SW Engineering CSC 648/848 Fall 2019

CATDOG

Group Number 5

Tahar Touati, ttouati@mail.sfsu.edu

Amir Anjomshoaa

Ivan Briseno

Melissa Estrada

Xiaopeng Rong

"Milestone 1"

09/30/2019

History Table	
Submission No. 1	Feedback

Table of Contents

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 an ethic service for their pet companions. We understand the demand for this service in San Francisco. In this city, there is a running joke that there are more pets residing than there are children and according to the U.S census San Francisco has the lowest percentage of children of any major city.

To emphasize the quality of pet care we will provide and the availability of our service. 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 and continue to receive the care they themselves would provide. We would provide this service with a certified staff that is insured and bonded.

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 a number of 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 realizing 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 without company as easy as possible starting with our user interface.

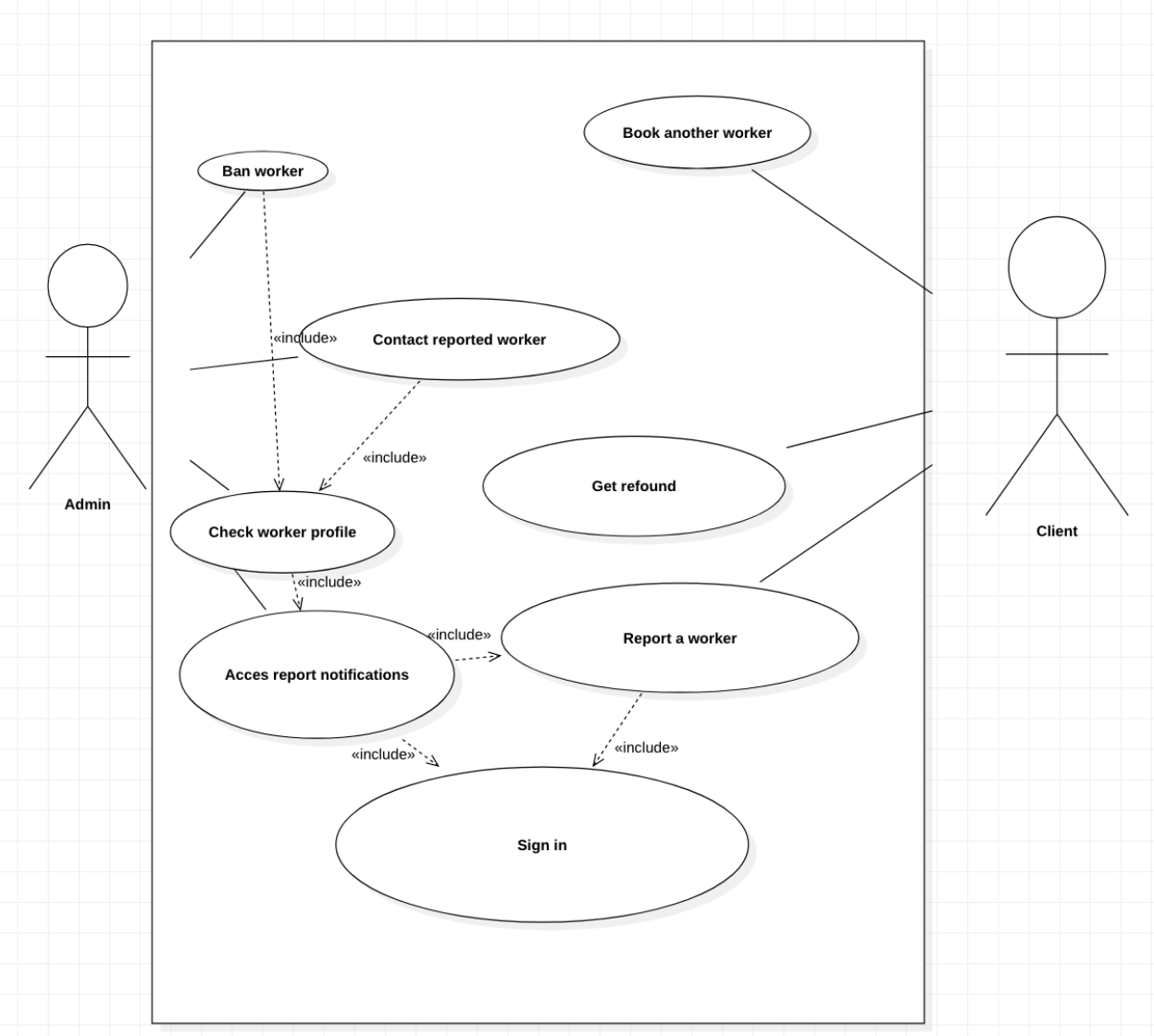
CatDog is hassle-free and simple, our backend engineers have integrated a system that provides thoroughly vetted sitters; making the safety of pets our highest priority. While our frontend engineers have made our interface slick and streamline by creating a minimal approach to our user interface. In this interface, we put our focus on pets by building a specified animal profile displaying the needs of the pet forming a more personalized pet sitter experience. We are focused on generating a fast, secure, and efficient application.

Our team consists of five individuals Tahar Touati, Amir Anjomshoaa, Ivan Briseno, Melissa Estrada, and Xiaopeng Rong. We came together to solve a problem that Melissa was experiencing first hand she shared with us how hard it was to find the perfect sitter for her pet and the stress of traveling with an animal. Our solution to this problem stemmed from personal experiences and personal expectations and understanding that to some pets are the highest investment.

2-Main Use Cases:

Admin - Privileges/Technical Support/Customer Support:

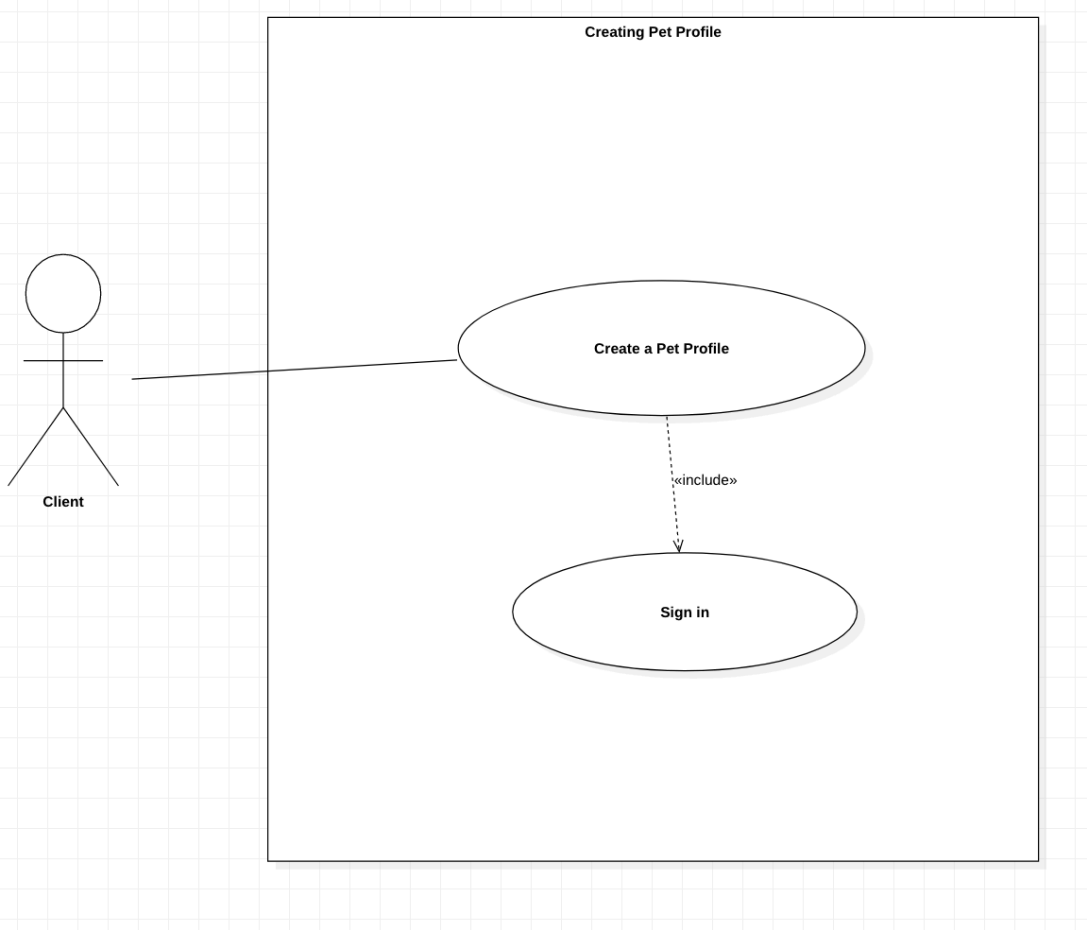
Admin 1 receives a notification marked as “Urgent”. **Admin 1** logs into their internal company email and notices a report message from **Client 1** reporting **Worker 1** for a “No Show at Specified Time”. This is **Worker 1** first “No Show” out of two allowed. Were **Worker 1** to receive a second “no show” **Admin 1** will enforce a ban on the worker and remove **Worker 1** profile. **Admin 1** accesses the internal database to retrieve **Worker 1** information and proceeds to contact **Worker 1**. **Admin 1** contacts **Client 1** regarding the situation and gives them 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):

Client 1 has finished signing up and sign in to the website. He can now make a "Pet Profile". **Client 1** proceeds to make a profile for his/her cat **Sparkles** answering some basic questions, Now that **Client 1** has completely finished creating the profile he is ready to save it and make it visible to workers.



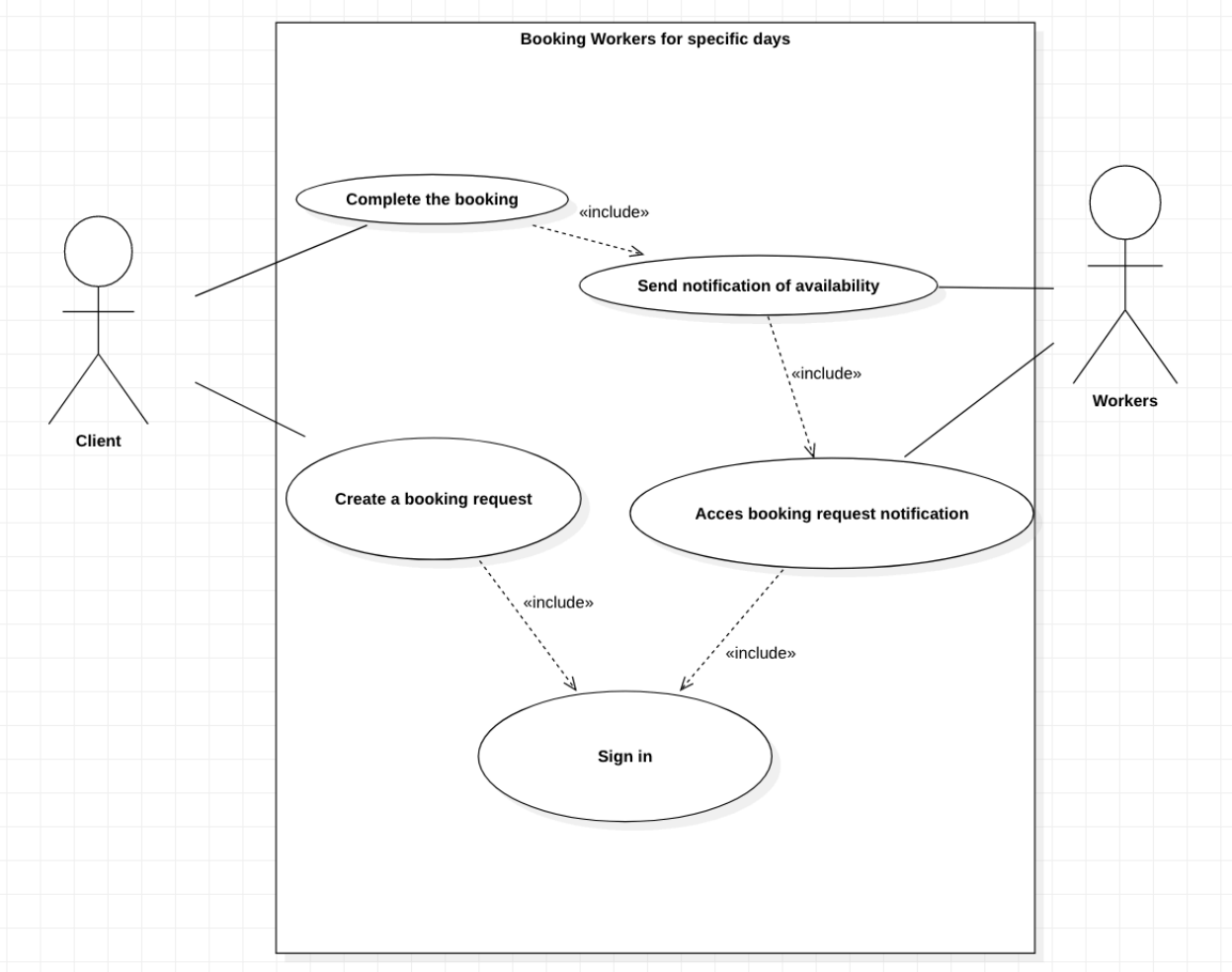
Use case diagram 2: Create a Pet Profile.

Client (Booking Worker for Specific Days):

Client 1 has an important work trip coming up in a week and needs someone to look over his/her cat **Sparkles** for three days**. Client 1** logs into his/her client account and proceeds to the option of "create a booking request" specifying the three days which he/she will be gone. After completing the required booking information and agreeing to the required "Terms & Conditions", **Client 1** can now receive notifications from workers who are available for the specified days. **Client 1** receives a notification from a **Worker 1** saying they are available for the specified dates, and proceeds to check **Worker 1** profile and decides to complete the booking with **Worker 1**.

Worker (Request Client Booking):

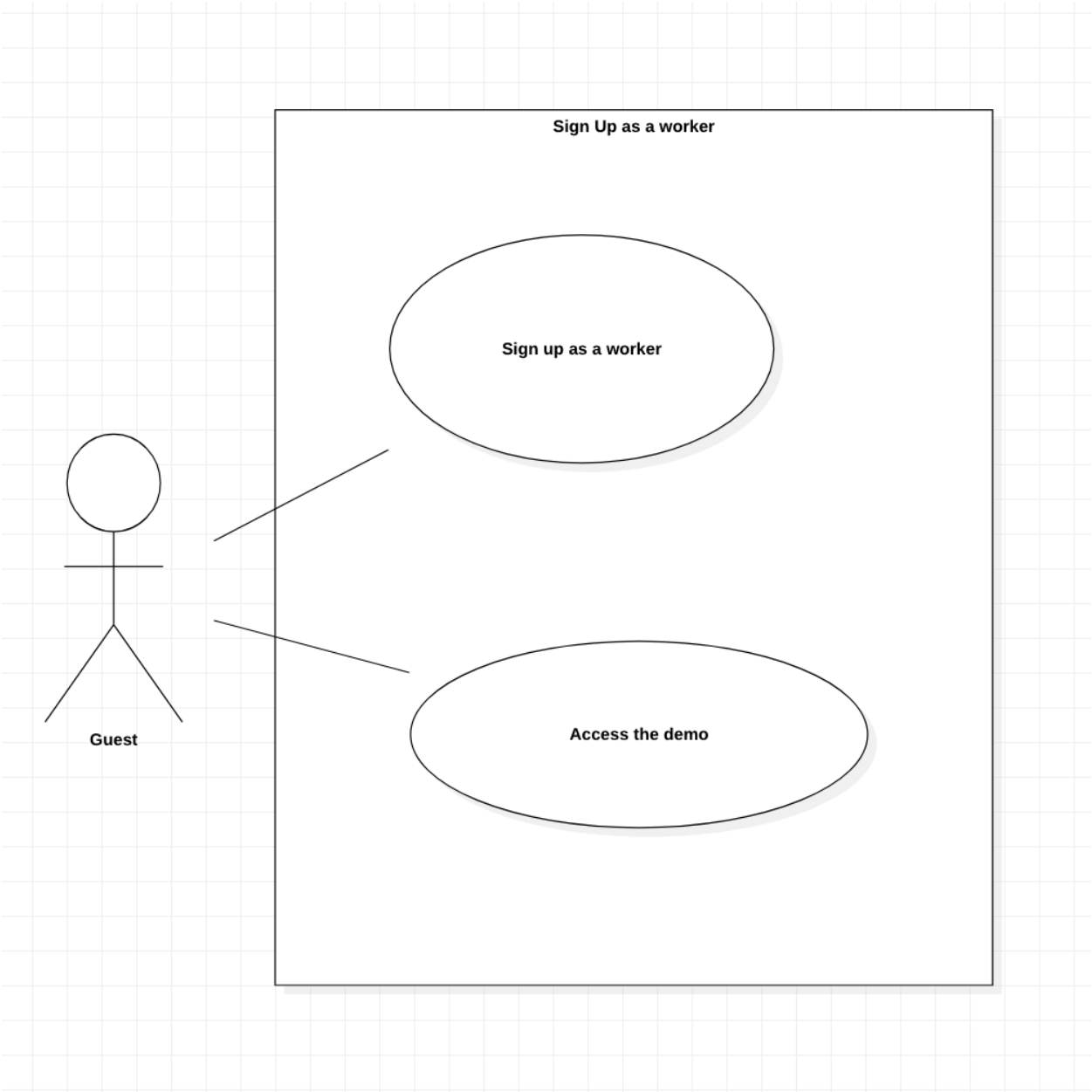
Worker 1 has some free time since it's summer and school's out so he/she decides to change his availability on their worker profile to "Open". Now that **Worker 1** changed his/her availability he/she stars receiving notifications of clients looking for a worker. **Worker 1** finds a client that will be gone for three days and after looking at their "Pet Profile" **Worker 1** decides to "Request This Booking" sending a message to the client saying that **Worker 1** is available for the specified dates and is requesting the booking.



Use case diagram 3: Worker (Request Client Booking) / Client: Book Worker

Guest (Register/Sign Up as Worker):

Guest 1 attends college 4 days out of the week and has afternoons, evenings and nights mostly free so **Guest 1** comes to the conclusion that some extra income and working around your own schedule would be nice. **Guest 1** hears about the CatDog website and decides to browse to the CatDog website, after accessing a demonstration of the website, **Guest 1** Decides to "Sign Up".



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, and is in charge of managing the users.

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 does not have any access to the live data available on the site.

Pet: A client's pet 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.

Demo: An interactive tutorial for Guest users to be able to get an understanding of how the site works.

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.

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.

Client:

4. Clients shall be able to sign in to the website.
5. Clients shall be able to log out from the website.
6. Clients shall be able to get support.
7. Clients shall be able to create posts.
8. Clients shall be able to book a worker.
9. Clients shall be able to review a worker.
10. Clients shall be able to create Pet profils.
11. Clients shall be able to delete Pet profils.
12. Clients shall be able to report a worker.

Workers:

13. Workers shall be able to sign in to the website.
14. Workers shall be able to log out from the website.
15. Workers shall be able to access notification.
16. Workers shall be able to get support.
17. Workers shall be able to book a client's post.
18. Workers shall be able to report a client.

Administrator:

19. Administrators shall be able to sign in to the website.
20. Administrators shall be able to log out from the website.
21. Administrators shall be able to cancel a booking.
22. Administrators shall be able to ban worker/client users.
23. Administrators shall be able to contact worker/client users.

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. User's data shall be inputted into MYSQL database.
4. Guest user should be able to see the functionality of the website in a demo before signing up.
5. Application UI shall be simple and efficient and easy to use for the user.
6. Application shall be very easy to use and intuitive
7. Application shall be hosted and deployed on Gcloud client server as specified in M0.
8. Application shall be optimized for standard desktop/laptop browsers.
9. All users should be able to contact support.
10. Client should be able to contact Admin or Worker in case of emergency.

Marketing:

11. Site should have a logo to be displayed next to the title.

Security:

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

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 exist
++: 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. The competitors are involving too many filters for their search function which makes the experience confusing and creating more steps for the user. Our goal is to implement an efficient search bar with useful filters for the client to be able to find his/her need. 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 care 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. The main goal of our application is the simplicity of our UI. As we checked other competitors, the moment you enter the site there is too much information on the main page with no organization. As a user, we felt pretty overwhelmed on how to get started on their website. Our goal in our application is to have a clear path at the beginning to show the functionality of our website and at the same time help the user to reach their need. And lastly, we plan to implement a map function for the worker to show listings of available clients in their area, unlike our competitors which only Rover implemented.

7-High-level system architecture and technologies used:

- Sever 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	On Track
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