Buck's Dog Training Scheduler

Will Grimmer and Emma Heiser
Client: Lauraine Wright, Buck's Dog Training of Central NJ

2025-04-05

Contents

1 Introduction

1.1 Client Information

1. Client name: Lauraine Wright
2. Client title: Head Trainer
3. Client email address: lauraine@bucksdogtraining.com
4. Client employer: Lauraine Wright
5. How you know the client: Buck's Dog Training is a small business located by Emma's hometown.

1.2 Overview

This project is a custom website for Buck's Dog Training of Central NJ, designed to improve upon their current Embr AI site to better serve their clients' needs. While their existing website contains information about the company and services for a general audience, scheduling is done manually over phone or email, and it lacks personalized features for current clients. To address this, the new website will implement unique features for employees and customers. Customers can sign up for an account which allows them to view their scheduled appointments digitally and gives them access to client-centric features like their appointment history and reports on their individual dogs' training progress. All of this displayed information will change depending on the logged-in account. This approach will allow for a more streamlined and personal experience with the site.

The main purpose of this project is to provide the clients and employees of Buck's Dog Training a user-friendly website tailored to the individual logged-in client. Appointment scheduling allows employees to book clients' sessions directly on the website, and appointment history and progress offer insights for the client. The website will also include personable features for all website visitors, like information on their services, photo galleries, a mission statement, and pages for each trainer on the website. The proposed features for the new app will help foster a sense of community for current and prospective clients and create a personalized experience to strengthen client relationships.

Employees will have privileges with trainer accounts to create and manage clients' appointments, add to the photo galleries, edit their personal subpage, and edit dogs' progress reports.

1.3 Key Features

Some key features of the project are:

- Appointment scheduling for employees to manage customer appointments, including creating appointments and deleting appointments
- Trainers can manage content like appointments, photo galleries, personal pages, and dogs' reports through admin accounts

- Viewable Appointment History/Future Appointments by Dog Name for each customer's account
- Viewable Dog Training Progress Reports after each appointment
- Automatic Responses to Client Inquiries like appointment creations, appointment reminders, or general questions
- Services information, photo galleries, mission statements, and trainer pages viewable for all website viewers.

1.4 Why this Project is Interesting

This project stands out as a capstone because it has real-world impact for a small business and its clients, allowing them to communicate and operate more efficiently while also keeping a sense of community. Dog training is engaging, and this project will help make that service more accessible and user-friendly.

This project is enticing because it contributes to a service that directly supports people and their pets, and it involves a feature-rich backend for both employees and clients and requires an aesthetic front end to effectively share the company's mission and services.

This project impacts the daily-life of the employees and clients of Buck's Dog Training. The features are useful to make the roles of both easier.

1.5 Areas of CS required

The relevant fields of computer science for this project are: Web development, Database management, and Security.

2 Requirements

2.1 User Roles

2.1.1 Admin

Has all the functionality of the trainers and can CRUD trainer and user accounts as well as directly edit the website.

2.1.2 Trainer

Can make appointments, change their schedule, write trainer reports, and edit their personal page / bio.

2.1.3 User/Customer

Can view trainers and their bios, see their dogs schedule, and contact an administrator to book an appointment.

2.1.4 Site Visitor

Can view the site and see trainers and their bios but cannot view their schedule as they do not have an account. Can make an account. Can contact an administrator to book an appointment

2.2 Functional Requirement User Stories

ID	Story Title	Points	Description
1	Register	2	As a site visitor, I want to create an account so that I can keep track of my scheduled appointments.
2	Log in / Log out	1	As a user, I want to log in and out of my account so that I can access my appointment information.
3	CRUD users	8	As an admin, I want to CRUD users of the site so that I can edit them.
4	CRUD trainer	8	As an admin, I want to CRUD trainers so that I can edit them.
5	Reset Password	2	As a user, I want to reset my password so that I can access my account if I forget login info.
6	CRUD services offered	8	As an admin, I want to CRUD the services we offer so that I can edit them if they change.
7	Set open appointment times	3	As a trainer, I want to set available appointment slots so that I can book appointments.
8	Block out unavailable times	2	As a trainer, I want to block out my unavailable times so that other trainers know when I'm not working.
9	CRUD appointments	8	As an admin, I want to CRUD all scheduled appointments so that I can make changes if necessary.
10	Filter appointments	1	As an admin, I want to filter all appointments by date or by dog so that I can narrow my searches.
11	CRUD my own dogs	8	As a user, I want to CRUD my own dogs to my profile so that I can edit my own info.
12	View trainer info page	2	As a user, I want to view the trainer info page so that I can see who is training my dog.
13	CRUD dog	8	As an admin, I want to CRUD dogs so that I can link multiple dogs to one account.
14	Appointment confirmation	1	As a user, I want to receive appointment confirmations when my appointment is created so that I can confirm it was made.
15	Appointment reminder	1	As a user, I want to receive appointment reminders by email so that I don't forget about my appointment.
16	View past/future scheduled appointments	2	As a user, I want to view my past and future appointments so that I can see everything I scheduled.
17	Search appointments by dog	1	As a user, I want to filter my appointments by dog name so that I can stay organized with multiple dogs.
18	Create training report (OVERRIDDEN BY S27)	2	As a trainer, I want to create training progress reports so that customers can see progress.

19	Delete training report (OVERRIDDEN BY S27)	2	As a trainer, I want to delete training progress reports so that I can fix any mistakes.
20	View training report (OVERRIDDEN BY S27)	2	As a user, I want to view my dog's training report so that I can see their progress.
21	Upload to gallery	2	As an admin, I want to upload photos to a gallery so that site visitors can see them.
22	Remove from gallery	2	As an admin, I want to remove a photo from the gallery so that I can fix any mistakes.
23	Browse gallery	2	As a site visitor, I want to browse the photo gallery so that I can see examples of dogs that were trained.
24	View mission statement	1	As a site visitor, I want to view the mission statement so that I can see the company's values.
25	Contact trainer	1	As a site visitor, I want to contact a trainer so that I can ask any general inquiries.
26	Manage profile	1	As a user, I want to edit my profile information so that I do not need to ask an admin to change it.
27	CRUD training report	8	As an admin, I want to CRUD training reports so that trainers and users can know what happened during an appointment.
T1	Refactor CRUD appointments for fullcalendar	1	As the software devs. we have to change the appointment collection created before the implementation of fullcalendar to align with the fullcalendar package's needs.
T2	Refactor and polish UI	1	As the software devs. we want to refactor the code and update the UI now that most of the logic is implemented.

2.3 Non-Functional Requirements

ID	NFR Title	Category	Description
1	Window size compatibility	Portability	Site should be usable on different window sizes
2	Password security	Security	Create account should encourage strong password
3	Admin and trainer permissions	Security	Only the admin and trainer has access to edit the site
4	Social media links	Usability	Social media links should be visible and clickable
5	Include company logo	Usability	Company logo should be visible throughout site pages
6	Blue color theme	Usability	Site should have a blue color theme to match the logo

3 Iterations

3.1 Iteration 1 Feb. 12

3.1.1 Plan

Planned User Stories	Points
S1: Register	2
S2: Log in/Logout	1
S3: CRUD users	8
S4: CRUD trainer	8
S5: Reset password	2
Total points planned:	21

This iteration is mainly focused on setting the foundation for the project with a stable UI, back-end, and database for user storage. In addition to these stories, this iteration focuses on setting up the project's environment and frameworks.

3.1.2 Results

Planned User Stories	Points	Contributor	Status
S1: Register	2	Will and Emma	Done
S2: Log in/Logout	1	Will and Emma	Done
S3: CRUD users	8	Will	Done
S4: CRUD trainer	8	Will	Unfinished
S5: Reset password	2	Emma	Done
		Total points Completed:	13

Table 1: Table for User Story statuses

For the most part, the team kept on track with the plan with the exception of S4, CRUD trainer. While partially implemented, this story remains uncompleted due to being unsure how to allow an admin to manually create an auth account (trainer) from an admin panel in the project.

S1 and S2 had multiple contributors. The work was split between building the UI and handling user input, and connecting the input to firebase.

Team Member	Points Attempted	Points Completed	Hours Worked	Cycle (Points/Hours)
Emma	5	5	5	1
Will	19	11	10	1.1

Table 2: Individual Team Metrics

This iteration resulted in fast cycles for all team members. As stated above, this iteration focused on setting up the project, which means some hours spent doing technical work were not included in Table 2. In addition to the user stories above, the team spent time setting up the project environment, fixing module dependencies, creating static placeholder pages and page routing, and setting up the database.

3.1.3 Updated Class Diagram

An updated class diagram was treated and is visible in section 4.1.2. While it only updates a relation between a Dog and an Appointment, the class diagram is open to more change in Iteration 2 when the team begins to focus on making the scheduler component.

3.1.4 Testing

Testing File	Statement Coverage (%)	Branch Coverage (%)
SignIn	92	100
LogIn	73.68	50
ResetPassword	100	50
Averages:	88.59	66.66

Table 3: Iteration 1 testing coverage

The testing for this iteration includes the auth-related files. We plan on improving the Jest setup in the future to make it easier to test all features of the project. Currently, the statement coverage meets the standard of 60%. We plan on going back and increasing the coverage in the future.

In addition, the branch coverage for LogIn and ResetPassword are relatively low. This is due to each file only containing two branches: successful form submission and unsuccessful submission. The unsuccessful branches only contain json response error messages.

3.1.5 Retrospective and Reflection

Looking back at this iteration, the team worked well to complete tasks. The team communicated through texting and calls when necessary and always answered within a reasonable time, so the current means of communication works well for the team.

In terms of software, the team learned to not underestimate the time needed to fix bugs and framework related errors. This will become more important as the project gets larger. The team also learned that it's best to push code a little bit at a time, which helps promote working in small increments over the course of a few days.

3.1.6 Planning for Iteration 2

For iteration 2, within the first week, we plan to tie up loose ends from iteration 1, including S4 CRUD Trainer and coverage numbers. At the same time, we plan to set up the foundation for the scheduling feature.

3.2 Iteration 2 Feb. 26

3.2.1 Plan

Planned User Stories	Points
S4: CRUD Trainer	8
S6: CRUD services offered	8
S7: Set open appointment times	3
S8: Block out unavailable times	2
S9: CRUD Appointments	8
Total points planned:	29

The plan for this iteration is to finish S4 CRUD Trainer, make more website general admin privilege and management tools, and begin the scheduling feature.

3.2.2 Results

User Stories	Points	Contributor	Status
S4: CRUD Trainer	8	Will	Done
S6: CRUD services offered	8	Emma	Done
S7: Set open appointment times	3	None	Unfinished
S8: Block out unavailable times	2	None	Unfinished
S9: CRUD Appointments	8	Emma	Done
		Total points Completed:	24

Table 4: Table for User Story statuses

This iteration successfully completed the unfinished stories from iteration one, created more CRUD pages for admin management, and gave more thought to the design of the scheduler.

We completed CRUD Trainer by utilizing Firebase custom claims, which allowed us to create multiple levels of authorization for users to allow for more or less data access depending on the claim. The admin had read and write access to everything including users and the scheduler, trainers have read and write access to the scheduler, and customers have limited read access.

With the completion of the admin CRUD pages, we have most of those features completed, allowing us to fully focus on the scheduler for future iterations.

While we planned on starting the coding for the scheduler, we did not complete that in this iteration. Upon further discussion, we decided that it would be easier in the long run to use a calendar software to help us rather than trying to create a calendar from scratch. Although the implementation is delayed, we explored options for calendar software that can fit the needs of the project. We believe that spending more time on finding the right software now and delaying coding is better than coding now with our initial plan and running into problems later on.

3.2.3 Class Diagram

The class diagram has not been updated.

3.2.4 Testing

Testing File	Statement Coverage (%)	Branch Coverage (%)
ManageServices	81.25	86.41
ManageAppointments	68.75	76.92
Averages:	75	81.67

Table 5: Iteration 2 testing coverage

The testing for this iteration mostly consists of testing api calls from the client side to the server side.

3.2.5 Retrospective and Reflection

Looking back on this iteration, we reached a hurdle that made us decide between leaving stories unfinished and taking more time to plan so that later iterations can be easier versus completing stories now to make progress on the main feature of the project but being less sure about the architecture.

As we finish this iteration, we reflect on how much planning before starting any iteration could have made this go smoother, or if creating the code and seeing it all interact live let us see what we needed. This delay for the scheduler means that we have less features ready than we planned, but since we saw the problem early, we are able to pivot without losing previous work. One of our strengths this iteration was the ability of the group to quickly adapt and try to find a solution to this problem.

3.2.6 Plan for Iteration 3

Although the scheduler was delayed, iteration 3's main focus was on the scheduler. Thus, we will end up spending more time setting up the scheduler as planned in this iteration on top of the user stories related to the scheduler that were previously planned for iteration 3.

3.3 Iteration 3 March 19

3.3.1 Plan

The plan for this iteration is to begin actualizing the scheduler feature and finishing up CRUDs that are required to create appointments on the client side. The plan begins with initial research to find the right software to build the scheduler feature. Then it transitions into coding the rest of the information needed to create appointments, the calendar, and displaying appointments. Then we need to find a way to send email confirmations to customers when their appointments are created, and we need to install a software to schedule appointment email reminders.

Planned User Stories	Points
S7: Set open appointment times	3
S8: Block out unavailable times	2
S10: Filter appointments	1
S11: CRUD my own dog	1
S12: View trainer info page	2
S13: CRUD Dog	8
S14: Appointment confirmation	1
S15: Appointment reminder	1
Total points planned:	19

3.3.2 Results

User Stories	Points	Contributor	Status
S7: Set open appointment times	3	Will	Done
S8: Block out unavailable times	2	Will	Done
S16: View past/future scheduled appointments	2	Will	Done
S10: Filter appointments	1	None	Unfinished
S11: CRUD my own dog	1	Emma	Done
S12: View trainer info page	2	Emma	Done
S13: CRUD Dog	8	Emma	Done
S14: Appointment confirmation	1	Emma	Done
S15: Appointment reminder	1	Emma	Done
S26: Manage Profile	1	Emma	Done
T1: Refactor CRUD appointments for fullcalendar	1	Emma	Done
		Total points completed:	22

We decided to use the fullcalendar package to implement the appointment calendar UI. We chose this package because of colleague recommendation and the capabilities it has so that customers can view their appointments on one calendar and then trainers can manage appointments on another calendar.

Currently, the fullcalendar allows multiple calendar views of appointments, like the entire month, the week, and the day. The weekly and daily view provide specific hours of the appointments.

We also use nodemailer to send emails and node-cron to schedule the event that searches for appointments that start on the current day and sends a reminder email. The event is scheduled to run at 8am every day.

During this sprint, we added the Manage Profile user story during the process of completing S11, realizing that if a user can manage the dogs they own, they should be able to manage their email and other personal information.

We also added a technical task Refactor CRUD appointments for fullcalendar. Now that we have concrete knowledge of how we are handling and displaying appointments, we had to add timestamp fields to each appointment for when the appointment starts and when the appointment ends. The appointment date was originally stored in a string, but using timestamps will allow fullcalendar to display the appointments on a calendar.

Due to the nature of the stories, we completed S16 while in the process of doing S7 and S8 even though S16 was not planned for this iteration.

3.3.3 Testing

Testing File	Statement Coverage (%)	Branch Coverage (%)
Navbar	100	88.88
AdminDashboard	100	100
Login	100	100
ManageAppointments	89.52	82.35
ManageProfile	96.77	85.29
ManageServices	100	92.3
ManageTrainers	95.83	100
ManageUsers	90	76.78
OurTeam	100	100
ResetPassword	100	100
Services	100	100
Signin	100	100
Averages:	97.68	93.8

Table 6: Iteration 3 testing coverage

The testing for this iteration tests the files made during this iteration and the files from previous iterations to increase their coverage.

A problem where the testing framework could not find the react-router-dom module despite the module being installed was recently fixed by downgrading the module from version 7 to version 6. This fix allowed for more extensive texting on previous files that included routing.

Uncovered statements and branches are lines that address error handling.

3.3.4 Retrospective and Reflection

Looking back on Iteration 3, the team did well in setting up the individual components to make the integration of said components into the appointments and scheduler go smoothly. Because they were created in such a way that we could easily build off of the code from previous iterations, we ran into very few problems when combining these different components into the new stories from this iteration.

During this iteration, the team faced some procrastination. To reduce this problem in the next iteration, we will set milestones and dates for the milestones early on.

3.3.5 Plan for Iteration 4

For Iteration 4, we plan on refining the scheduler by implementing appointment filtering so that both customers and trainers can find the right appointment quickly. We also plan on making the training report feature, which trainers will be able to add to appointments once it is over. Finally, since the scheduler feature is mostly finished, we will take time to organize and refactor code as needed and polish the UI to comply with our NFRs and be more visually appealing.

3.4 Iteration 4 April 2

3.4.1 Plan

Planned User Stories	Points
S10: Filter appointments	1
S17: Search appointments by dog	1
S18: Create training report	2
S19: Delete training report	2
S20: View training report	2
T2: Refactor as needed and polish UI	1
Total points planned:	9

The plan for iteration 4 involves filtering the appointments, specifically by date so that the appointments can be sorted by past appointments, upcoming appointments, and current appointments, and by dog so that only one specific dog's appointments can be viewed at a time.

Additionally, we plan to add the training report feature so that trainers can keep track of what happens during an appointment and the dog's owner can have record of the dog's progress.

We also added the second technical task to refactor code and improve UI. We plan to complete this task last if time allows.

3.4.2 Results

The results of this iteration include the completion of all planned story tasks and the completion of tasks we planned for the future, thus making us ahead of schedule.

Appointments are now filtered by 'Today's Appointments', 'Upcoming Appointments', and 'Past Appointments'. 'Today's Appointments' includes the appointments with a start date of the current day, and it will move to the 'Past Appointments' tab once the end time of the appointment passes.

Appointments can also be filtered by dog's name. One dog can be filtered at a time. Trainers and admins will be able to filter the appointments of all dogs in the database while users can only filter by the dogs that they own.

Notably, the table includes a new notion of 'OVERRIDDEN'. This note is applied to all user stories relating to the training reports. During the initial planning stage, the training

User Stories	Points	Contributor	Status
S10: Filter appointments	1	Emma	Done
S17: Search appointments by dog	1	Emma	Done
S27: CRUD training report	8	Emma	Done
T2: Refactor as needed and polish UI	1	Emma	Unfinished
S18: Create training report			(OVERRIDDEN)
S19: Delete training report			(OVERRIDDEN)
S20: View training report			(OVERRIDDEN)
S21: Upload to gallery	2	Will	Done
S22: Remove to gallery	2	Will	Done
		Total points completed:	14

reports did not include an edit feature because they are meant to describe what happened in an appointment that already occurred. A trainer would have no reason to go back to an appointment that ended days or weeks ago and change the training report.

During the coding phase of this iteration, we took multi-day appointments (like overnight boarding) into consideration. We wanted trainers to be able to edit the same training report on a daily basis rather than only being able to make a report after the entire multi-day appointment ended. Because of this, we added the edit appointment feature for all in-progress and past appointments and consolidated S18, S19, and S20 into S27: CRUD training reports.

Although not initially planned, we were also able to implement some of the photo gallery feature. Now the admin can upload and delete pictures that they want to be publically displayed on the website's photo gallery.

We did not complete the T2 due to time constraints.

3.4.3 Testing

Testing File	Statement Coverage (%)	Branch Coverage (%)
AllCalendar	96.15	100
ManageAppointments	92.03	80.46
SelfCalendar	96.66	100
MyAppointments	93.18	83.52
Averages:	94.50	90.99

Table 7: Iteration 4 testing coverage

The testing for this iteration includes both the user's view and the admin's view of their respective calendars and appointment lists. The lower branch coverage on ManageAppointments and MyAppointments is due to the amount of filtering the file does in regards to appointments, which is hard to completely cover with a mocked database for testing.

3.4.4 Retrospective and Reflection

Looking back at this iteration, this is the first iteration that will not carry backlog (not including technical tasks) into the following iteration. We are pleased with this progress and the work the team put into the project to make it happen.

This progress and the progress made on the entire project so far was partly due to planning more points in the first few iterations, even if we thought the amount of planned points was a reach. We begin to see this pay off during this iteration where the amount of planned points are significantly lower.

3.4.5 Plan for Iteration 5

Iteration 5 focuses on cleanup and UI. Because the main logic for the project is done, we can focus more on the appearance of the website and the user interaction with the current features.

3.5 Iteration 5 April 16

3.5.1 Plan

Planned User Stories	Points
S23: Browse gallery	2
S24: View mission statement	1
S25: Contact trainer	1
T2: Refactor as needed and polish UI	1
Total points planned:	5

3.5.2 Activities

3.5.3 Retrospective

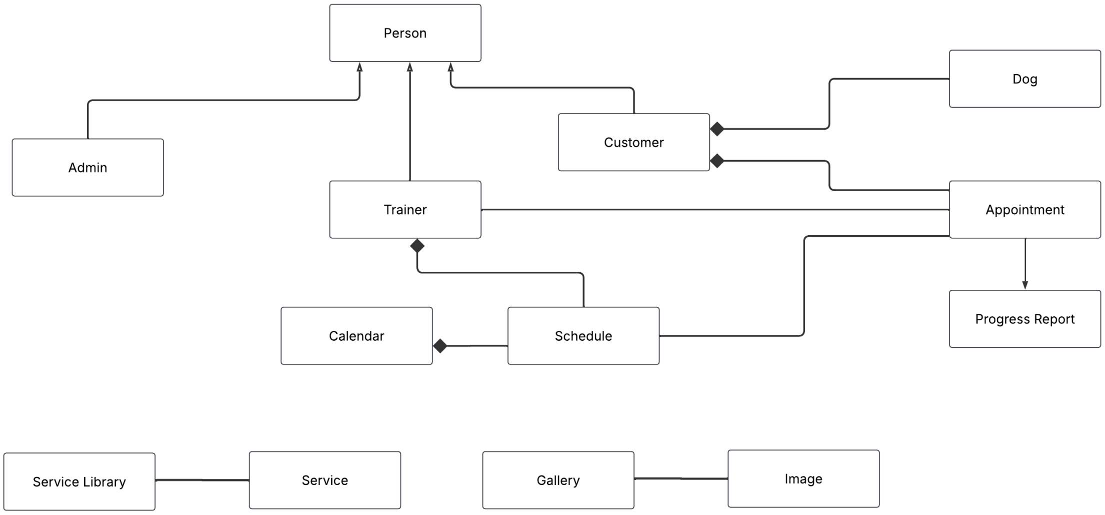
4 Final System Architecture and Design

4.1 Architecture

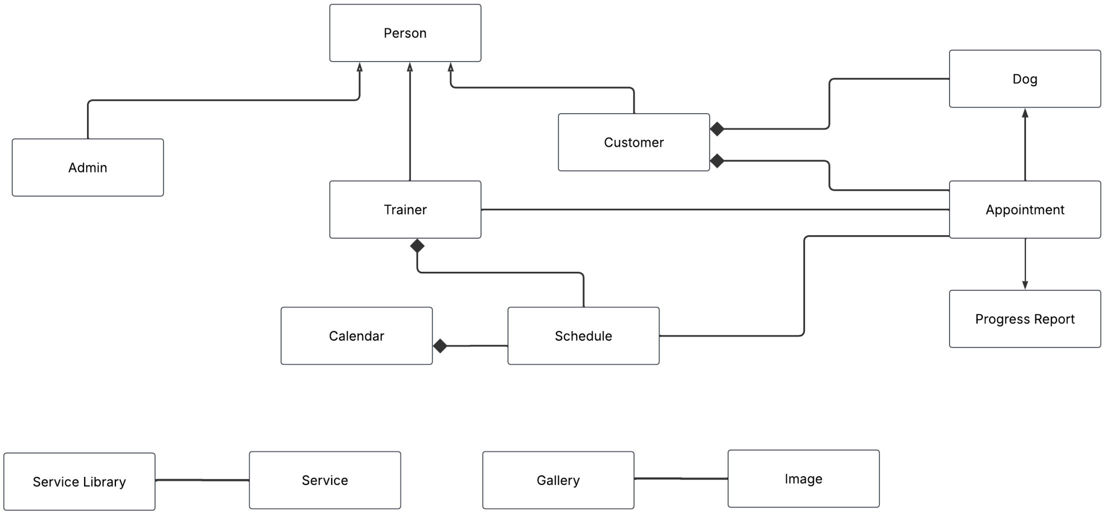
For this project, we use layered architecture. We chose this particular architecture due to prior experience with using it.

The layers we include are the client-sided layer, the business logic and the database layer. The client layer contains all the HTML and client-oriented visuals and buttons that facilitate the business-logic. the business logic layer contains the server-sided Node.js features and routes to facilitate communication with the database. The database layer is firebase for this project and is the persistent storage for the project's information.

4.1.1 Class Diagram



4.1.2 Updated Class Diagram Iteration 1



4.2 Technology

The main programming language we use for this project is Javascript, chosen because of prior experience with this programming language for similar website development projects.

The main frameworks and libraries we use are:

- NodeJS
- React
- Jest
- Javascript Sessions

We use NodeJS for the website's environment, and the React library to develop the front-end of the website. For testing, we opt for the Jest testing framework since it is one of the most used testing frameworks for Javascript software, and it is compatible with NodeJS and React. Javascript Sessions is used to help maintain persistent user data with each log in.

For persistent data storage, we use Firebase for a NoSQL database. We use this because of previous experience with Firebase, and we expect the database's capabilities to be sufficient for the needs of the website traffic.

4.3 Data

Looking at the use of Firebase in-depth, we include numerous collections and subcollections:

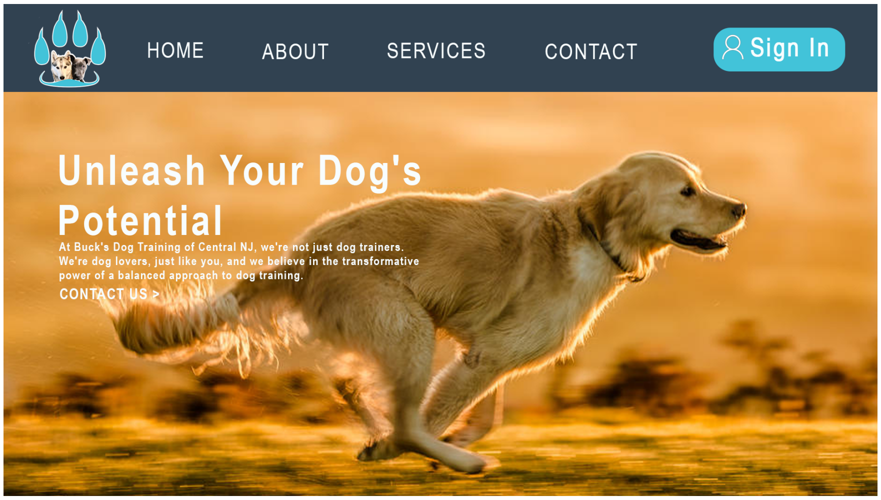
- Person Collection: a person contains name, usernameand email
  - Admin Subcollection: an admin contains admin permission (true or false).
  - Trainer Subcollection: a trainer contains biography and schedule.
  - Customer Subcollection: a customer contains appointments and dogs (array of Dog Collection)
- Dog Collection: a dog contains name, age, ownerID (user ID).
- Appointment Collection: an appointment contains trainer, dog, date, location, purpose, dropoff time, pickup time, balance due.
- Schedule Collection is a collection of appointments.
- Calendar is a collection of schedules.
- Gallery Collection is a collection of images.
- Service Library Collection is a collection of services.
  - Service: contains name, description and price.

4.4 Coding Standards

To ensure consistency and organization throughout the project, we implement these coding standards:

- We will utilize camel case for this project's naming convention, including collections and attributes.
- We will only commit working code to the repo.
- We will commit to the repo early and often, committing work as soon as the changes are made and working.
- We will push small amounts of working code. This standard will keep the versioning of the repo organized and easily debuggable.
- For testing, we will only push with at least 60% coverage. We will aim to finish with at least 80% coverage.
- We will utilize trunk-based coding, pulling and pushing from the main branch. We will not utilize individual branches.

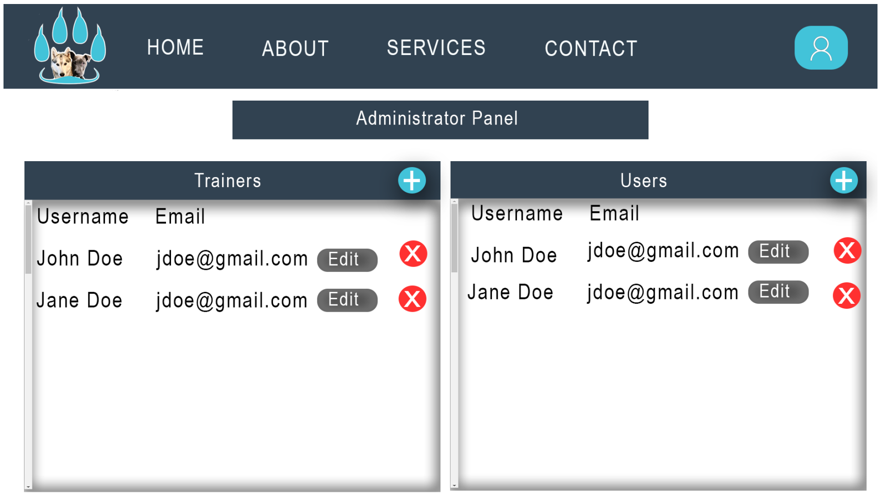
4.5 UI



Unleash Your Dog's Potential

At Buck's Dog Training of Central NJ, we're not just dog trainers. We're dog lovers, just like you, and we believe in the transformative power of a balanced approach to dog training.

Figure 1: Homepage of the website, first page a user sees



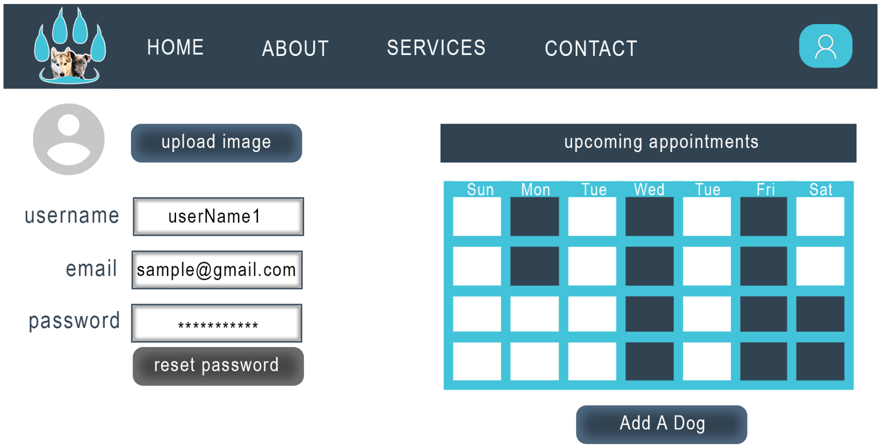
Trainers

Username	Email		
John Doe	jdoe@gmail.com	Edit	☒
Jane Doe	jdoe@gmail.com	Edit	☒

Users

Username	Email		
John Doe	jdoe@gmail.com	Edit	☒
Jane Doe	jdoe@gmail.com	Edit	☒

Figure 2: Admin panel to manage users and trainers



username: userName1

email: sample@gmail.com

password: ***********

Sun	Mon	Tue	Wed	Tue	Fri	Sat
□	■	□	■	□	■	□
□	■	□	■	□	■	□
□	□	□	■	□	■	■
□	□	□	■	□	■	■

Figure 3: User profile to see personal account information, appointments, and dogs

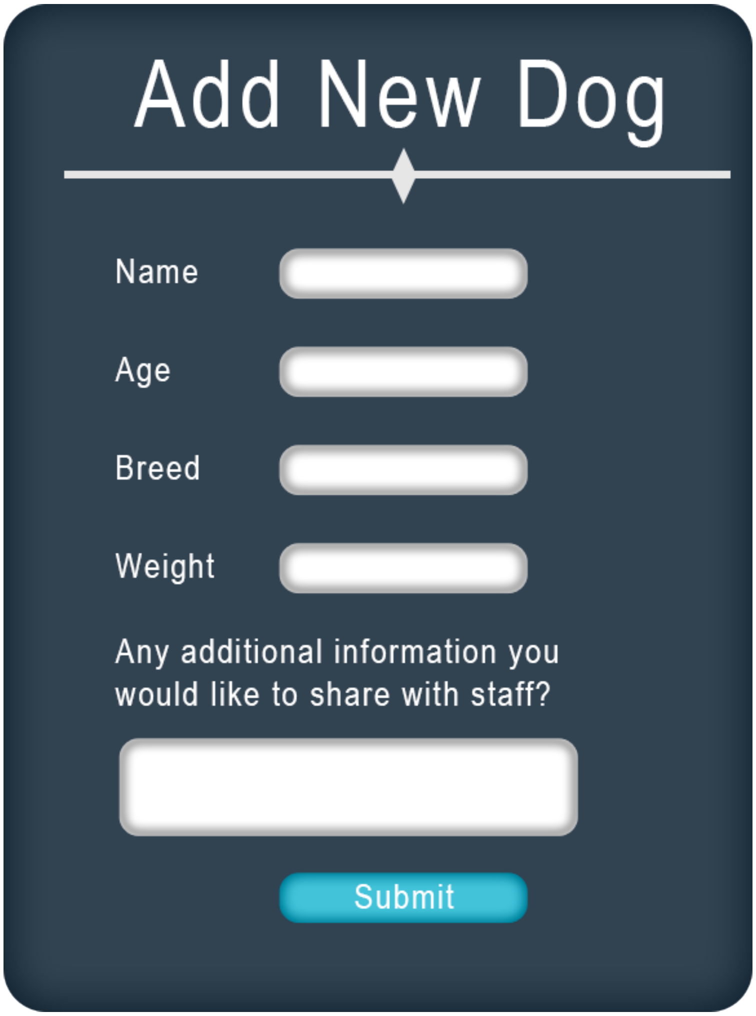
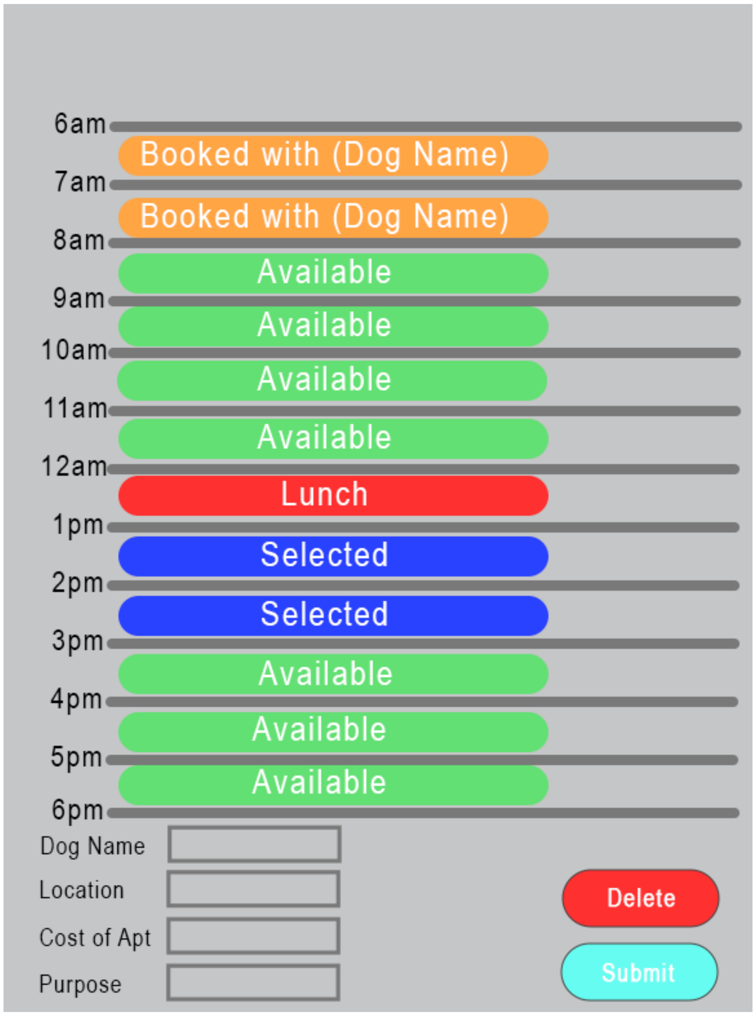

Add New Dog
Name
Age
Breed
Weight
Any additional information you would like to share with staff?
Submit

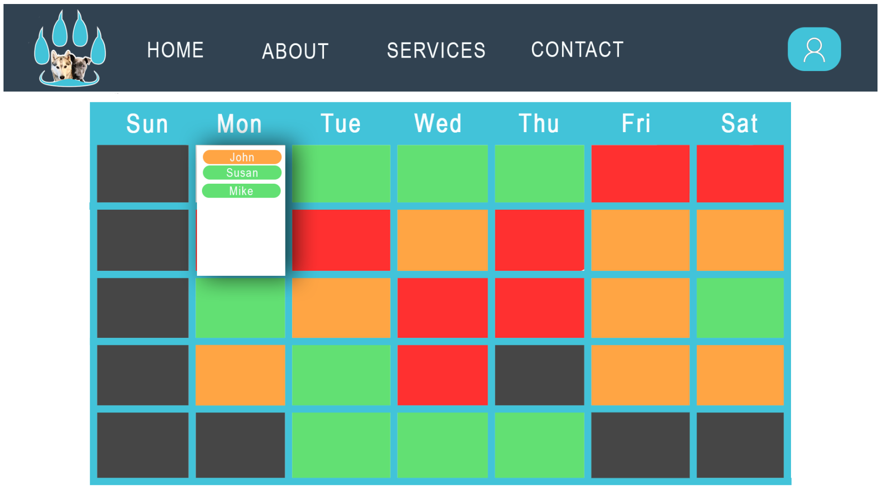
Figure 4: UI for a customer to add a dog to their profile



6am
Booked with (Dog Name)
7am
Booked with (Dog Name)
8am
Available
9am
Available
10am
Available
11am
Available
12am
Lunch
1pm
Selected
2pm
Selected
3pm
Available
4pm
Available
5pm
Available
6pm

Dog Name
Location
Cost of Apt
Purpose

Figure 5: UI for an individual trainer's schedule



Sun	Mon	Tue	Wed	Thu	Fri	Sat
⬛	John Susan Mike	🟩	🟩	🟩	🟥	🟥
⬛		🟥	🟧	🟥	🟧	🟧
⬛	🟩	🟧	🟥	🟥	🟧	🟩
⬛	🟧	🟩	🟥	⬛	🟧	🟧
⬛	⬛	🟩	🟩	🟩	⬛	⬛

Figure 6: UI for the general schedule and calendar