9k9N9NWh5eLBDnJQ/yVjg...    |
| asaddad    | asdad     | reza      | asdas                     | asdas         | 95556   | ar.amrjanovic@gmail.com       | \$2a\$10\$ASQDR6sA3Lc0205uy0fW9Ht4O...     |
| ben        | ben       | ben       | ben                       | ben           | 94132   | ben@gmail.com                 | \$2a\$10\$Pd7DmXQJLXqT2zIwH4owuU.S...      |
| benjamin   | benjamin  | lelio     | 1214 Main st              | san francisco | 94132   | benjamin.lelio@gmail.com      | \$2a\$10\$Pd7DmXQJLXqT2zIwH4owuU.S...      |
| bassas     | bassan    | reza      | asdas                     | asdas         | 95556   | ar.bassanreza@gmail.com       | \$2a\$10\$H4l3m4hDfMhCQ2Q5enAO4C8tBf...    |
| mellissa   | mellissa  | m         | sada555                   | San Francisco | 94132   | mellissastredd10109@gmail.com | \$2a\$10\$H4l3m4hDfMhCQ2Q5enAO4C8tBf...    |
| mike       | mike      | trump     | 1234 trump street         | San Francisco | 94115   | mike@trump.com                | \$2a\$10\$GOM6gobuOLVfImh8DXdnL7DgAu4...   |
| risog      | Risog     | Amir      | Amorphous                 | asdas         | 95556   | ar.amrjanovic@gmail.com       | \$2a\$10\$gDUQX0iGZCZ...H4fr32eBfRofdV3... |
| swade      | swade     | dhanve    | asdas                     | asdas         | 95536   | shane.au.wade@gmail.com       | \$2a\$10\$HOn6CLGyMENtY6es0f5uTeysaf79P... |

MySQL Workbench

File Edit View Query Database Server Tools Scripting Help

Schemas

catdog

Tables

Client Pet Post subscription Worker

Views

Stored Procedures Functions

sys

Post Worker Worker Client Worker Client Post Administration - Data Import/Res... Client Pet Pet Worker

1 • SELECT \* FROM catdog.worker;

Result Grid | Filter Rows: [ ] | Edit: [ ] | Export/Import: [ ] | Wrap Cell Content: [ ]

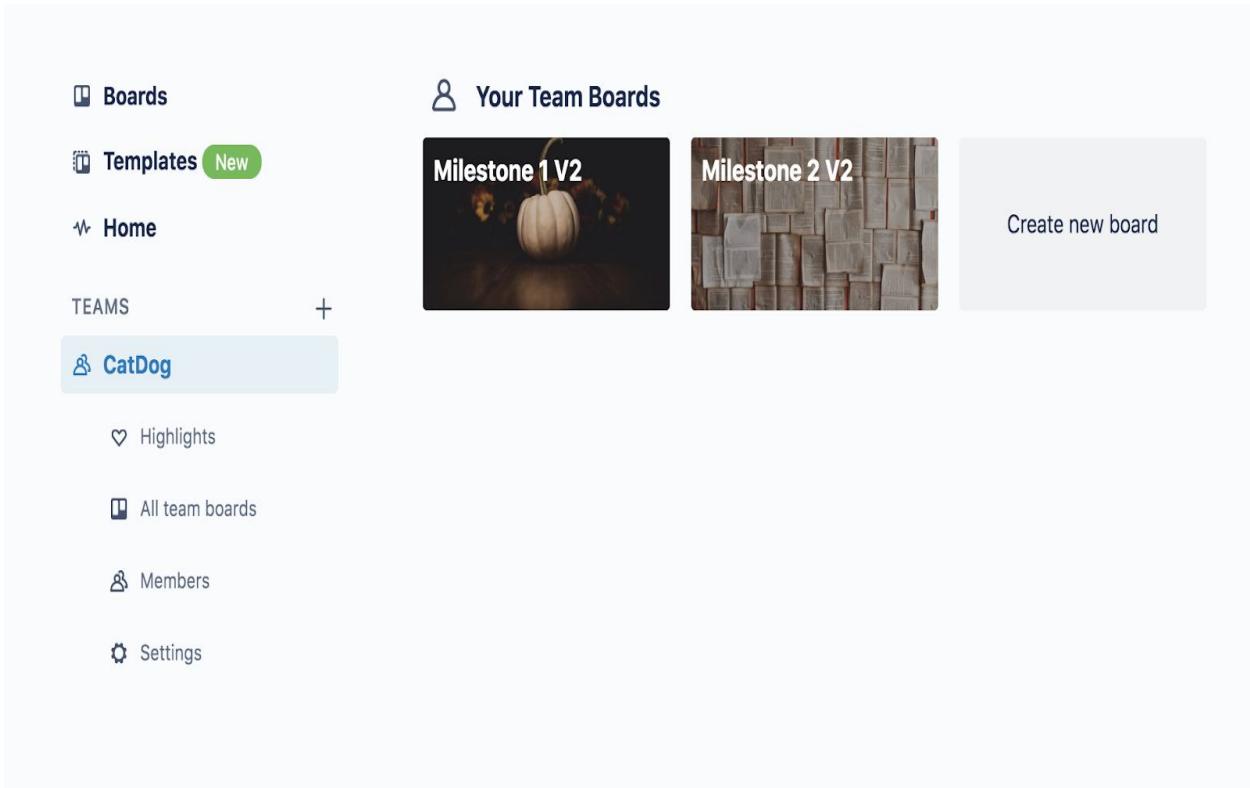
| user    | firstName | lastName   | address                                   | password   | email                   | certification | phone |
|---------|-----------|------------|---|--|-------------------------|---------------|-------|
| aaa     | anjom     | reza       | 595 John Muir Dr, San Francisco, Ca 94132 | this   | ar.anjomshoaa@gmail.com | 12313         | NULL  |
| aas     | anjom     | reza       | adsa                                      | \$2a\$04\$juwO2WeMEEmPeAtnFA8...TmEkJuu...                               | ar.anjomshoaa@gmail.com | 555           | NULL  |
| aasd    | anjoma    | reza       | adsa                                      | \$2a\$04\$keWVPiVqzBQcuSa7Q).OnqEWgpd...ar.anjomshoaa@gmail.com          | 132123                  | NULL          |       |
| amir    | Amir      | Anjomshoaa | 595 John Muir Dr, San Francisco, Ca 94132 | \$2a\$04\$3yGvR7XQ8jKxeZcOuenmpgP0dVl...ar.anjomshoaa@gmail.com          | 111                     | 5305741700    |       |
| asdeddq | anjom     | reza       | adsa                                      | \$2a\$04\$js3PFPhp7n2awSArYJUUm7P;Pufdv...ar.anjomshoaa@gmail.com        | 31231                   | NULL          |       |
| cpt.loc | loc       | nguyen     | 99 Caine Avenue San Francisco 94112       | \$2a\$04\$6t7QD06EAxJVi6UetTuh..bV4u9LV...melissaazeada10.1009@gmail.com | 44444                   | 7142034135    |       |
| ivan    | anjom     | reza       | adsa                                      | \$2a\$04\$8uFverx00y0Ormd000Y4013bzfO3...ar.anjomshoaa@gmail.com         | 111111                  | NULL          |       |
| john    | john      | john       | john                                      | \$2a\$04\$yv0QeXeZC.M0AxY1.3N6EPNwTq...john@john...                      | 55555                   | NULL          |       |
| rksog   | anjom     | reza       | adsa                                      | \$2a\$04\$fpGYS8d51y)ED59xkXONfTBzqhp...ar.anjomshoaa@gmail.com          | 55555                   | NULL          |       |
| rksogg  | anjom     | reza       | adsa                                      | \$2a\$04\$ULACf6j..0\$0\$#fa0BQmfF720oGgi...ar.anjomshoaa@gmail.com      | 55555                   | NULL          |       |
| rksogga | anjom     | reza       | adsa                                      | \$2a\$04\$75FUuMS)(eTq)hV\$p92vqepvPbWOn5...ar.anjomshoaa@gmail.com      | 55555                   | NULL          |       |
| rkiss   | anjom     | reza       | adsa                                      | \$2a\$04\$Sc7XTP82tsJzp.sqABhNO)fz13vR3vcz...ar.anjomshoaa@gmail.com     | 55555                   | NULL          |       |
| sadd    | anjom     | reza       | adsa                                      | \$2a\$04\$cFn5mgovrQfTCysee+4JxHV4cE...ar.anjomshoaa@gmail.com           | 123312                  | NULL          |       |
| tahar   | tahar     | tahar      | 1829 bush st                              | \$2a\$04\$9Y2hZx453HaaxUj7YTF1.6CjRFV3fx...aaaaaaaaaaaa@gmail.com        | 12345                   | 5305741700    |       |

Object Info Session Output

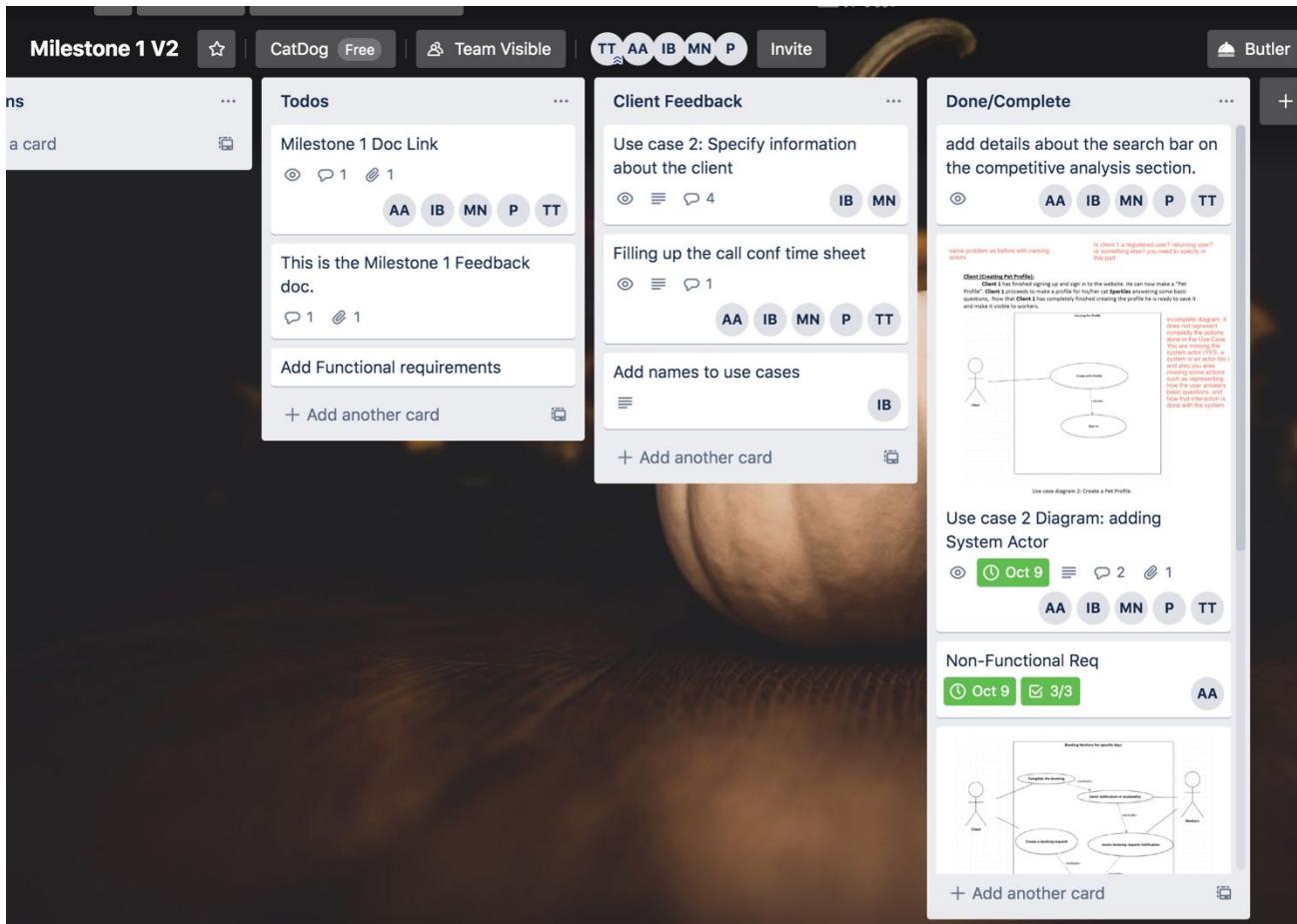
Worker 1 x

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

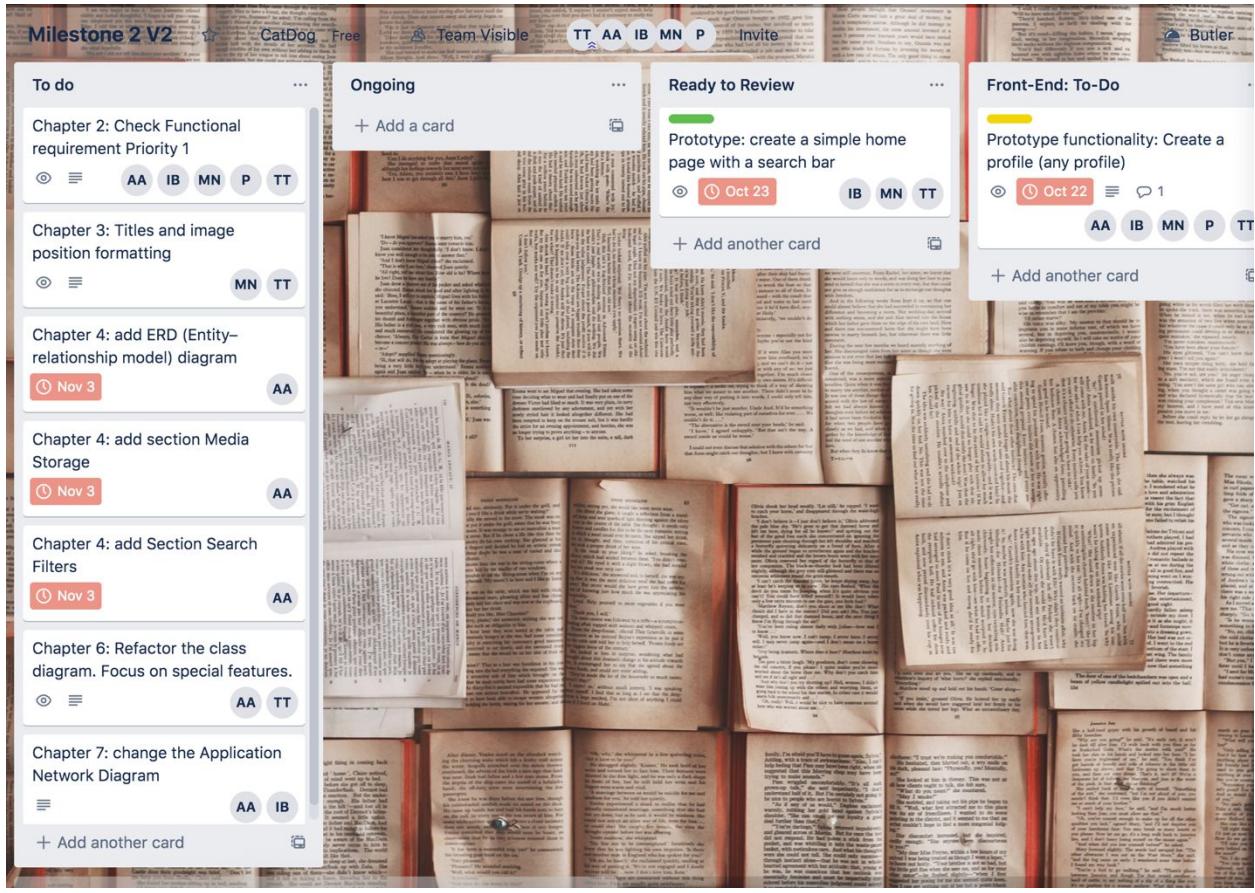
# Screen Shots Task Management System:



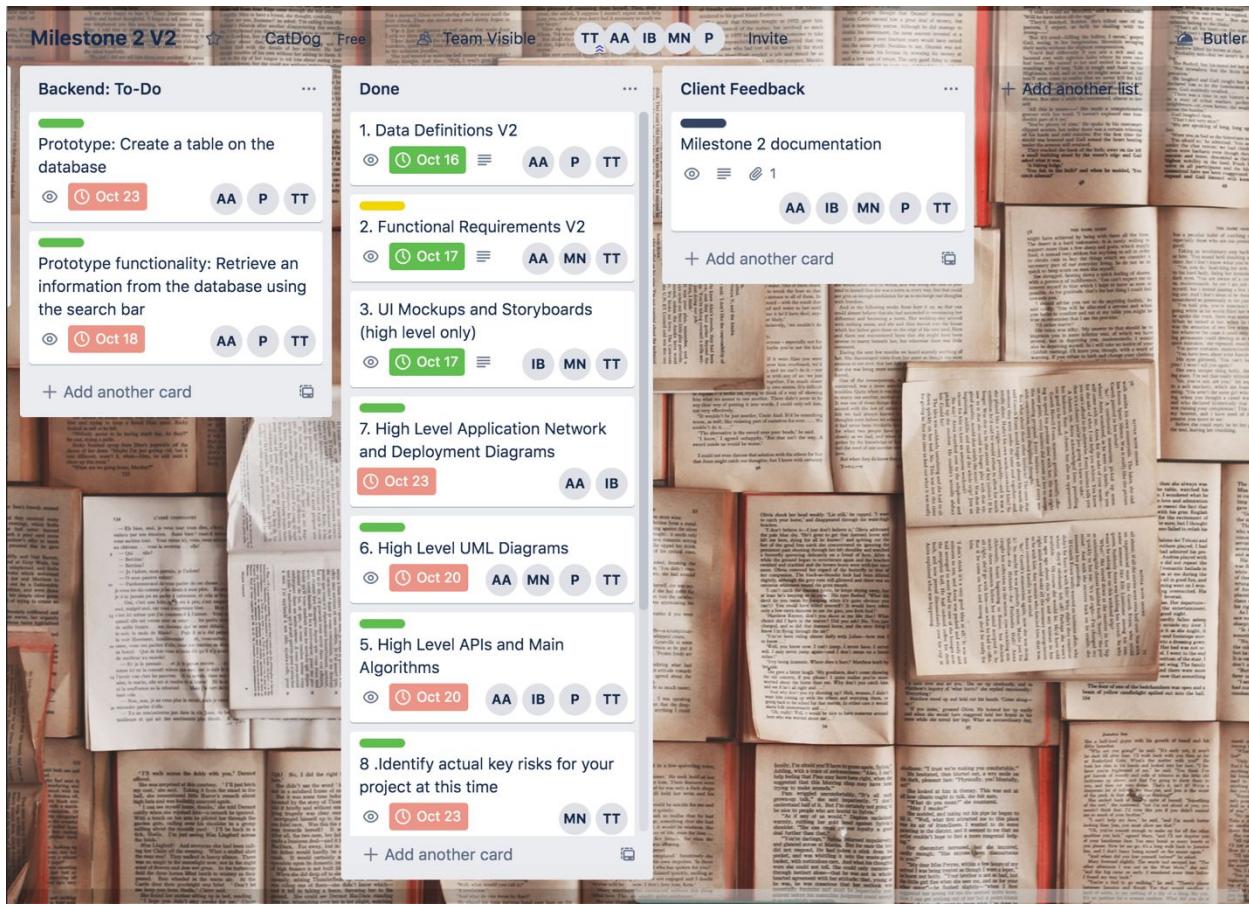
Trello boards M1 and M2



Trello board for Milestone 1



Trello board for Milestone 2 part 1.



Trello board for Milestone 2 part 2.

**#teamtamix**

Tuesday, December 10th

Ivan Briseno 12:08 PM  
Can't meet today, I have to finish my other project we expo it tomorrow

Amir Anjomshoaa 12:09 PM  
good luck

Ivan Briseno 12:10 PM  
I could do a video call for a bit cause Ima be at work  
just send me a link

Amir Anjomshoaa 3:53 PM  
I'll be there a bit late

Xiaopeng Rong 5:56 PM  
Hi everyone, hope you guys had a very good break. I am so sorry for disappearing a week, I know I am a bad person who doesn't know how to communicate and cooperate with other. And professors told me I will have a much lower grade than your guys, because I didn't contribute. It is fair enough for everyone. I might retake this class next semester. I know it is too late, but if it is possible I still want to participate in to complete this class. If there is something I could help. Thank you guys!

Yesterday

Amir Anjomshoaa 8:43 AM  
@channel we need to start on M5. If possible everyone read M5 and start sending email to each other for the assesment we have to do and agree

Tahar Touati 11:23 AM  
I will send it to you guys this evening

Tahar Touati 11:30 AM  
other than that, I will create an M5 the product summary from M4 and copied version of M1V2 M2V2 M3, screenshots of our trello boards and send it to you here tonight. we will need to add to it M4V2 and Screenshots of principeal DBs + the two remaining chapters (screenshots of contribution emails and lesson learned)

Amir Anjomshoaa 11:31 AM  
Yea M4V2 though im gonna finish it today and try to push it to master im gonna finalize the functional requirements  
That sounds good thank you @Tahar Touati

Tahar Touati 11:32 AM  
okk ofc

Xiaopeng Rong 2:58 PM  
I could print the documentation, if you guys need print work help. Just let me know! Thanks!

Tahar Touati 5:44 PM  
@channel I sent you guy the contribution email, you have to reply to all of us with you contribution to the project in a list format and numbers of commits at the end

Tahar Touati 5:50 PM  
@Xiaopeng Rong I think Melissa is going to print M5

Xiaopeng Rong 5:51 PM  
@Tahar Touati OK. No problem

Message teamtamix

Downloads

- Final Review (1).pdf
- Milestone 2 V2.docx
- Milestone 2 V2.docx
- Thornton Hall.m4a
- Milestone 2 V2.pdf
- Milestone 2 V2.pdf
- Image from iOS.png
- IMG\_20191024\_125116.jpg
- CSC648-B48 Fall 2019 Milestone1 Team 05.docx
- CSC648-B48 Fall 2019 Milestone1 Team 05.docx

Team slack channel

 New conversation

 melissa estrada, AmirRez... Dec 1  
You were in a video call

 AmirReza Anjomshoaa, I... Nov 26  
You were in a video call

 melissa estrada, Ivan, A... Nov 17  
You were in a video call

 AmirReza Anjomshoaa, I... Nov 3  
You were in a video call

 melissa estrada, AmirRez... Oct 30  
You were in a video call

 melissa estrada, Ivan, A... Oct 20  
You were in a video call

 Ivan, melissa estrada, A... Oct 16  
You were in a video call

 AmirReza Anjomshoaa, ... Oct 13  
You were in a video call

 Ivan, melissa estrada, pa... Oct 13  
You were in a video call



Google hangout Team conference calls.

# Team Members Contribution:



Tahar Touati

Sun 12/15/2019 2:43 AM

Melissa Estrada; Ivan Jesus Briseno; Xiaopeng Rong; Amir Reza Anjomshoaa ✎



Hello Team 05,

This is my contribution to the project:

- Contribution to M1 (functional requirements, use case diagrams).
- Contribution to M2 (functional requirements V2, class diagram, deployment diagram).
- Contribution to M3 (added feedbacks from Vertical prototype demonstration).
- Contribution to M4 (Usability testing, QA testing).
- Scheduled and attended all meetings.
- Created and managed Trello boards for M1 and M2.
- Sent Weekly report to class CEO.
- Implemented demo page.

Number of commits: 30



Ivan Jesus Briseno

Thu 12/19/2019 7:58 AM

Tahar Touati; Amir Reza Anjomshoaa; Melissa Estrada; Xiaopeng Rong ✎



Number of Commits: 15

1. Contributions to documentation
2. Attended some of the scheduled group meetings either in person or by conference call
3. Contributed to Trello Management
4. UI/UX Implementation

AA

Amir Reza Anjomshoaa

Fri 12/20/2019 3:45 AM

Ivan Jesus Briseno; Melissa Estrada; Tahar Touati; Xiaopeng Rong ⇐

## Contributions

1. Back-end Lead
2. Assisted with front-end
3. Attended all meetings
4. Contributions to all documentation
5. Implementation of APIs
6. Task management

Number of Commits: 60

Interesting.

Got it, thanks!

Got it!

Are the suggestions above helpful? Yes No

ME

Melissa Estrada

Thu 12/19/2019 4:45 PM

Ivan Jesus Briseno; Tahar Touati; Xiaopeng Rong ✎

Melissa Estrada

...

---

**From:** Melissa Estrada <mestrada7@mail.sfsu.edu>

**Sent:** Wednesday, December 18, 2019 5:44:53 PM

**To:** Amir Reza Anjomshoaa <aanjomshoaa@mail.sfsu.edu>

**Subject:** Re: Contribution to CSC 648 Project

- Contributed to all documentation
- designed website
- implemented front end
- 71 commits
- attended all meetings
- attended all-conference calls

thanks

XR

Xiaopeng Rong

Thu 12/19/2019 9:46 PM

Tahar Touati; Melissa Estrada; Amir Reza Anjomshoaa; Ivan Jesus Briseno ✎



This is my contribution to the project:

- Contribution to M1 (data definition)
- Contribution to M2 (functional requirements, deployment diagram)
- Contribution to M3 (tried to implement the demo page)
- working on google doc and attending some of the in person weekly meetings

Number of commit on GitHub branch: 5

# Post analysis – lessons learned:



Tahar Touati

Thu 12/19/2019 8:45 PM

Amir Reza Anjomshoaa; Melissa Estrada; Ivan Jesus Briseno; Xiaopeng Rong



Hi guys, this is the post analysis I made.

## Main challenges:

- Lack of knowledge of software stack.
- Communication problems at the beginning of the project.
- Putting too much pressure and work for some of the team members.
- Difficulties to handle conflicts and problems at the beginning.

## Future Corrections of Main Challenges:

- Spend more time learning software stack and preferably plan group study session with team members.
- More involvement in the development part for all team members.
- Better distribution of the work with respect of team members skills and schedule, so each member feels a sense of responsibility and fairness at the same time.
- Better use of SCRUM process management to manage meetings.
- Organizing team activities outside of project to create a more friendly atmosphere. Which will encourage team work and improve team communication.
- Define a clear communication process which transforms critics into feedbacks.