Bark Park Online

CMSC 495-7380 Group 1

Anas Abdulrazzaq
Ryan Austin
Allison McDonald
Raymond “Moe” Rantala
Christina Reiss

Final Report

Date: 12/15/2020

Table of Contents

Revision History

Name	Date	Reason for Changes	Version
Anas Abdulrazzaq	12/14/2020	Document Draft Creation	1
Allison McDonald	12/15/2020	Document Updates and Additions for Final	2

1. Overview

1.1 Project Purpose

The project is a website for a dog park where users can update a calendar to inform others when they plan to visit the park. Users have the option to create an account or to remain as a guest. Users with accounts can add, view, and edit their visits. Guests are only allowed to view visits.

1.2 Individual Contributions

- Anas Abdulrazzaq – Back-end Development and Reporting
- Ryan Austin – Technical Lead, Back-end Development, Front-end Development, and Database Engineer
- Allison McDonald – Reporting and Back-end Development
- Raymond “Moe” Rantala – Front-end Development
- Christina Reiss – Back-end Development and Test Engineer

2. Project Plan

2.1 Entity Definitions

There are four entities for this project. The entities are Dog, Owner, Park, and Visit. Each entity has various associated attributes.

The Dog entity is defined as a domesticated canine possessed by an Owner with a Bark Park account. Dog attributes are Name, Breed, Date of Birth, and Gender. Name is defined as the name of the Dog. Breed is defined as the type based on physical traits, behavioral traits, and personality traits. Date of Birth is defined as when the Dog was born. Gender is defined as the sex of the Dog.

The Owner entity is defined as a person who owns a dog(s) and creates a Bark Park account. Owner attributes are Name, Email Address, Username, and Password. Name is defined as the first and last name of the Owner. Email Address is defined as the contact information of the Owner. Username is defined as a unique account name the owner provides that is associated with a Bark Park account and is used in the login process. Password is defined as a string of characters and numbers the owner provides that is associated with a Username and is used in the login process. The Password has a minimum length of eight and a maximum length of sixty.

The Park entity is defined as a controlled environment dedicated for the purposes of off-leash exercise and play for dogs under the supervision of their Owner. Park attributes are Name and Location (Address). Name is defined as the name of the Park. Location (Address) is defined as the Park’s street number, street name, city name, state name, and zip code.

The Visit entity is defined as an Owner scheduled appointment intended to show the Owner’s intention to be at the Park with their Dog(s). Visit attributes are Date, Time, Owner, Dog, Park. Date is defined as the month, day, and year of the appointment. Time is defined as the 12-hour clock notation of the start time of the appointment. Owner is defined as the name of the scheduler/attendee of the appointment. Dog is defined as the name, breed, and age of the dog(s) attending the appointment. Park is defined as the location of the appointment.

3. Requirements Specification

3.1 Scenarios

There are three scenarios for this project. The scenarios are General, Dog Management, and Visit Management. Each scenario has various associated actions.

The General scenario actions are Login, Logout, and Add Account.

- Login – The system shall provide the ability for a user with an existing Bark Park account to gain access by entering the Username and Password associated with their Bark Park account. When both Username and Password are verified by the system as matching a Bark Park account, access is granted. When one or both Username and Password does not match a Bark Park account, access is denied.
- Logout – The system shall provide the ability for a currently logged in Bark Park account to be exited.
- Add Account – The system shall provide the ability for a user to create a Bark Park account. The creation of a Bark Park account shall require the user to register as an Owner with at least one dog. To complete registration, the user must complete the Owner and Dog profile. The Owner profile requires their Name, Email Address, creating a Username, and creating a Password. The Dog profile requires their dog's Name, Breed, Date of Birth, and Gender.

The Dog Management scenario actions are Add Dog, Edit Dog, View Dog, Delete Dog (Soft Delete).

- Add Dog – The system shall provide the ability for a logged in Owner to create Dog profiles to their account. The Dog profile requires their dog's Name, Breed, Date of Birth, and Gender.
- Edit Dog – The system shall provide the ability for a logged in Owner to edit the dog's Breed, Date of Birth, and Gender in an existing Dog profile associated with their account.
- View Dog – The system shall provide the ability for a logged in Owner to view their Dog profile. The view shall include the dog's Name, Breed, Date of Birth, and Gender.
- Delete Dog (Soft Delete) – The system shall provide the ability for a logged in Owner to remove a Dog from their account. When an Owner deletes a Dog, it is marked as inactive and cannot be reinstated.

The Visit Management scenario actions are Add Visit, Edit Visit, View Visit, and Delete Visit (Soft Delete).

- Add Visit – The system shall provide the ability for a logged in Owner to create a Visit on their account. The creation of a Visit shall require the Owner to enter a Date, Time, and select the Dog profile(s) that will attend.
- Edit Visit – The system shall provide the ability for a logged in Owner to edit the Date, Time, and Dogs in an existing Visit on their account.
- View Visit – The system shall provide the ability for a logged in Owner to view only future Visits on their account or to view all future Visits on all Owner accounts. The view shall include the Visit Date, Visit Time, Visit Park, Owner name, and Dog Name. The view shall be sorted by Visit Date.
- Delete Visit (Soft Delete) – The system shall provide the ability for a logged in Owner to remove a Visit from their account. When an Owner deletes a Visit, it is marked as inactive and cannot be reinstated.

3.2 Modules

There are three modules. The modules are Login/Logout/Signup, Account Management, and Visit Management.

The Login/Logout/Signup module is the user view where and existing Owner can Login or Logout and where a new user can Add Account.

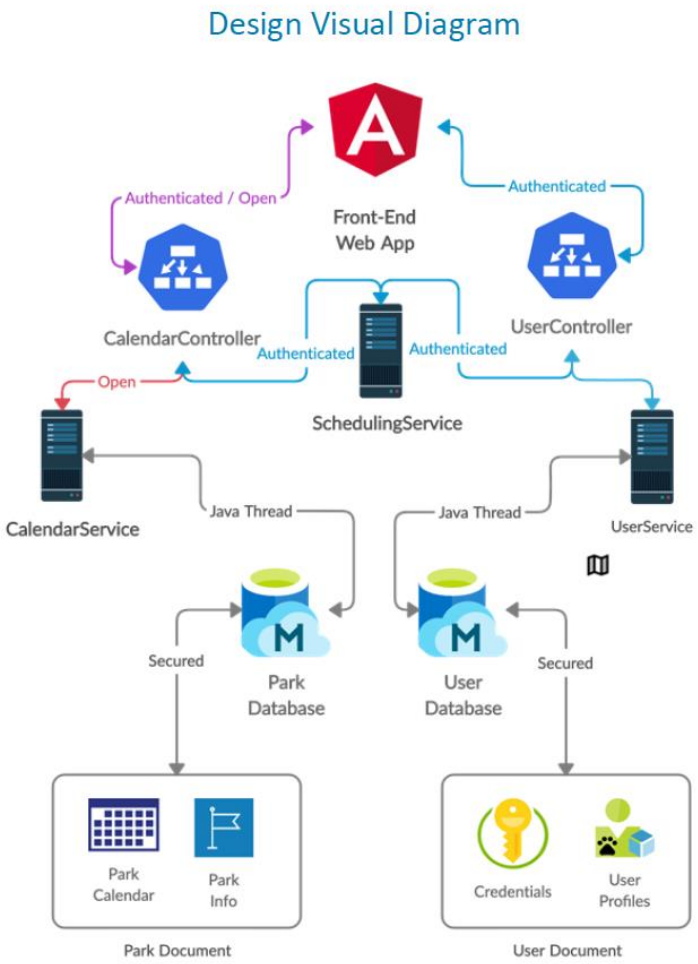
The Account Management module is the logged in Owner view where Owners can Update their Profile, Add Dog, Edit Dog, View Dog, and Delete Dog.

The Visit Management module is the logged in Owner view where Owners can Add Visit, Edit Visit, View Visit, and Delete Visit.

4. System Specification

4.1 System Architecture

Diagram of the System Architecture



The intent of the above diagram is to show the back and forth of data between the Web Application, Front-end Framework, Back-end Framework, Servers, and Databases for this System.

The system shall run on all operating systems using FireFox, Chrome, or Edge web browsers.

4.2 Technical Stack

The system shall use the web server framework Spring Boot v2.3.3, web framework and libraries from Angular v10.0.7, programming language JavaSE v1.8, building tool Gradle v6.5.1, and database MongoDB v4.4.1.

5. User's Guide

5.1 Login

1. Navigate to the Bark Park Online website
2. Click Login
3. Enter your Bark Park Account (Note: To create a Bark Park Account click Sign Up.)
    a. Username
    b. Password
4. Click 'Login'

5.2 Logout

1. When logged in click the Drop-Down Arrow by your name
2. Select 'Logout'

5.3 Add Account

1. Click 'Sign Up' to create a Bark Park Account
2. Fill out Your Credentials
    a. Username – Enter six to twenty characters
    b. Email Address – Enter email address
    c. Password – Enter a minimum of eight characters
3. Fill out Your Profile
    a. Name – Enter alphabetic characters ranging from a to z
4. Fill out Add Dogs
    a. Name – Enter alphabetic characters ranging from a to z
    b. Breed – Enter alphabetic characters ranging from a to z
    c. Date of Birth – Select date using the calendar icon
    d. Gender – Select from the Drop-Down menu
5. Click Done

5.4 Add Dog

1. Login to Bark Park Account
2. Click the Drop-Down Arrow by your name
3. Select 'Edit Profile'
4. Click the '+1' icon shown in the Dogs section
5. Fill out the Dog profile

a. Name – Enter alphabetic characters ranging from a to z
b. Breed – Enter alphabetic characters ranging from a to z
c. Date of Birth – Select date using the calendar icon
d. Gender – Select from the Drop-Down menu

6. Click ‘Save Dog’ when all fields are filled out

5.5 Edit Dog

1. Login to Bark Park Account
2. Click the Drop-Down Arrow by your name
3. Select ‘Edit Profile’
4. Select the Dog to edit from the Drop-Down menu
5. Update the Dog profile
    a. Dog Breed – Enter alphabetic characters ranging from a to z
    b. Date of Birth – Select date using the calendar icon
    c. Gender – Select from the Drop-Down menu
6. Click ‘Update Dog’

5.6 View Dog

1. Login to Bark Park Account
2. Click the Drop-Down Arrow by your name
3. Select ‘Edit Profile’
4. Owner’s active Dog Profile(s) are shown as default

5.7 Delete Dog

Important: Once a Dog profile is deleted, it cannot be reinstated.

1. Login in Bark Park Account
2. Click the Drop-Down Arrow by your name
3. Select ‘Edit Profile’
4. Select the Dog profile to delete from the Drop-Down menu
5. Click ‘Remove Dog’

5.8 Add Visit

1. Login to Bark Park Account
2. Click ‘Calendars’
3. Select a ‘My Favorite Park’ from the Drop-Down menu
4. Click ‘New Visit’
5. Fill out the Visit form
    a. Date – Enter the month, day, and year
    b. Time – Enter the start time
    c. Select the Dog profile(s) that will attend
6. Click ‘Add Visit’ when all fields are filled out

5.9 Edit Visit

Note: Only the Date can be edited.

1. Login to Bark Park Account
2. Click ‘Calendars’ or ‘My Visits’
3. Select the Visit to edit
4. Click ‘Edit’
5. Update the Visit form
    a. Date – Enter the month, day, and year
    b. Dogs – Select the Dog profile(s) that will attend
6. Click ‘Save’ to update the Visit

5.10 View Visit

1. Login to Bark Park Account
2. Click ‘Calendars’
3. Select a ‘My Favorite Park’ from the Drop-Down menu
4. All Owner’s future visits are shown as default
5. Select ‘My Visits’ to view your future visits

5.11 Delete Visit

Important: Once the Visit is deleted, it cannot be reinstated.

1. Login to Bark Park Account
2. Click ‘My Visits’
3. Select the Visit to delete
4. Click the ‘Red X’ icon

6. Test Plan and Results

6.1 End-to-End Testing Plan

Test #	Test Scenario	Input	Expected Output	Pass
1	Login	Valid Account Username Valid Account Password	Pass: Successful Login	**Yes**
2		Valid Account Username Invalid Account Password	Exception	**Yes**
3		Invalid Account Username Valid Account Password	Exception	**Yes**
4		Invalid Account Username Invalid Account Username	Exception	**Yes**
5	Logout	Logged in Account Click 'Logout'	Pass: Successful Logout	**Yes**

Test #	Test Scenario	Input	Expected Output	Pass
6		Not Logged in Click 'Logout'	Exception	Yes
7	Add Account	Click 'Sign Up' Fill out Owner Profile: Valid Owner Name entered Valid Owner Username created Valid Owner Password created Fill out Dog Profile: Valid Dog Name entered Valid Dog Breed entered Valid Dog Age entered Click 'Create Account'	Pass: Bark Park Account created	Yes
8		Fill out Owner Profile: Invalid Owner Name entered Click 'Create Account'	Exception	Yes
9		Fill out Owner Profile: Invalid Owner Username created Click 'Create Account'	Exception	Yes
10		Fill out Owner Profile: Invalid Owner Password created Click 'Create Account'	Exception	Yes
11		Fill out Dog Profile: Invalid Dog Name entered Click 'Create Account'	Exception	Yes
12		Fill out Dog Profile: Invalid Dog Breed entered Click 'Create Account'	Exception	Yes
13		Fill out Dog Profile: Invalid Dog Age entered Click 'Create Account'	Exception	Yes
14	Add Dog	Fill out Dog Profile: Valid Dog Name entered Valid Dog Breed entered Valid Dog Age entered Click 'Save'	Pass: New Dog profile created and added to Owner's account	Yes
15		Fill out Dog Profile: Invalid Dog Name entered Click 'Save'	Exception	Yes
16		Fill out Dog Profile: Invalid Dog Breed entered Click 'Save'	Exception	Yes
17		Fill out Dog Profile: Invalid Dog Age entered Click 'Save'	Exception	Yes

Test #	Test Scenario	Input	Expected Output	Pass
18	Edit Dog	Login Click 'Dog Management' Select Dog to edit Click 'Edit' Valid Dog Breed entered Click 'Save'	Pass: Dog's profile breed is updated	Yes
19		Invalid Dog Breed entered Click 'Save'	Exception	Yes
20	View Dog	Login Click 'Dog Management'	Pass: Owner's all active Dog Profile(s) are shown	Yes
21	Delete Dog	Login Click 'Dog Management' Select Dog profile to delete Click 'Delete' Click 'Save'	Pass: Dog profile no longer shows in Owner's Dog Management	Yes
22	Add Visit	Login Click 'Visit Management' Click 'Add Visit' Fill out Visit form: Valid Date entered Valid Time entered Valid Duration entered Valid Dog Profile(s) selected Click 'Save'	Pass: New Visit is created	Yes
23		Fill out Visit form: Invalid Date entered Click 'Save'	Exception	Yes
24		Fill out Visit form: Invalid Time entered Click 'Save'	Exception	Yes
25		Fill out Visit form: No Dog Profile(s) selected Click 'Save'	Exception	Yes
26	Edit Visit	Login Click 'Visit Management' Select the Visit to edit Click 'Edit' Valid Visit Date entered Click 'Save'	Pass: Visit Date is updated	Yes
27		Invalid Visit Date entered Click 'Save'	Exception	Yes
28	View Visit	Login Click 'Visit Management'	Pass: Owner's all future Visits are shown	Yes

Test #	Test Scenario	Input	Expected Output	Pass
29	Delete Visit	Login Click 'Visit Management' Select Visit to delete Click 'Delete' Click 'Save'	Pass: Visit no longer shows in Owner's Visit Management	Yes

6.2 End-to-End Test Summary

Test Summary

38	0	0	6m29.04s
tests	failures	ignored	duration

100% successful

Packages | Classes

Package	Tests	Failures	Ignored	Duration	Success rate
default-package	29	0	0	6m24.68s	100%
org.cmsc495.bpo.controllers	9	0	0	4.352s	100%
org.cmsc495.bpo.cucumber.tests	0	0	0	-	-

6.3 Unit Test Summary

Test Summary

86	0	0	7.358s
tests	failures	ignored	duration

100% successful

Packages | Classes

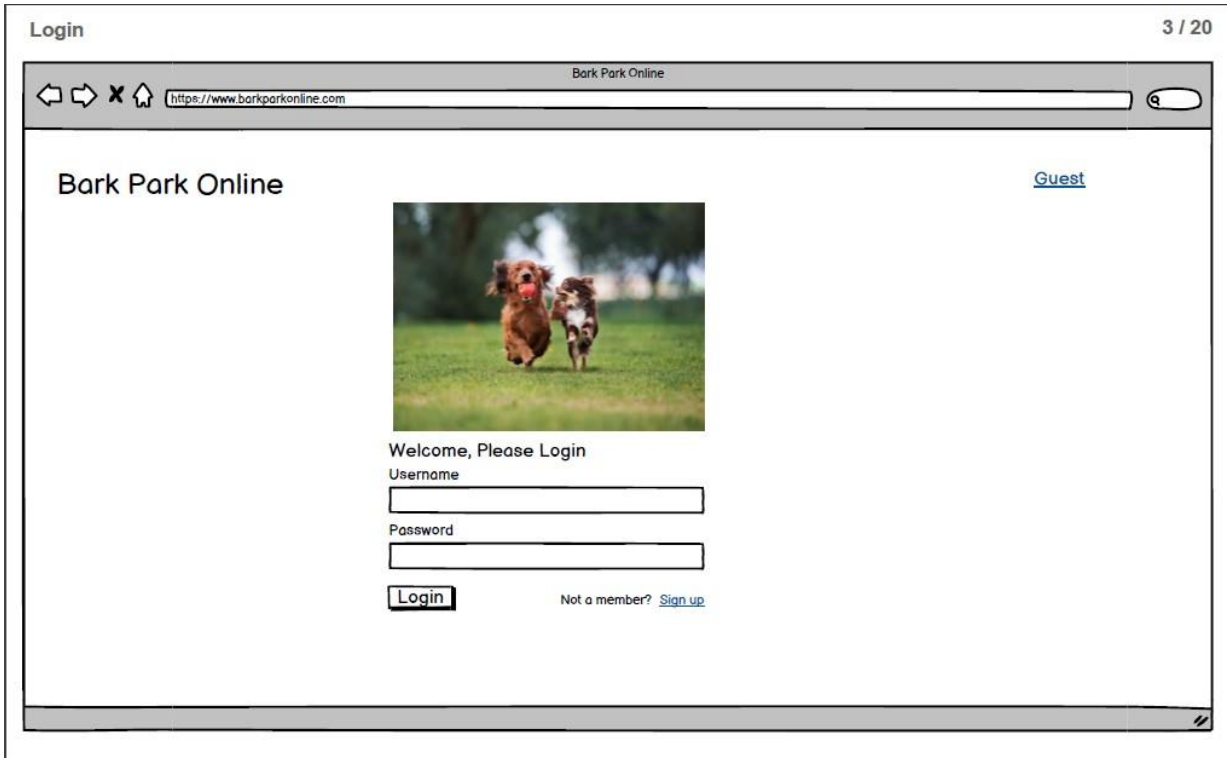
Package	Tests	Failures	Ignored	Duration	Success rate
org.cmsc495.bpo.controllers	16	0	0	3.314s	100%
org.cmsc495.bpo.controllers.response	7	0	0	1.059s	100%
org.cmsc495.bpo.dao	27	0	0	0.042s	100%
org.cmsc495.bpo.repositories	12	0	0	1.244s	100%
org.cmsc495.bpo.services	24	0	0	1.699s	100%

6.4 Unit Test Coverage

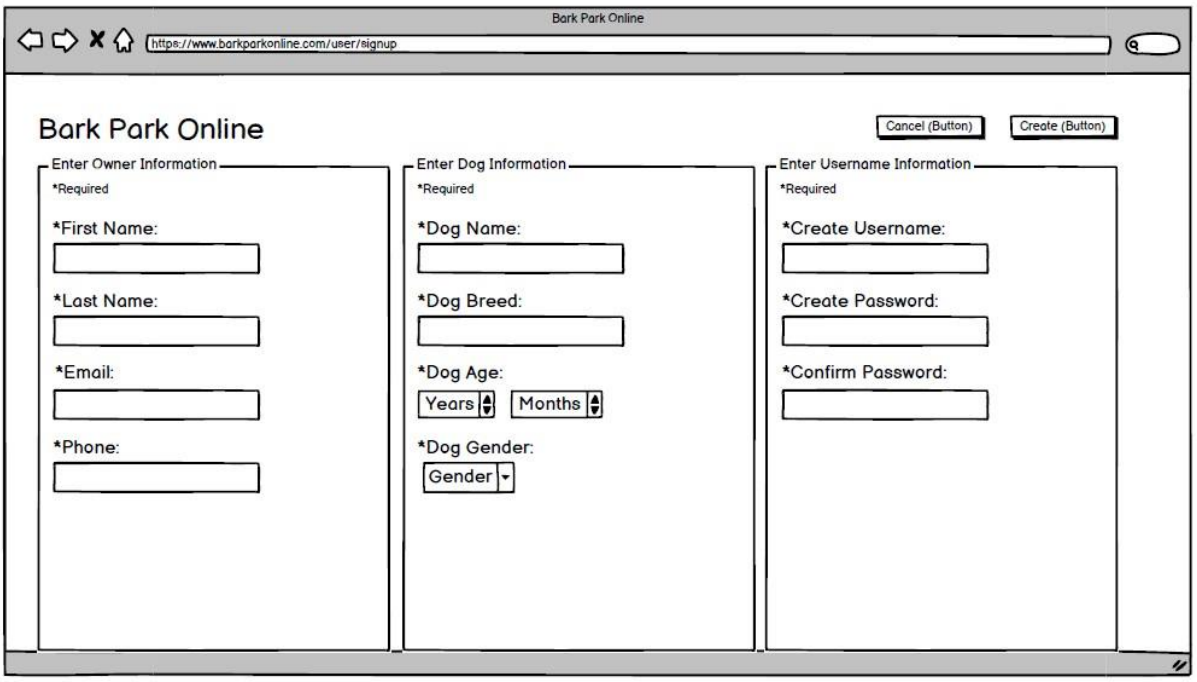
Element	Missed Instructions	Cov.	Missed Branches	Cov.	Missed	Cxty	Missed	Lines	Missed	Methods	Missed	Classes
org.cmsc495.bpo.services		63%		54%	41	89	104	283	13	43	2	8
org.cmsc495.bpo.dao		60%		43%	69	186	113	336	53	162	2	10
org.cmsc495.bpo.controllers		56%		45%	16	39	69	161	9	29	1	4
org.cmsc495.bpo.configs		0%		0%	23	23	66	66	20	20	8	8
org.cmsc495.bpo.repositories		60%		67%	14	38	40	96	7	24	0	2
org.cmsc495.bpo.security		0%		0%	10	10	18	18	7	7	1	1
org.cmsc495.bpo.controllers.response		52%		n/a	13	28	27	54	13	28	2	5
org.cmsc495.bpo.dao.interfaces		55%		0%	3	4	5	7	1	2	1	2
org.cmsc495.bpo.exceptions		66%		n/a	2	6	4	12	2	6	2	6
org.cmsc495.bpo		0%		n/a	2	2	3	3	2	2	1	1
org.cmsc495.bpo.util		89%		100%	1	3	1	5	1	2	0	1
Total	1,885 of 4,185	54%	105 of 206	49%	194	428	450	1,041	128	325	20	48

7. Design and Alternate Designs

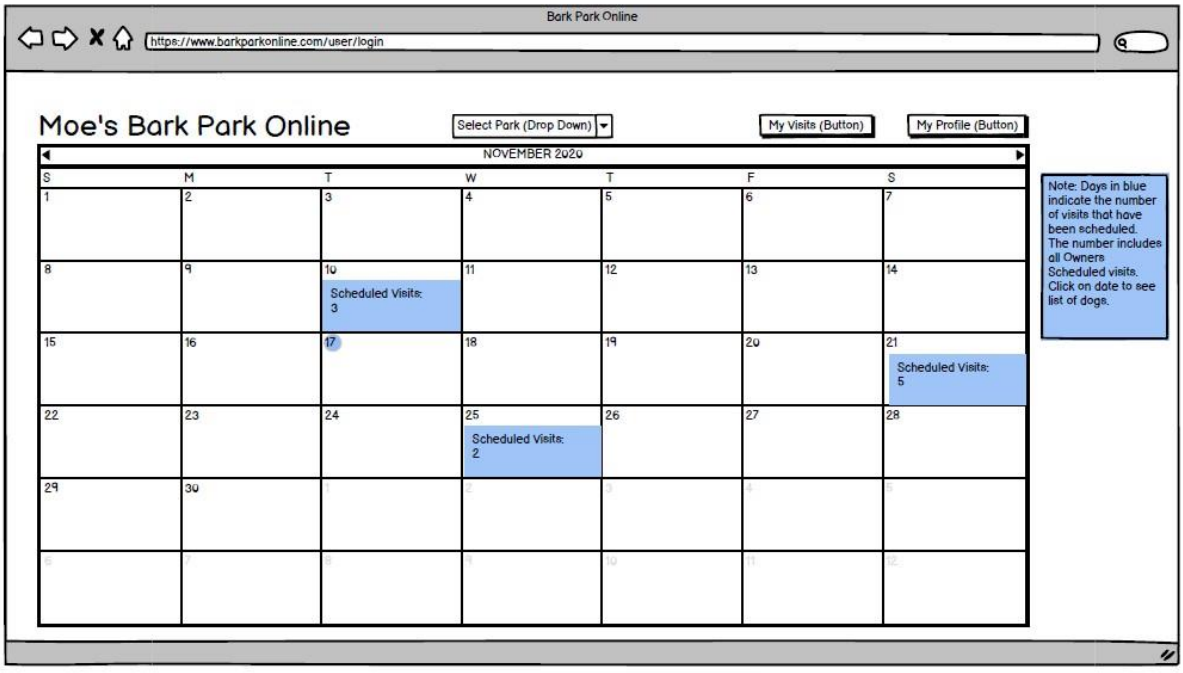
7.1 User Interface Wireframes



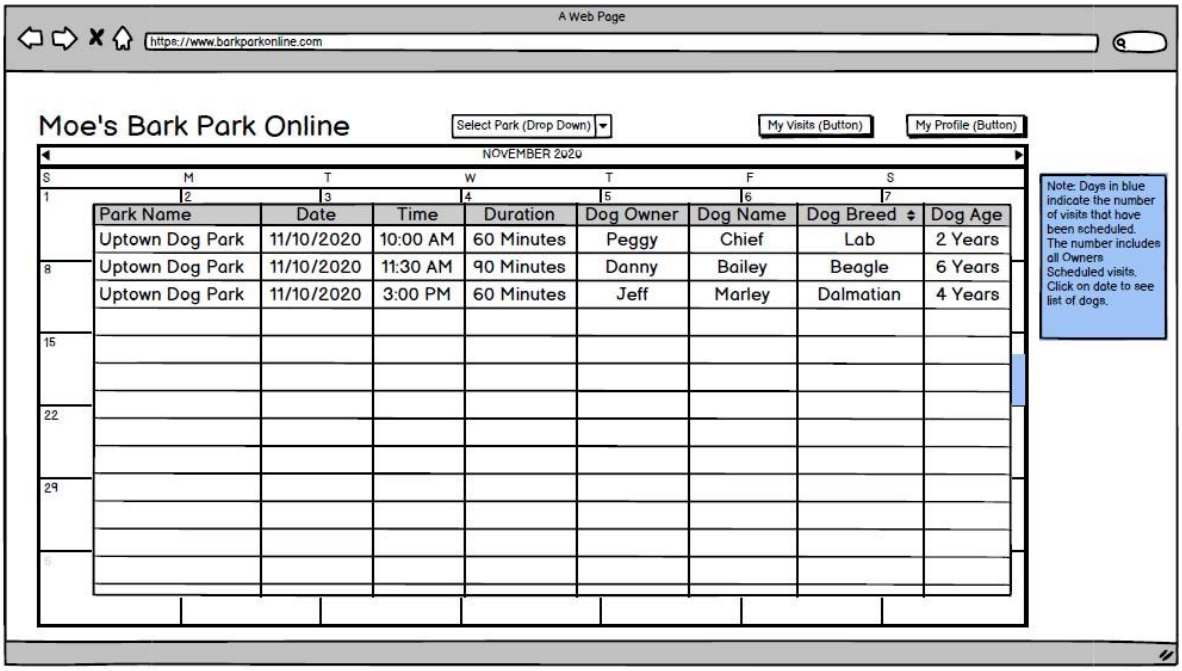
Sign Up 5 / 20



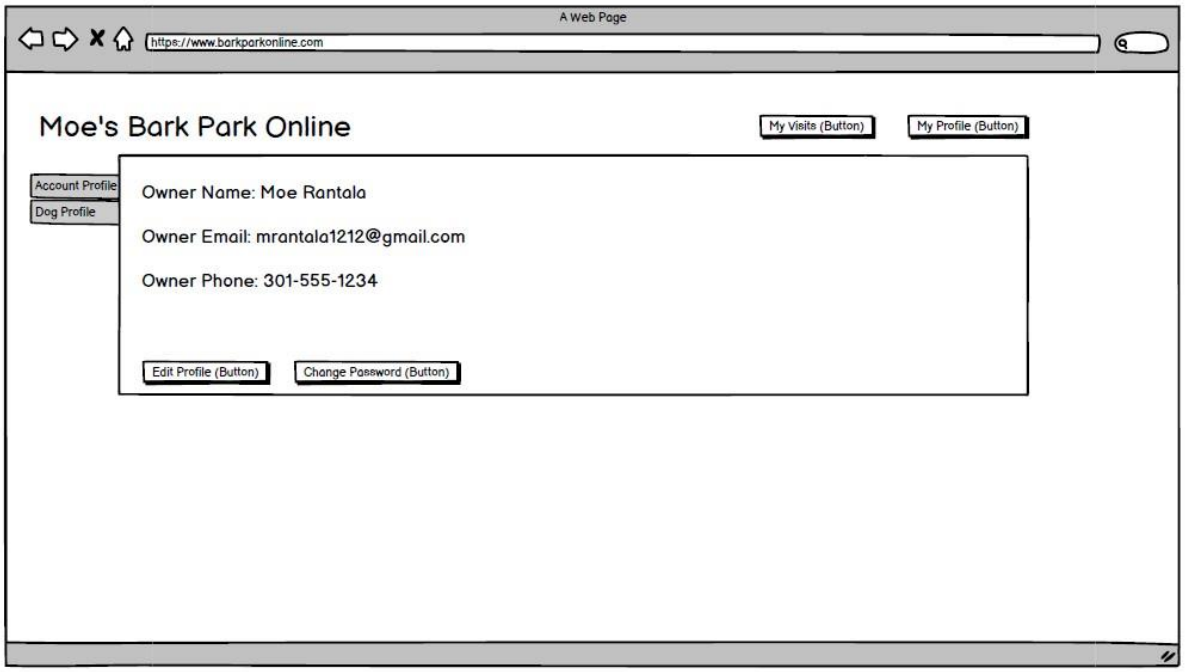
Main 6 / 20



View All Owner Scheduled Visits 7 / 20

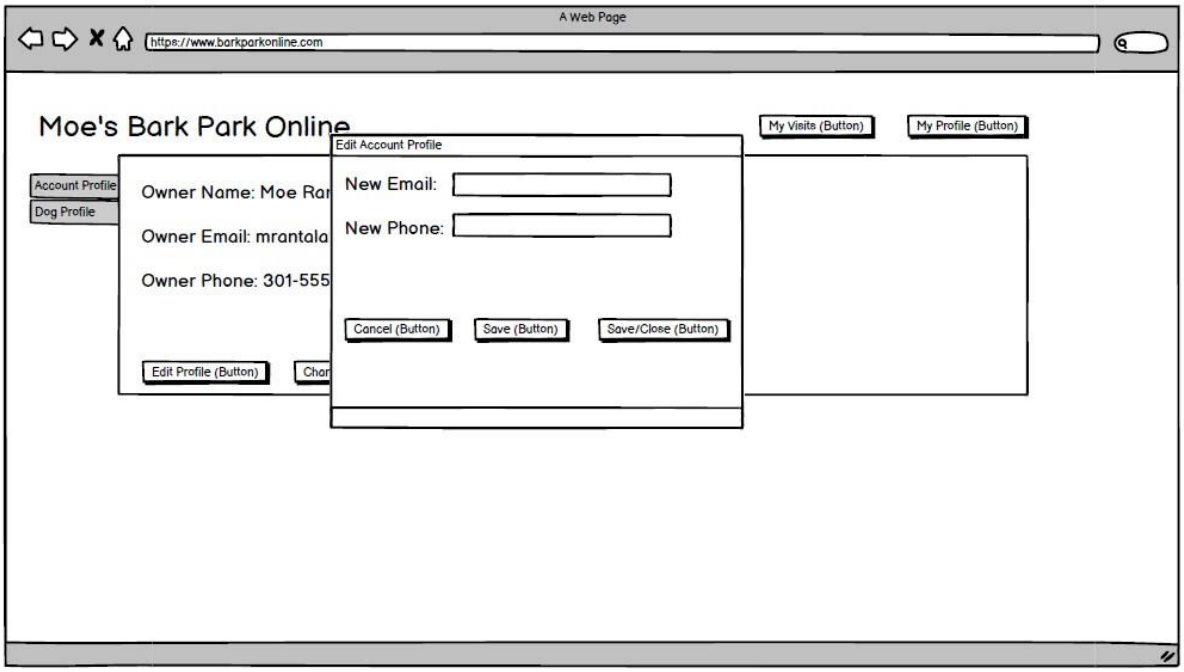


Account Profile 8 / 20



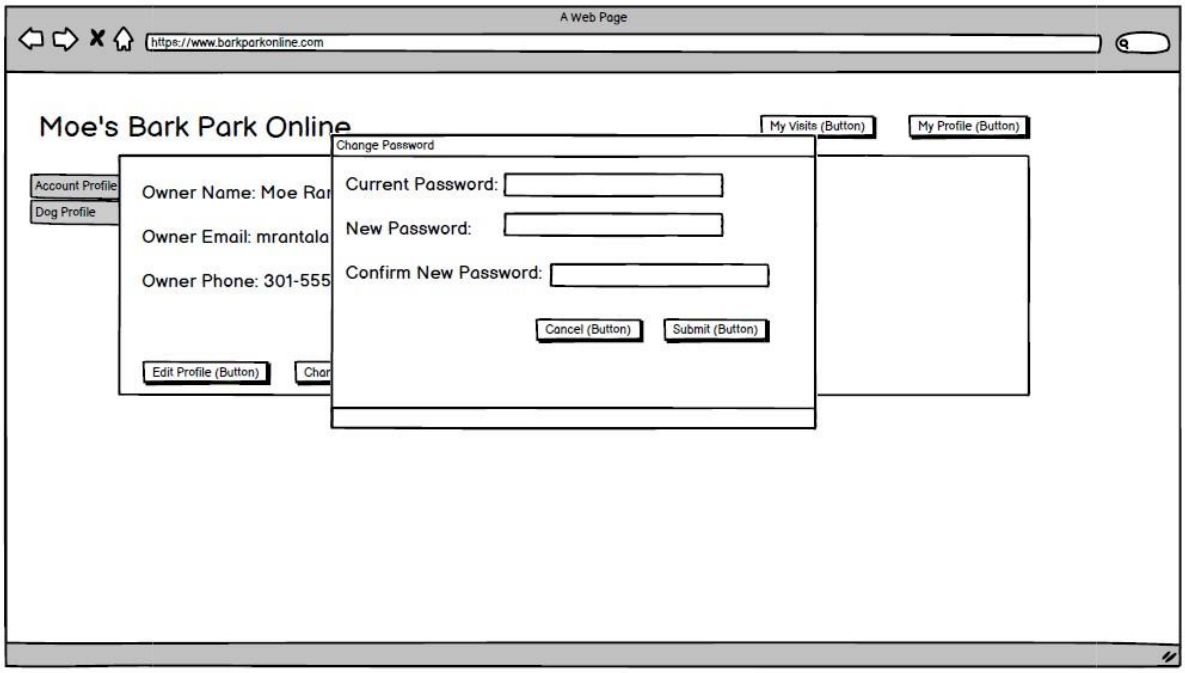
Edit Account Profile

9 / 20



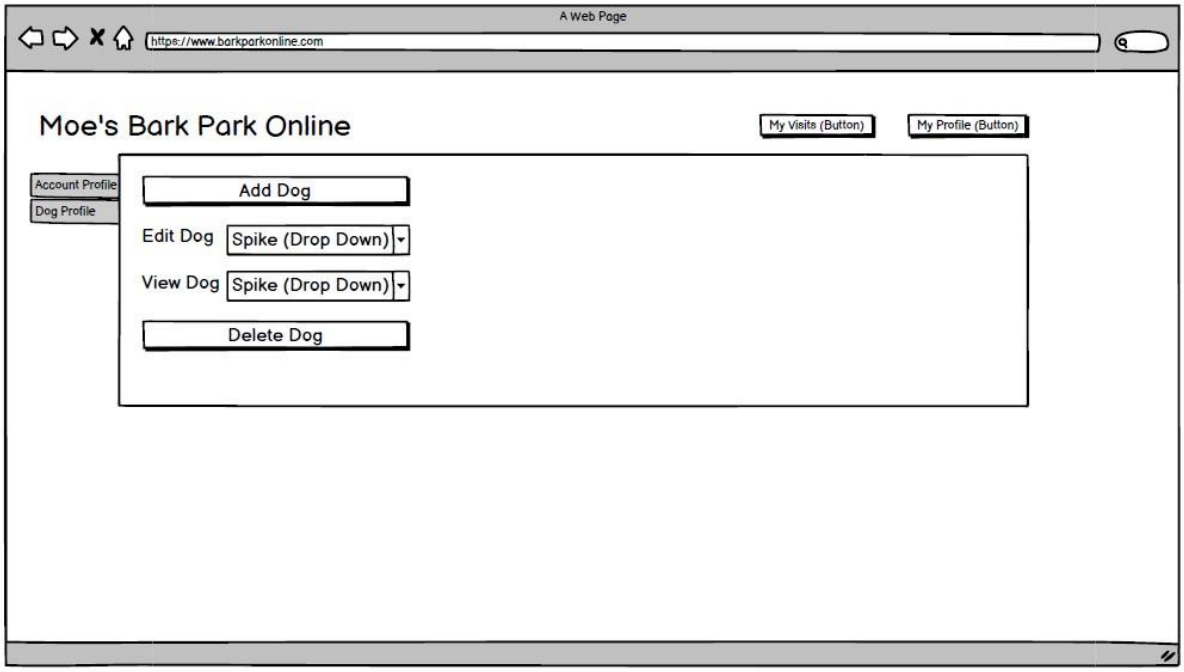
Change Password

10 / 20



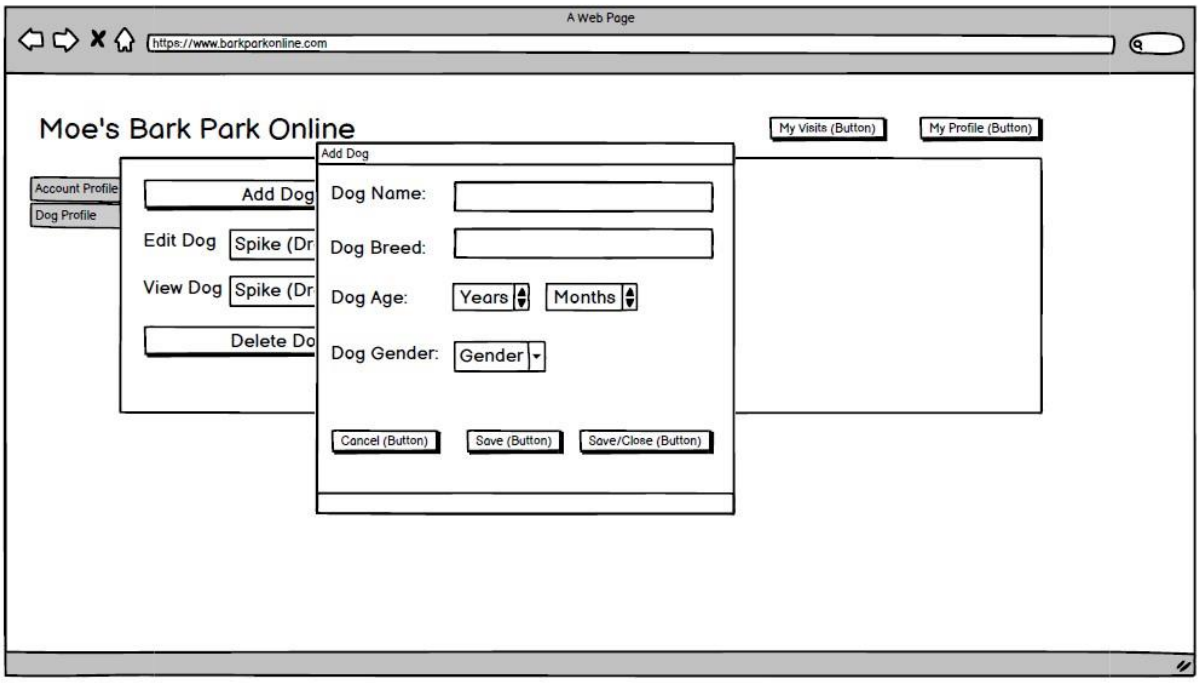
Dog Profile

11 / 20



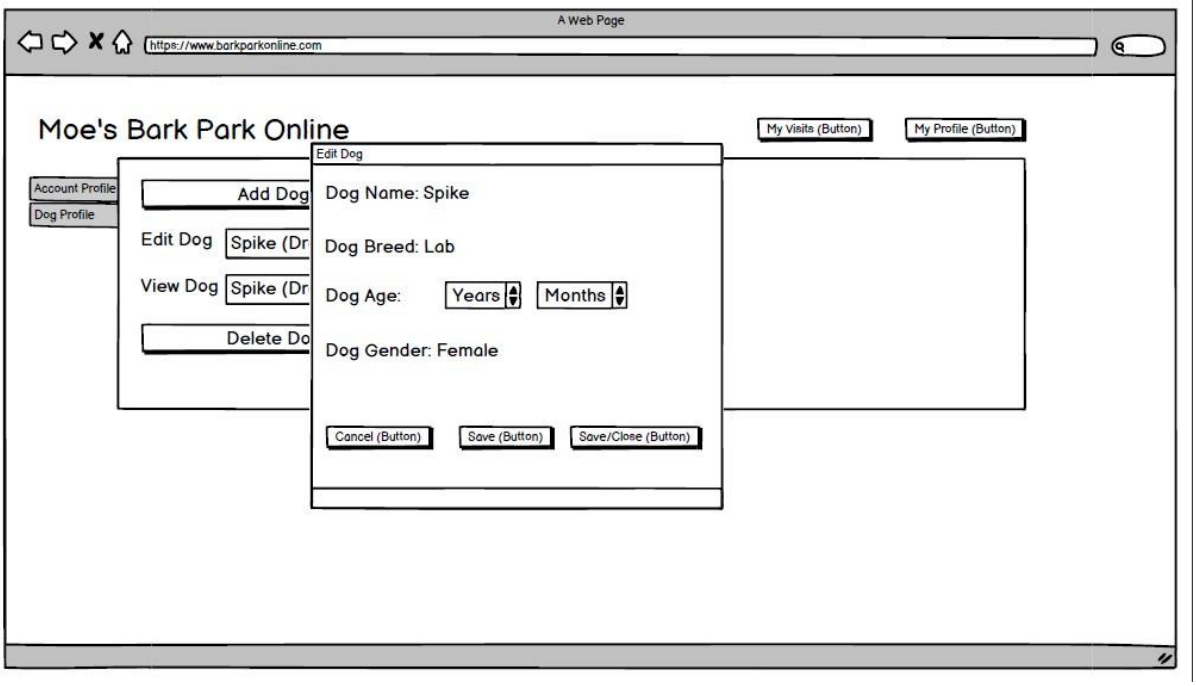
Add Dog

12 / 20



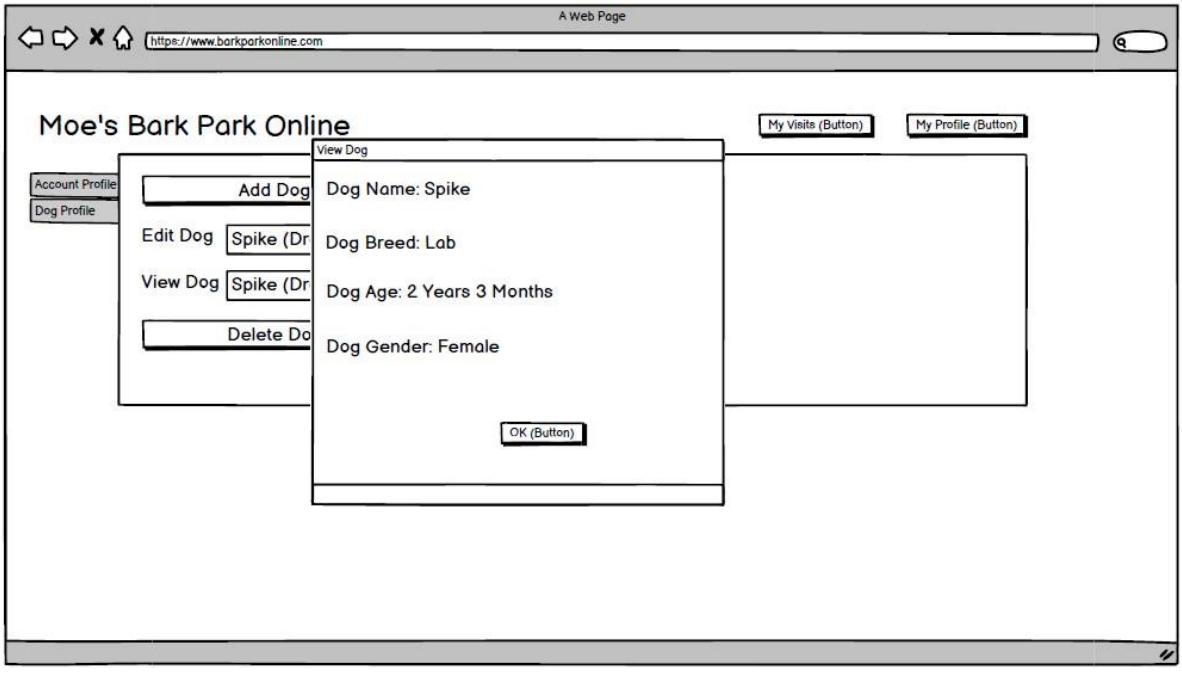
Edit Dog

13 / 20

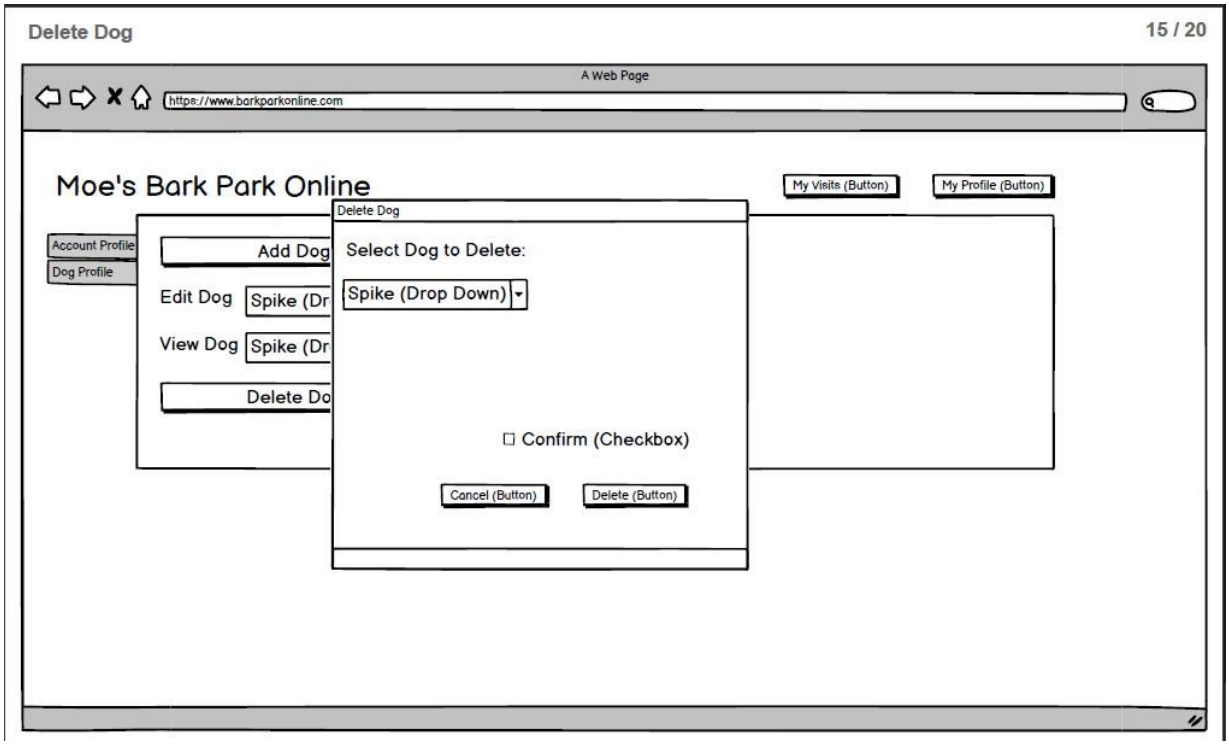


View Dog

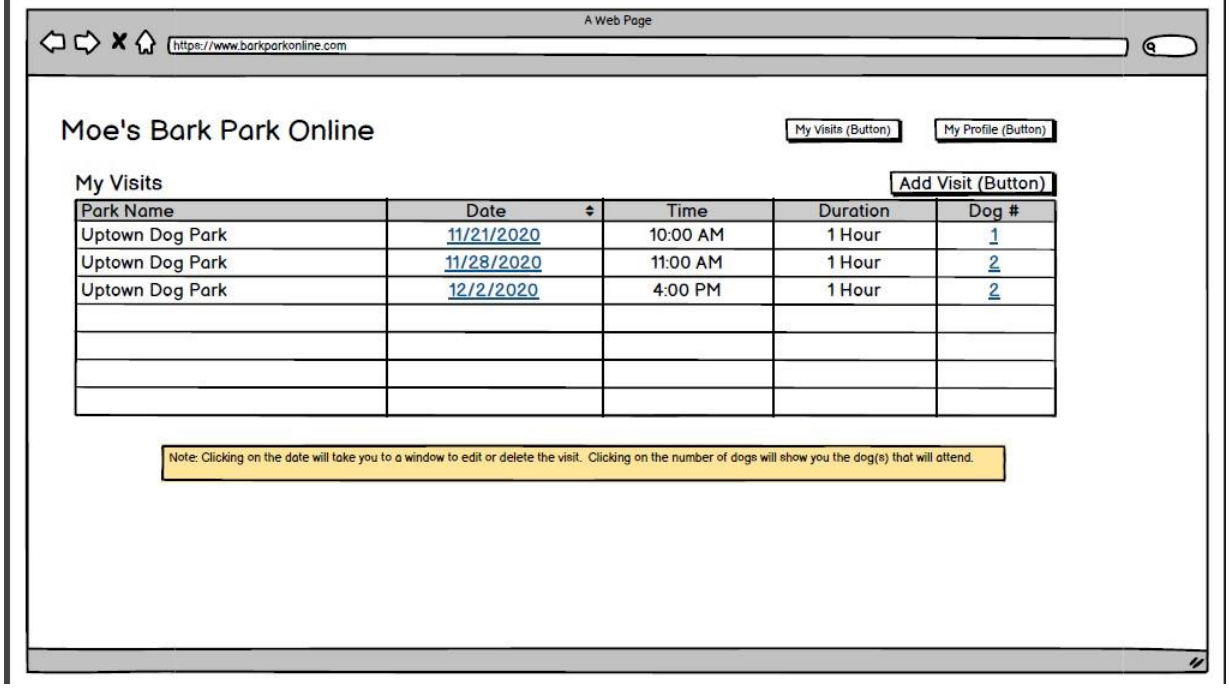
14 / 20

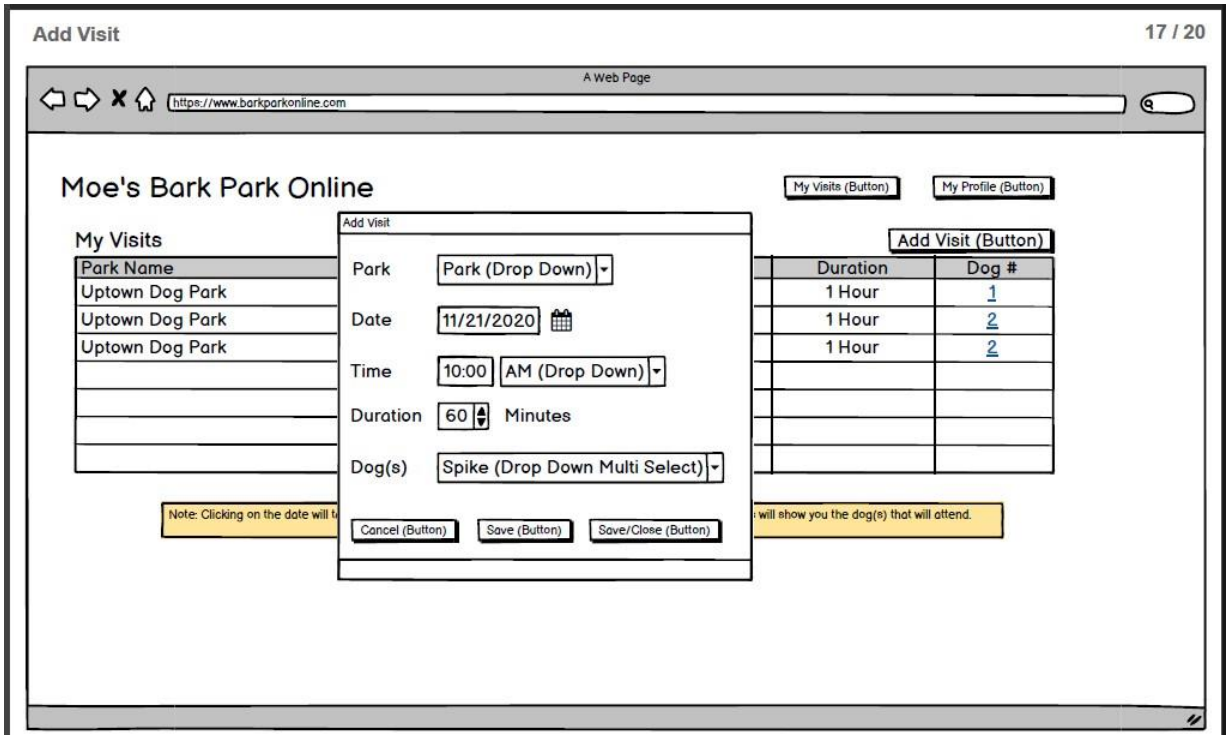


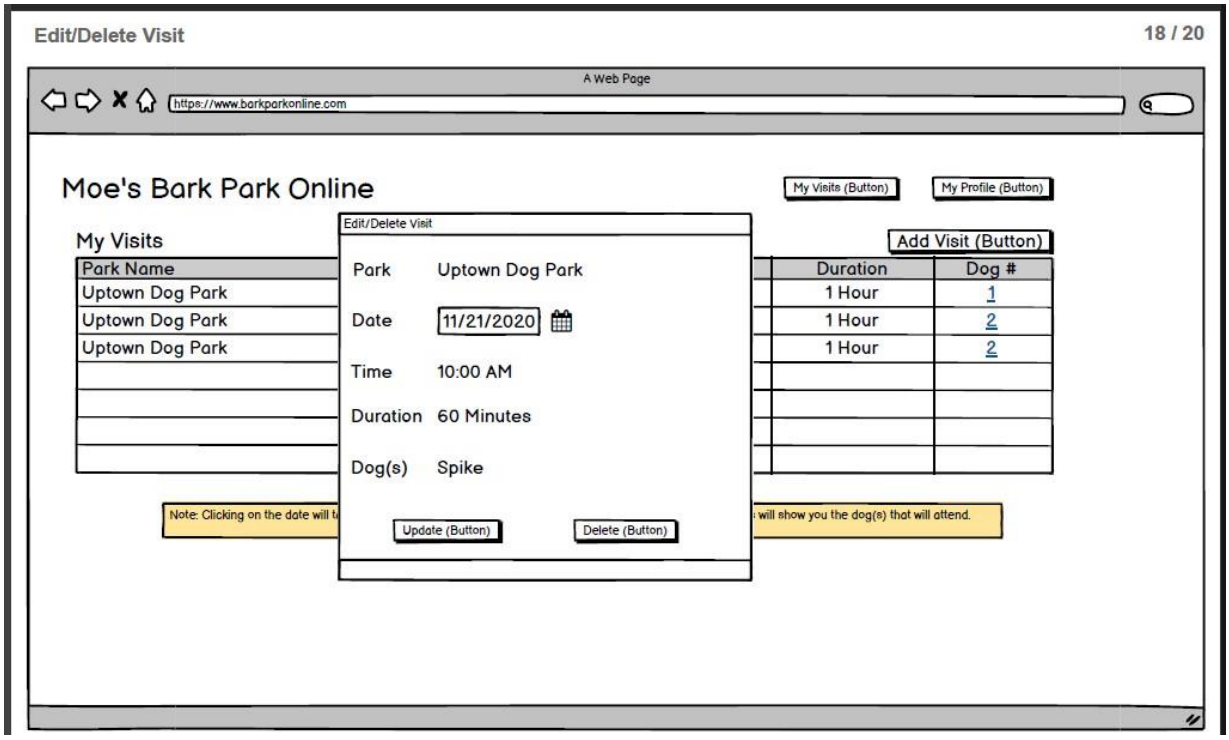
Delete Dog 15 / 20



My Visits 16 / 20

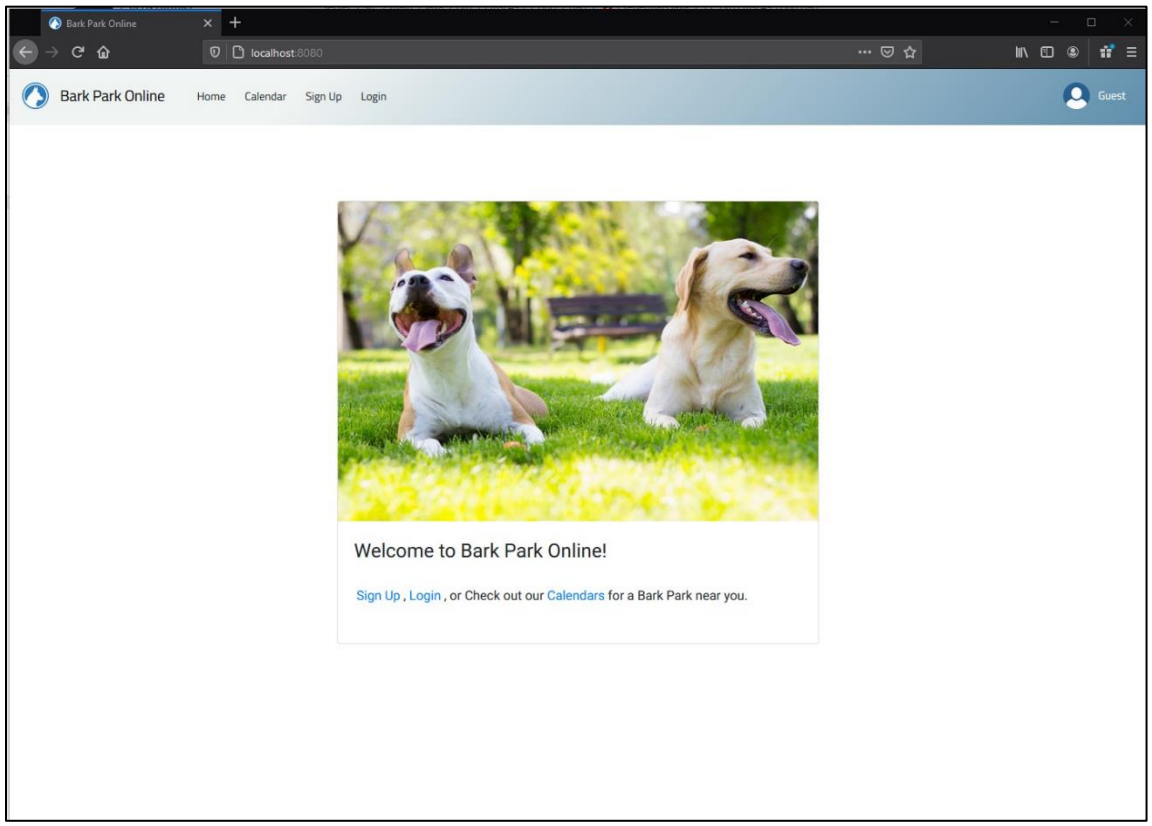




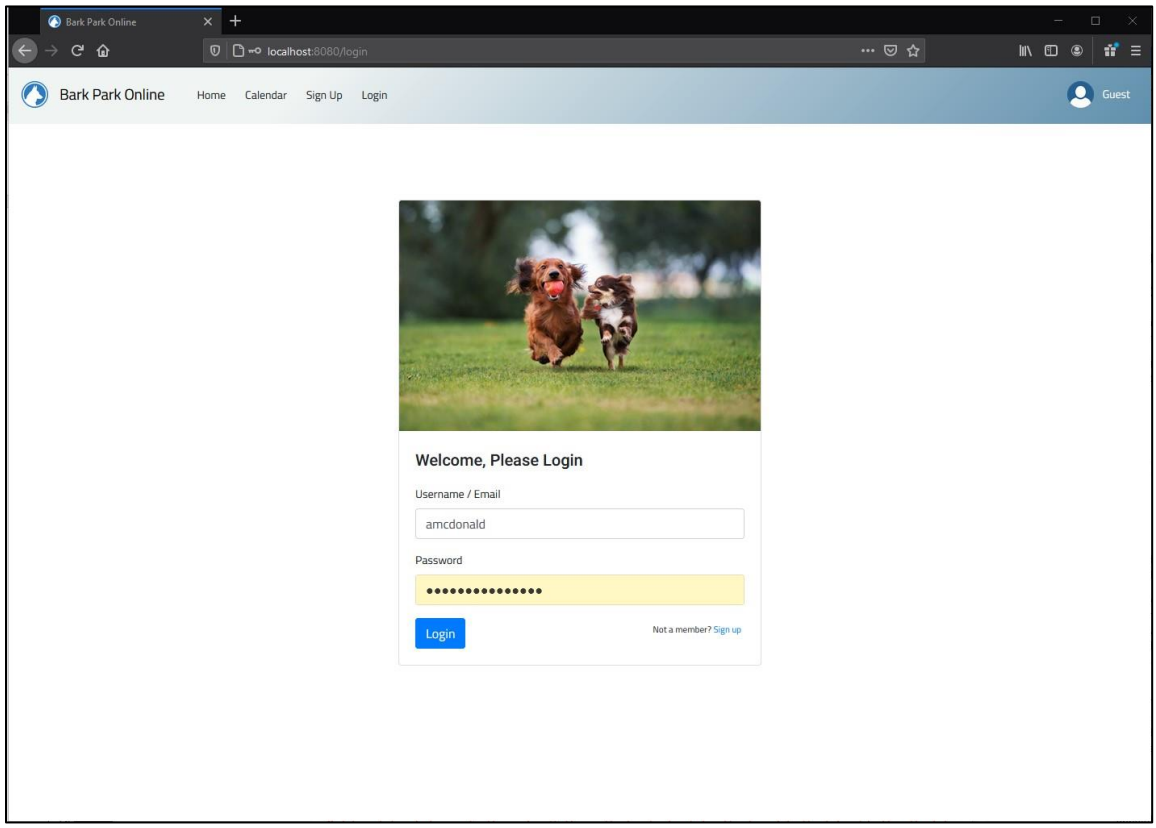


7.2 User Interface Production

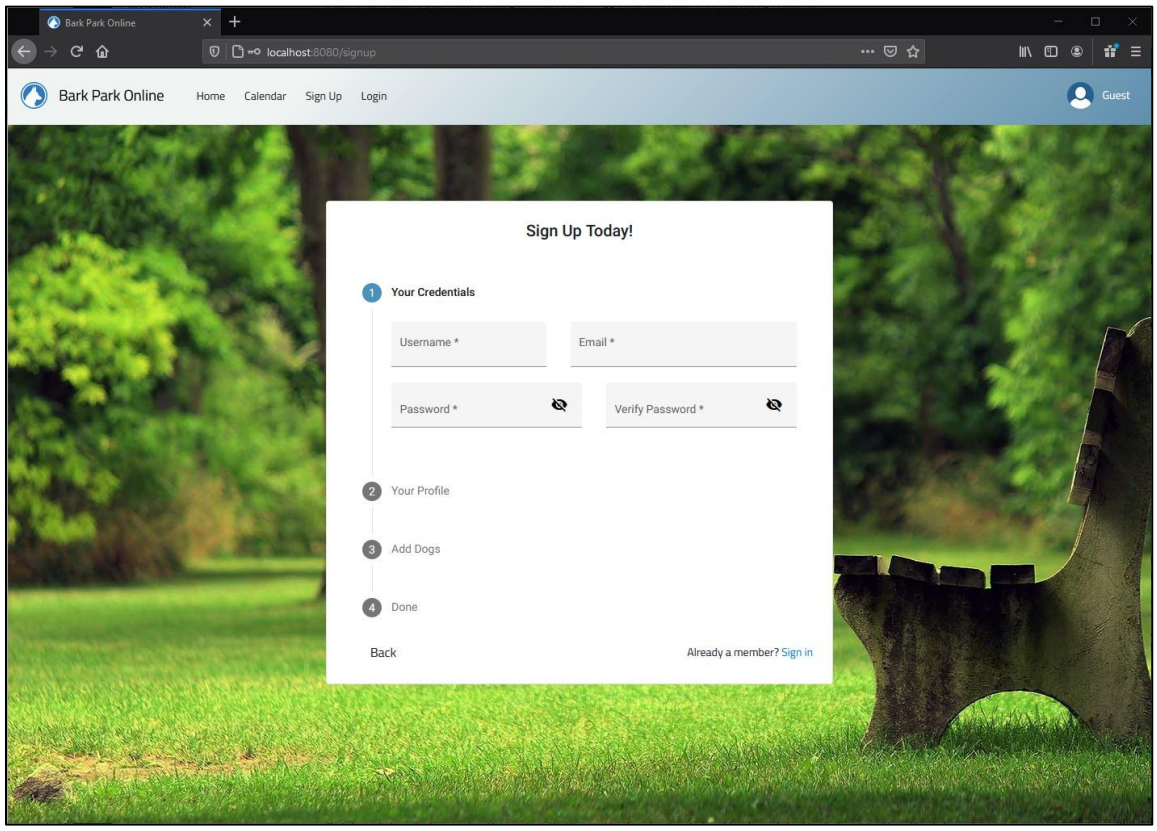
Greeting



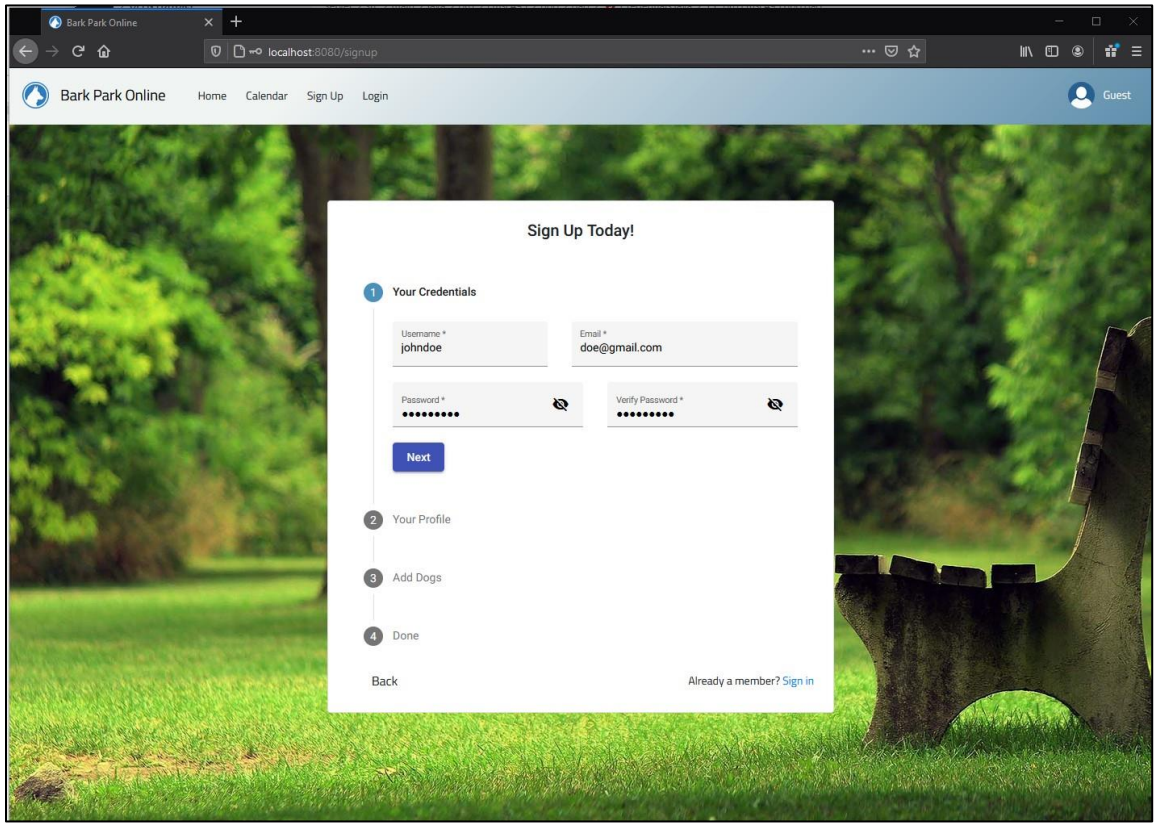
Login



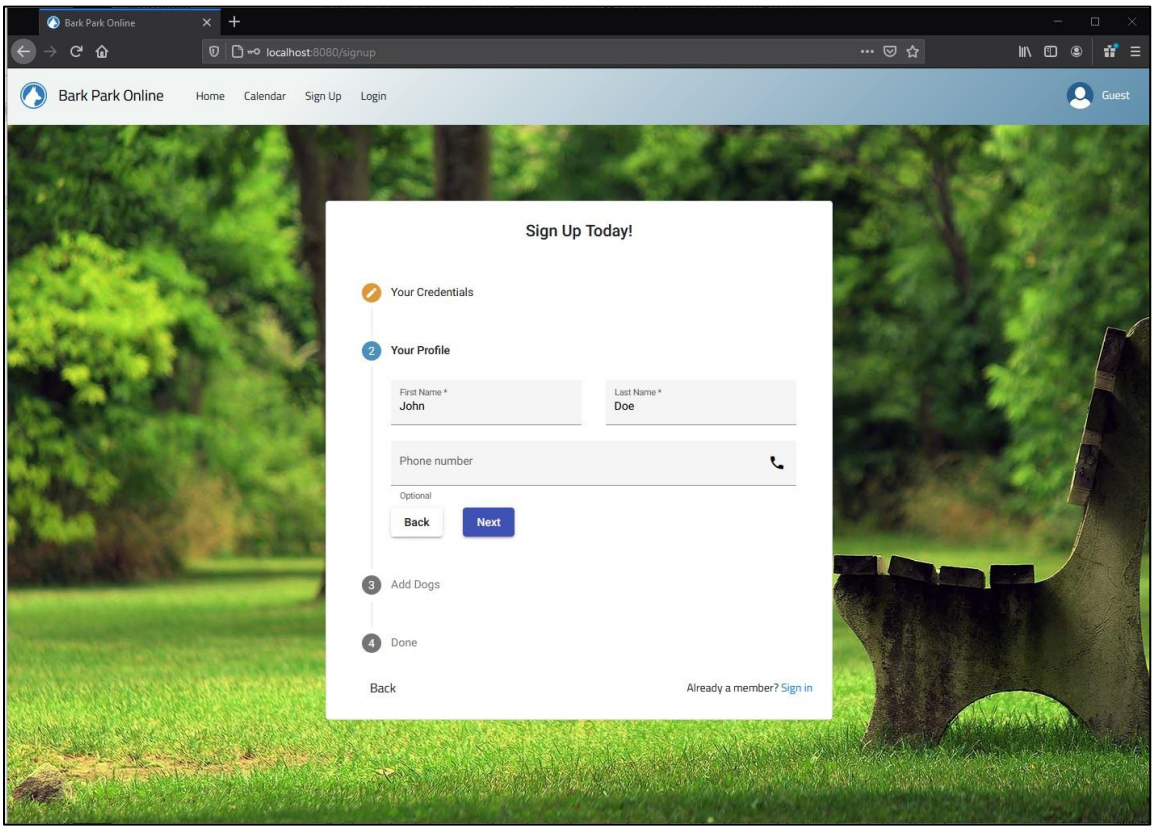
Sign-up 1



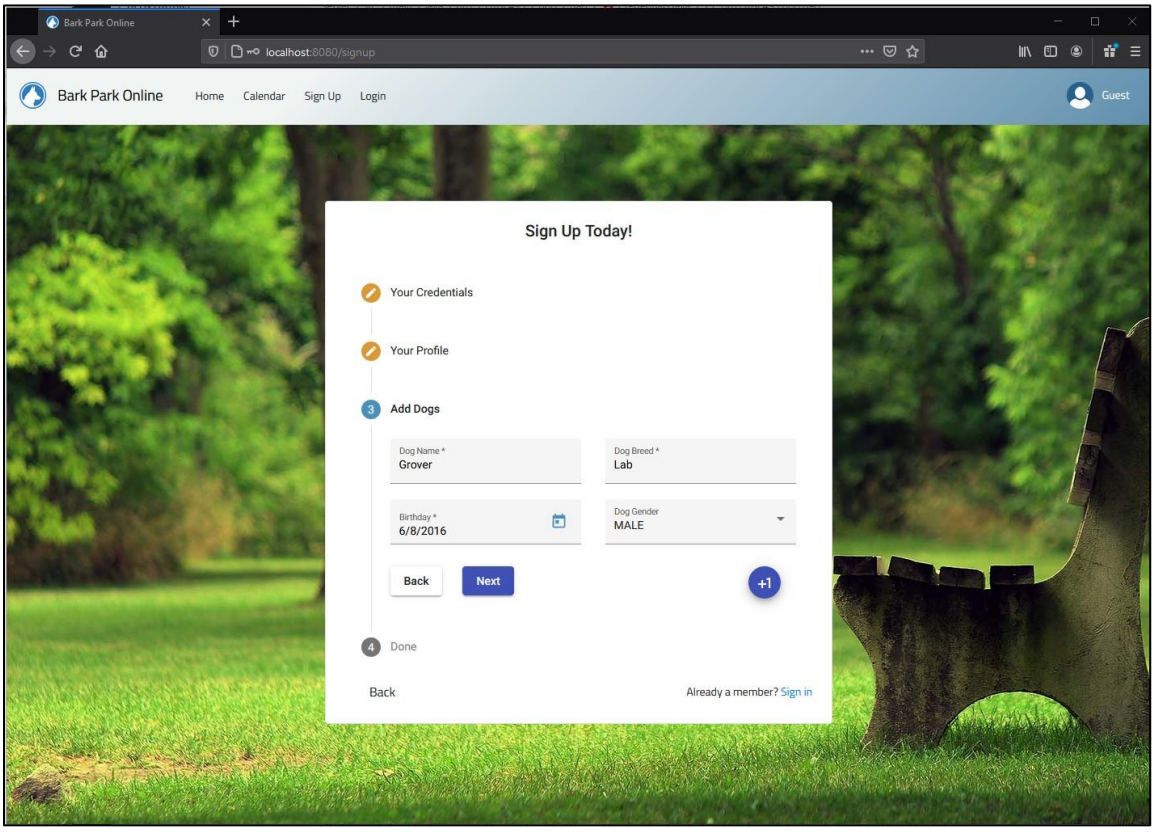
Sign-up 2



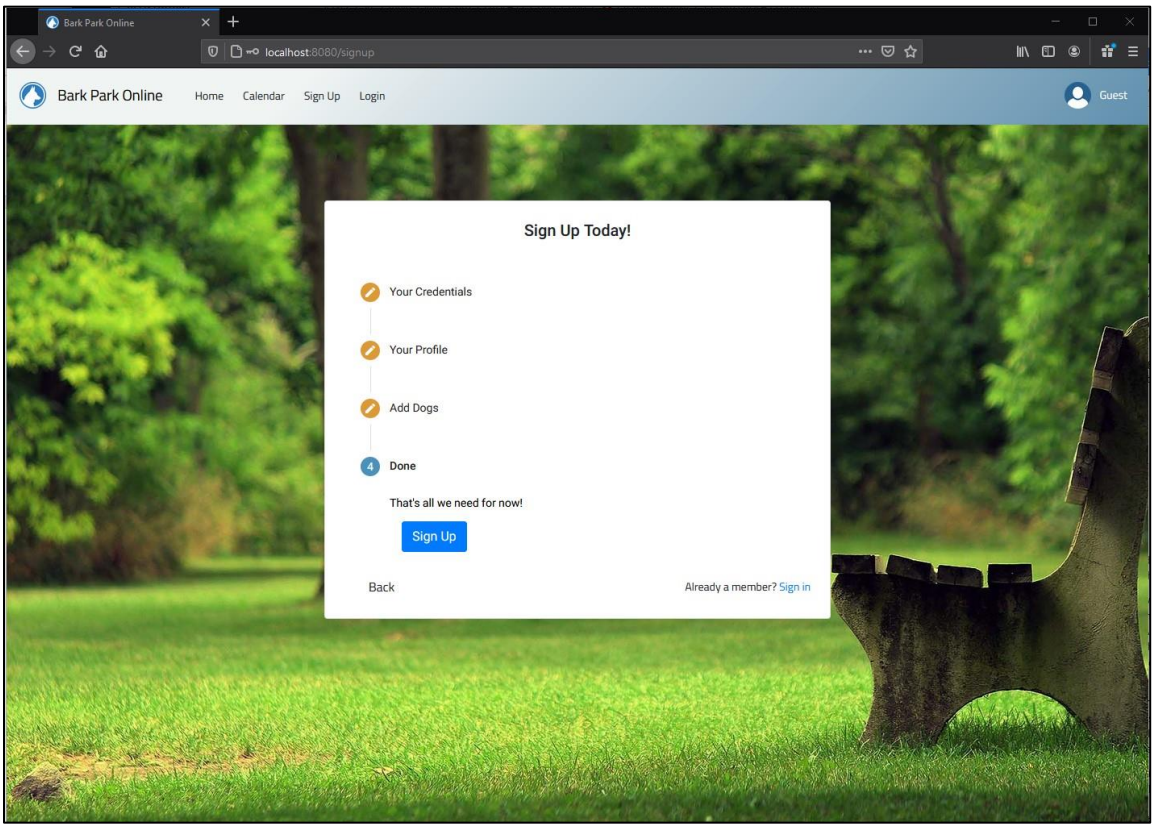
Sign-up 3



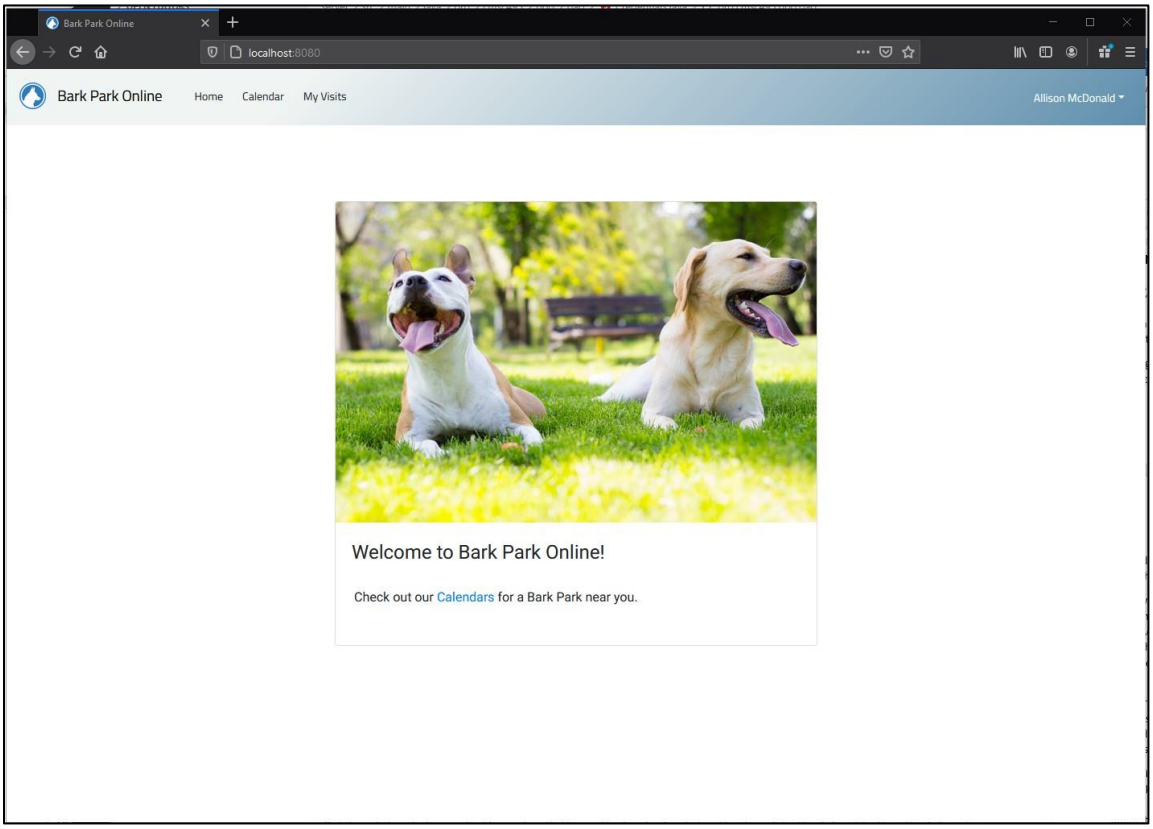
Sign-up 4



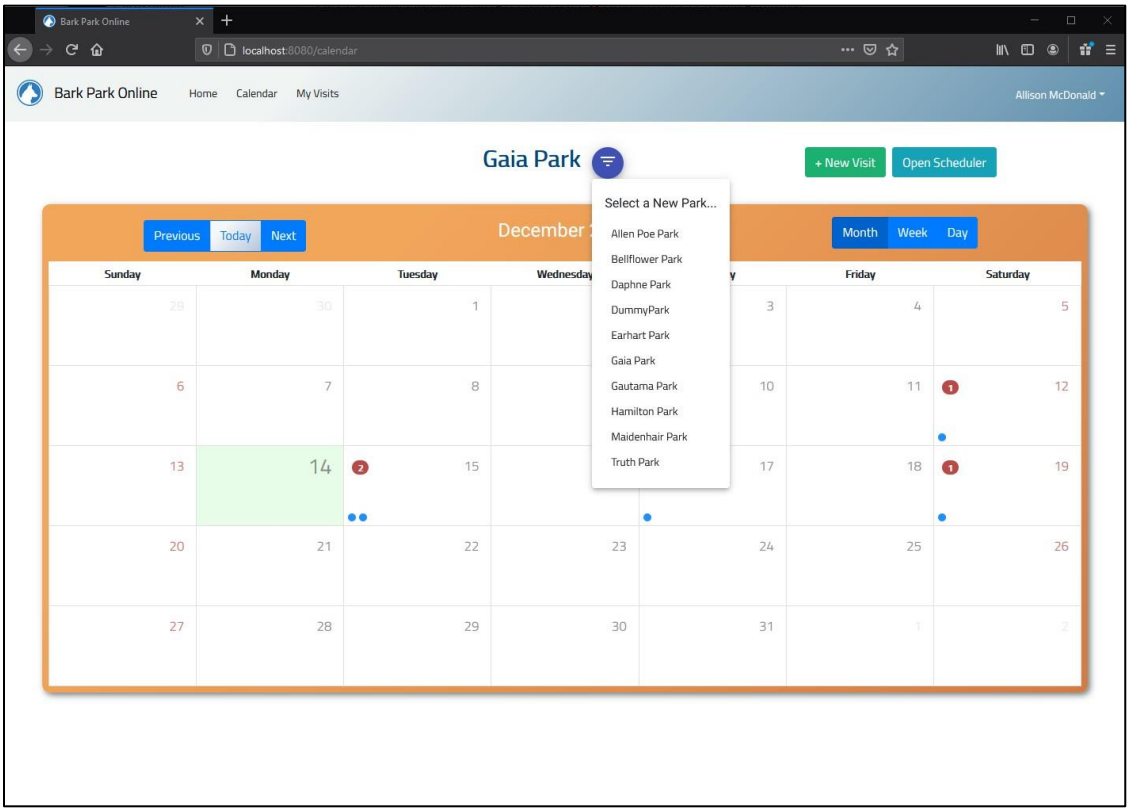
Sign-up 5



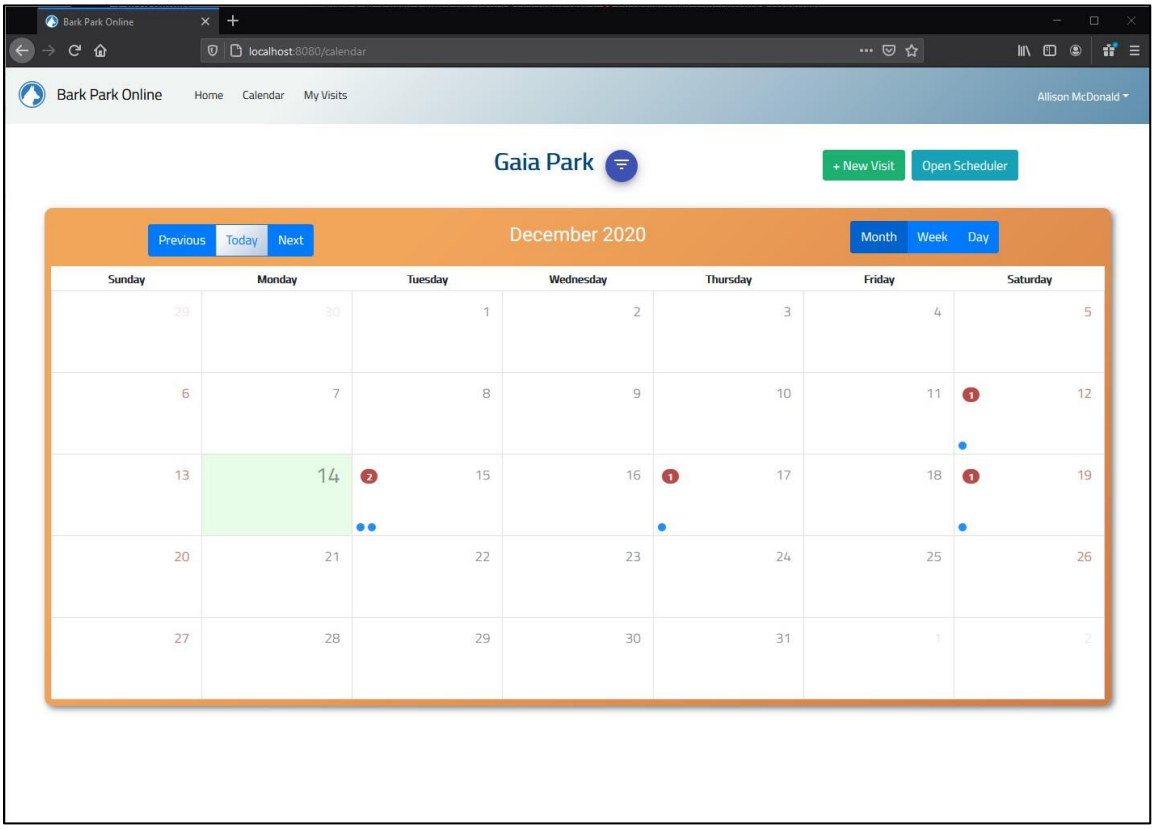
Home



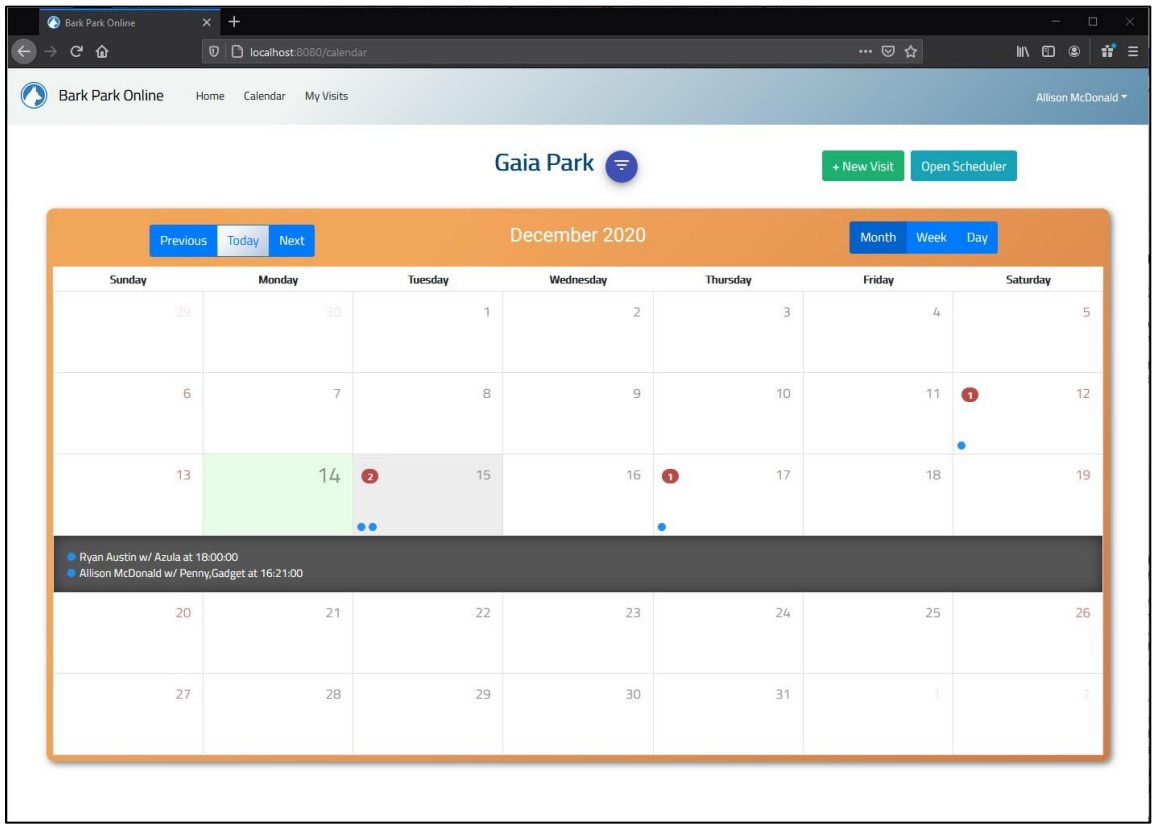
Select Park



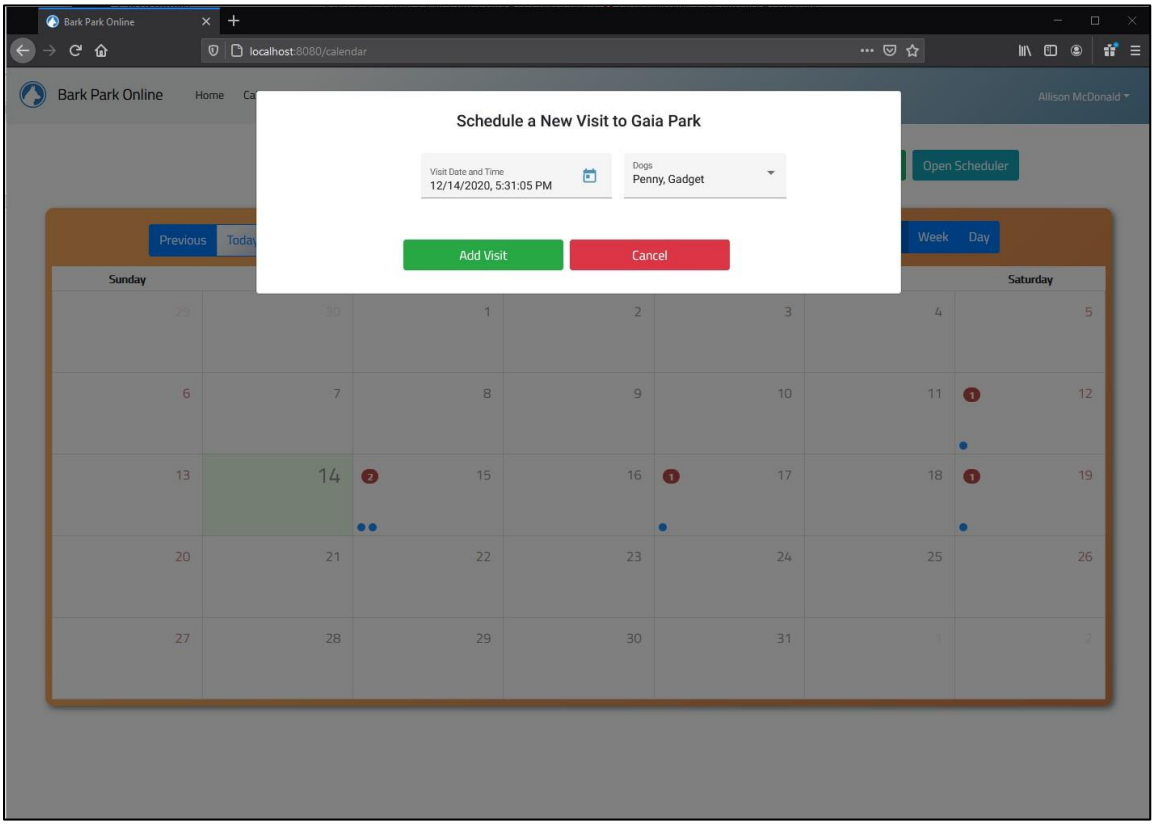
Calendar



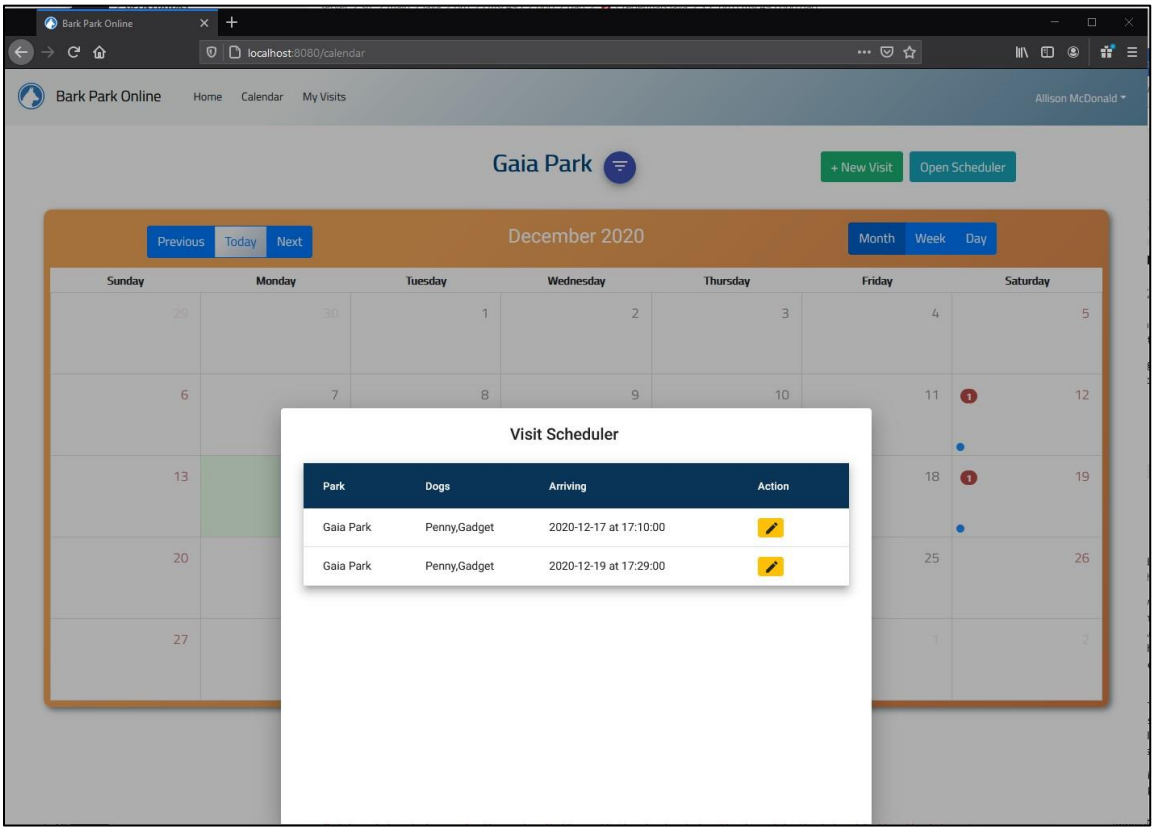
Calendar Showing Visits



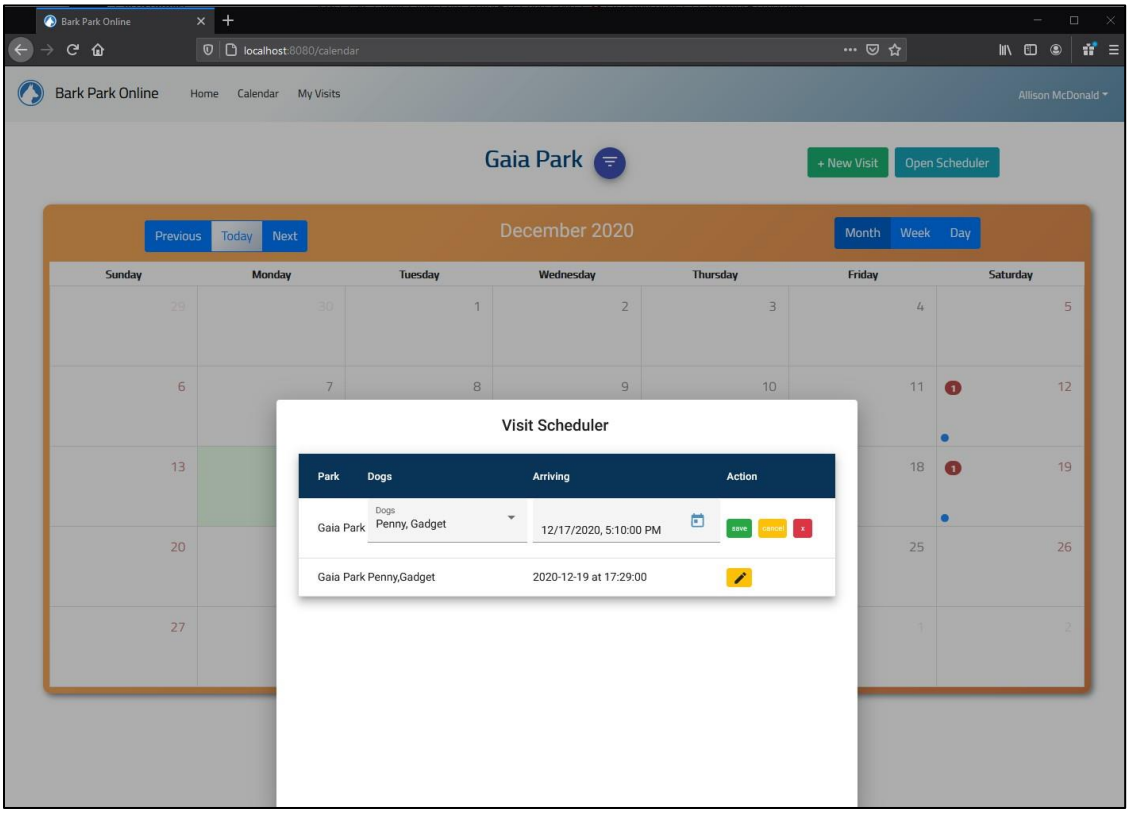
Schedule Visit



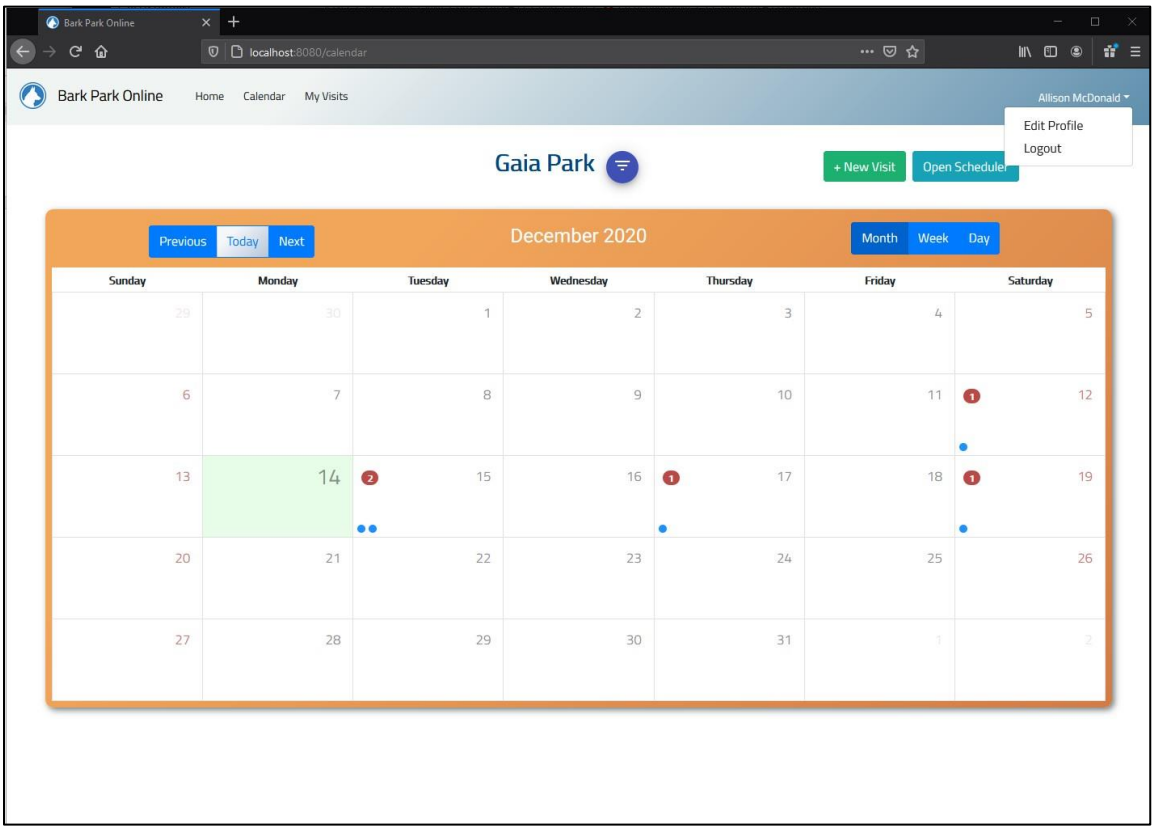
My Visits



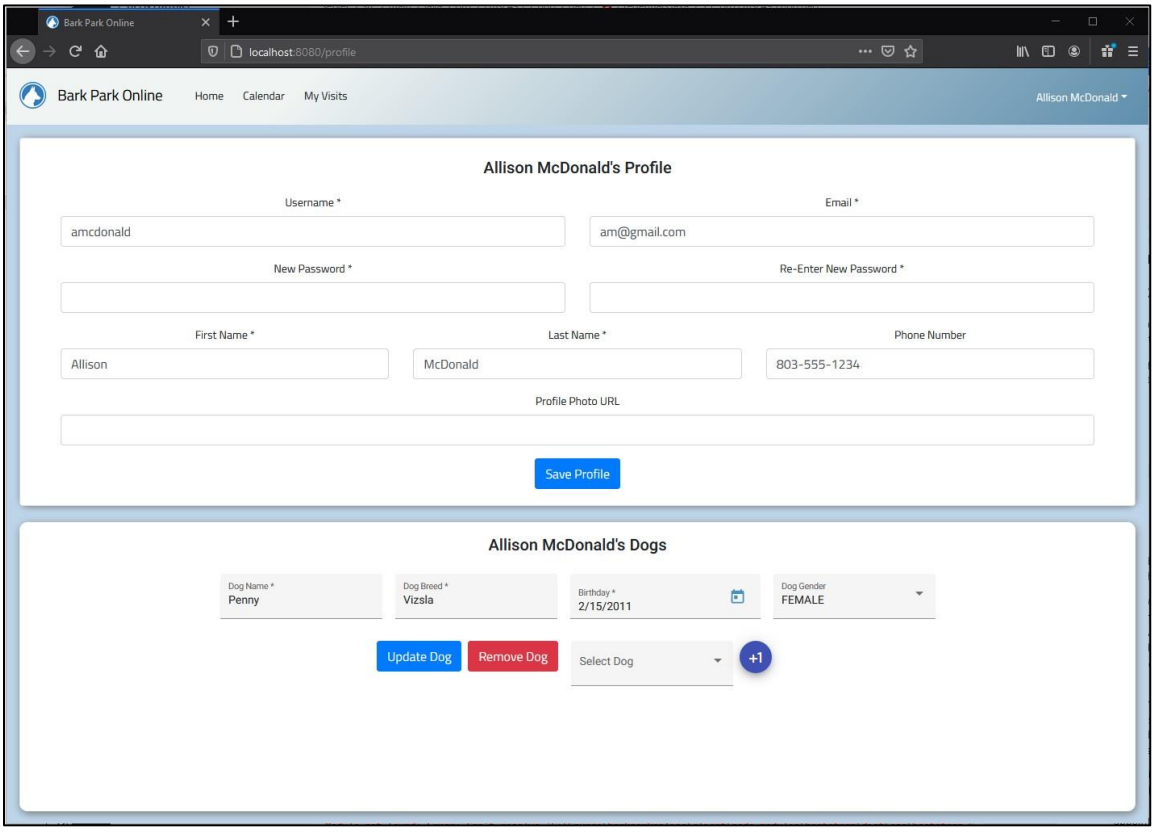
Delete or Edit Visit



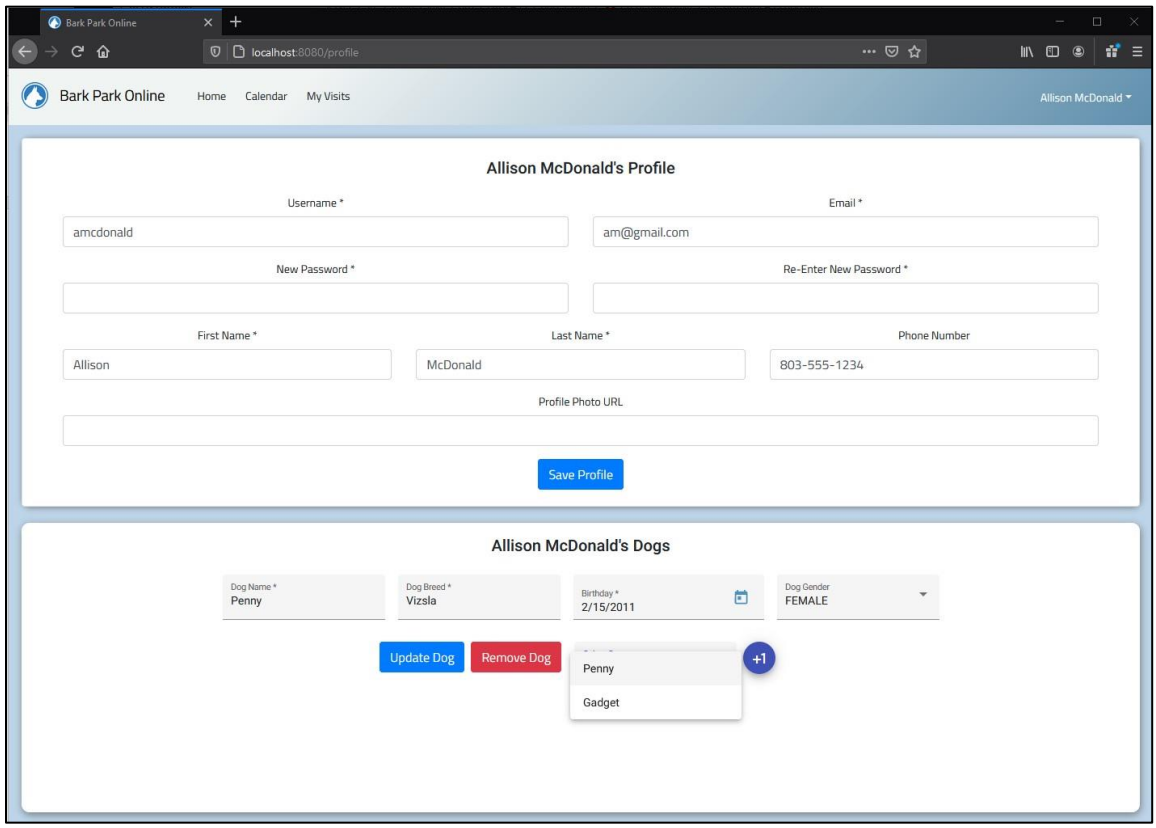
Edit Profile or Logout



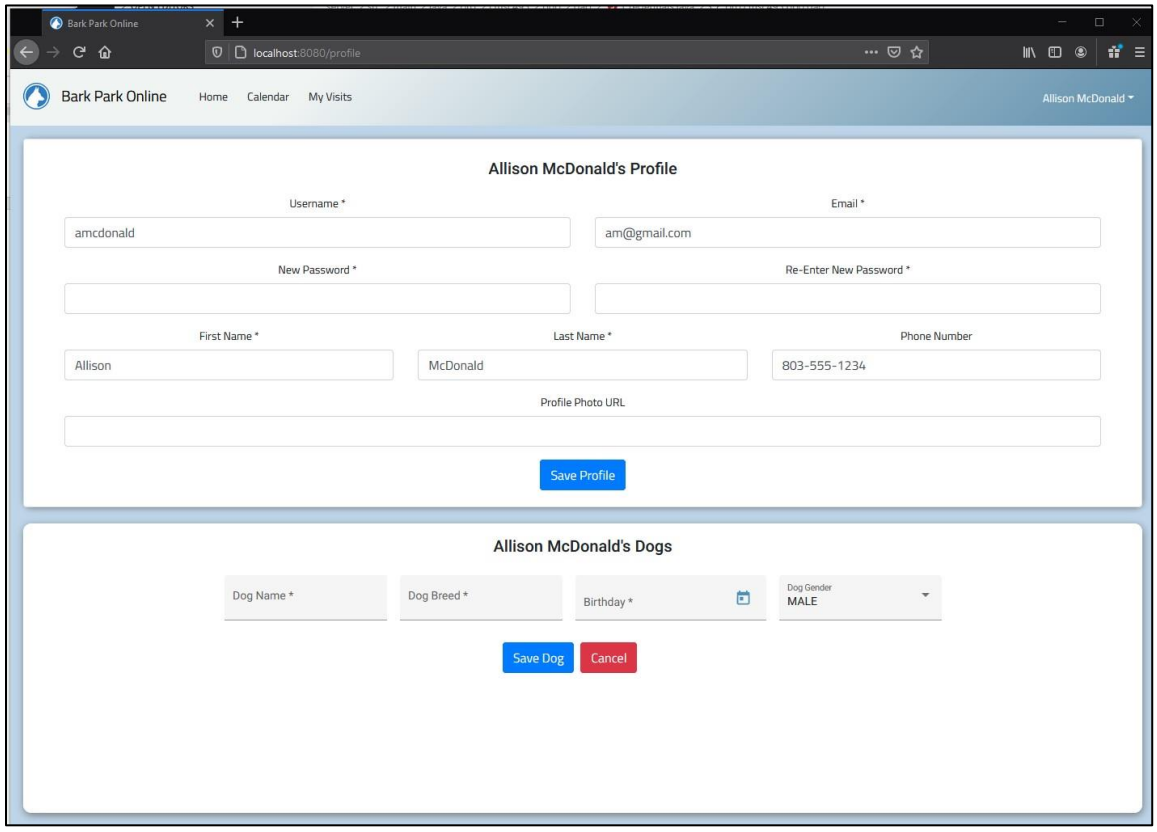
Account Profile



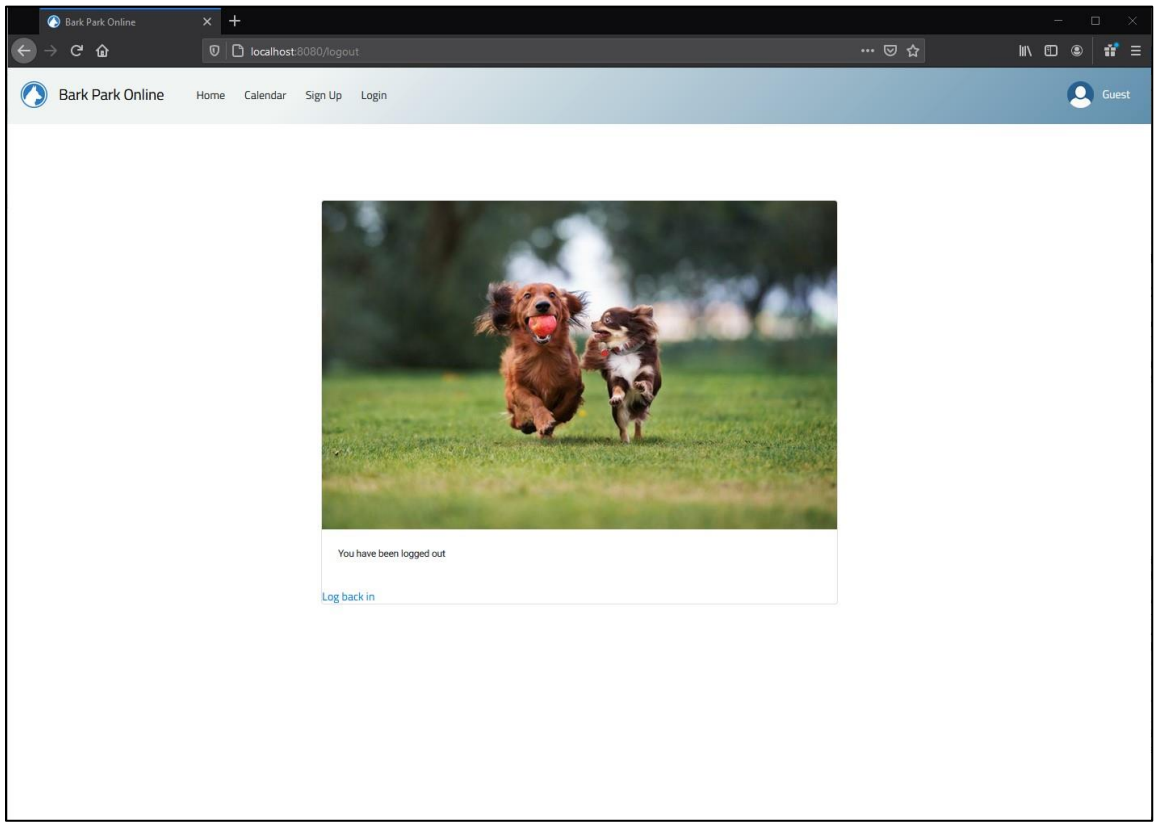
Select Dog



Add Dog



Logout



8. Development History

8.1 Development Overview

Date	Phase/Task	Goal	Complete
11/4/2020	Project Plan	Produce Requirements Specification and Milestone Documentation	**Yes**
11/11/2020	User's Guide and Test Plan	Produce User's Guide, Test Plan, and User Stories	**Yes**
11/18/2020	Design	Produce Design Guide and GUI	**Yes**
11/25/2020	Phase II	For Phase I, our goal was to develop the functionality for Users to Sign-Up for Bark Park Online (BPO). To make this happen, we needed to develop a Sign-Up form on our Angular application and give it the ability to send an HTTP POST to our back-end RESTful Springboot Server. Upon reception of the account creation request, we wired our User Service to store the User within our Mongo Database.	**Yes**

Date	Phase/Task	Goal	Complete
12/2/2020	Phase II	For Phase II, our goal was to develop the functionality the Dog Management and Visit Scheduling for Bark Park Online (BPO). To make this happen, we needed to develop the ability to add dogs after a User is created and visits to forms on our Angular application and give it the ability to send HTTP PUT and HTTP POST to our back-end RESTful Springboot Server. Upon reception of the dog, we wired our Dog Service to store the Dog attached to the User within our Mongo Database. Upon reception of the visit, we wired our Scheduling Service to store the Visit attached to the User within our Mongo Database.	**Yes**
12/9/2020	Phase III	For Phase III, our goal was to further refine the Bark Park project by implementing additional functionalities of the classes and to improve the test processes. Specifically, to implement the Edit Visit in CalendarController and endpoints for Dog Management. Implementation of Calendar component functionalities was another objective. Functionalities in the front-end were also implemented for these. We also wanted to make the Navbar accessible from all the pages of the Bar Park Online. As an additional stretch effort, we also aimed to implement functionality to remove a user, i.e., cancel the account. To improve the quality of the project, we aimed to implement a unit test for all the classes under bpo to achieve 50% test coverage.	**Yes**
12/15/2020	Final Deliverable	Software Build: All tests will have been conducted and reported to group members to ensure no further bugs exist and the application is ready for production.	**Yes**

8.2 Projected Schedule

Task		Start Date	End Date	Team Member
Requirements Specification Documentation				
	Write	11/1/2020	11/3/2020	Allison McDonald
	Review	11/2/2020	11/3/2020	Team
System Specification Documentation				
	Write	11/1/2020	11/3/2020	Allison McDonald
	Review	11/2/2020	11/3/2020	Team
Milestone Documentation				
	Write	11/2/2020	11/3/2020	Allison McDonald
	Review	11/2/2020	11/3/2020	Team
User's Guide				

Task		Start Date	End Date	Team Member
	Write	11/4/2020	11/10/2020	Allison McDonald
	Review	11/8/2020	11/9/2020	Team
Test Plan				
	Setup E2E Test Environment	11/4/2020	11/10/2020	Ryan Austin
	Setup Unit Test Environment	10/31/2020	11/10/2020	Ryan Austin
	Setup Code Coverage (JaCoCo)	10/31/2020	11/10/2020	Ryan Austin
	Create Test Database	11/4/2020	11/10/2020	Raymond "Moe" Rantala
Test Plan Guide				
	Contribute User Stories	11/4/2020	11/7/2020	Team
	Write	11/4/2020	11/10/2020	Allison McDonald
	Review	11/8/2020	11/9/2020	Team
Setup Everyone's Workstations				
	Synced with GitLab	11/4/2020	11/10/2020	Team
	Can compile/build	11/4/2020	11/10/2020	Team
	Can load web server	11/4/2020	11/10/2020	Team
	Can load client	11/4/2020	11/10/2020	Team
Design Guide				
	Contribute	11/11/2020	11/14/2020	Team
	Write	11/11/2020	11/17/2020	Allison McDonald
	Review	11/15/2020	11/16/2020	Team
Design State Objects				
	Owner/Owner Profile	11/11/2020	11/17/2020	Ryan Austin Raymond "Moe" Rantala
	Dog/Dog Profile	11/11/2020	11/17/2020	Ryan Austin Raymond "Moe" Rantala
	Park/Park Profile	11/11/2020	11/17/2020	Ryan Austin Raymond "Moe" Rantala
	Calendar Object Model (backend)	11/11/2020	11/17/2020	Anas Abdulrazzaq Christina Reiss
Design Decisions				
	Password Management	11/11/2020	11/17/2020	TBD
	Sign in protocol	11/11/2020	11/17/2020	TBD
	Sign up protocol	11/11/2020	11/17/2020	TBD
GUI				
	Layout	11/11/2020	11/17/2020	Team
Phase I Report				
	Write	11/18/2020	11/24/2020	Allison McDonald
	Review	11/22/2020	11/23/2020	Team
Controllers				
	Owner Controller	11/18/2020	11/24/2020	Anas Abdulrazzaq Christina Reiss
	Dog Management Controller	11/18/2020	11/24/2020	Anas Abdulrazzaq Christina Reiss
Endpoints				

Task		Start Date	End Date	Team Member
	Add Owner (Sign up)	11/18/2020	11/24/2020	Anas Abdulrazzaq Christina Reiss
	Login	11/18/2020	11/24/2020	Anas Abdulrazzaq Christina Reiss
	Logout	11/18/2020	11/24/2020	Anas Abdulrazzaq Christina Reiss
	Add Dog to Owner	11/18/2020	11/24/2020	Anas Abdulrazzaq Christina Reiss
	Remove Dog from Owner	11/18/2020	11/24/2020	Anas Abdulrazzaq Christina Reiss
Services				
	Basic Owner Service	11/18/2020	11/24/2020	Anas Abdulrazzaq Christina Reiss
	Dog Management Service	11/18/2020	11/24/2020	Anas Abdulrazzaq Christina Reiss
Client				
	Sign up page with form	11/18/2020	11/24/2020	Ryan Austin Raymond "Moe" Rantala
	Initial Calendar Page Design (Stretch Goal)	11/18/2020	11/24/2020	Ryan Austin Raymond "Moe" Rantala
Database				
	Setup User Database	11/18/2020	11/24/2020	Allison McDonald Raymond "Moe" Rantala
	Setup Park Calendar Database	11/18/2020	11/24/2020	Allison McDonald Raymond "Moe" Rantala
Phase II Report				
	Write	11/25/2020	12/1/2020	Allison McDonald
	Review	11/29/2020	11/30/2020	Team
Controllers				
	Calendar Controller	11/25/2020	12/1/2020	Anas Abdulrazzaq Christina Reiss
Endpoints				
	Update Profile	11/25/2020	12/1/2020	Anas Abdulrazzaq Christina Reiss
	Get Calendar Month	11/25/2020	12/1/2020	Anas Abdulrazzaq Christina Reiss
	Get Calendar Day	11/25/2020	12/1/2020	Anas Abdulrazzaq Christina Reiss
Services				
	Calendar Scheduling Service	11/25/2020	12/1/2020	Anas Abdulrazzaq Christina Reiss
Security				
	Use Secure Method to handle and store passwords	11/25/2020	12/1/2020	TBD
	Check for security concerns	11/25/2020	12/1/2020	TBD

Task		Start Date	End Date	Team Member
Client				
	Design or finish Calendar Page	11/25/2020	12/1/2020	Ryan Austin Raymond "Moe" Rantala
	Profile (Update) Page	11/25/2020	12/1/2020	Ryan Austin Raymond "Moe" Rantala
Database				
	Populate Database	11/25/2020	12/1/2020	Allison McDonald Raymond "Moe" Rantala
	Document actual Owner/Passwords for tests/demos	11/25/2020	12/1/2020	Allison McDonald Raymond "Moe" Rantala
Phase III Report				
	Write	12/2/2020	12/8/2020	Allison McDonald
	Review	12/6/2020	12/7/2020	Team
Testing				
	End to End	12/2/2020	12/8/2020	Team
Final Report				
	Conclusions	12/9/2020	12/13/2020	Team
	Write	12/9/2020	12/15/2020	Allison McDonald
	Review	12/13/2020	12/14/2020	Team

8.3 GitLab Issues List

Issue ID	Title	Assignee	Created	Closed
1	If a User is new, when they navigate to the homepage, then they are prompted with a login screen	Ryan Austin	11/2/2020	11/2/2020
2	Write User Stories for User Account Actions	Christina Reiss	11/4/2020	11/4/2020
3	Set up Mongo Database	Ryan Austin	11/4/2020	11/9/2020
4	Set up SQL Database for User Authentication	Ryan Austin	11/4/2020	11/9/2020
5	Write User Stories for Profile Editing	Moe Rantala	11/4/2020	11/9/2020
6	Write User Stories for Login and Logout	Allison McDonald	11/4/2020	11/11/2020
7	Write User Stories for Visit Management	Anas Abdulrazzaq	11/4/2020	11/9/2020
8	Write User Stories for Account Creation	Christina Reiss	11/4/2020	11/9/2020
9	Write Test Data	Ryan Austin, Allison McDonald, Christina Reiss, Anas Abdulrazzaq, Moe Rantala	11/5/2020	12/3/2020
10	Compile User Stories for User Guide by 6:30pm ET Saturday	Allison McDonald	11/5/2020	11/11/2020
11	Set up Jacoco for Code Coverage	Ryan Austin	11/7/2020	11/11/2020

Issue ID	Title	Assignee	Created	Closed
12	Everyone can compile, build, and run the application in their local environment	Ryan Austin, Allison McDonald, Christina Reiss, Anas Abdulrazzaq, Moe Rantala	11/7/2020	11/25/2020
13	Write Design Guide/Document for turning in	Allison McDonald, Moe Rantala	11/7/2020	11/18/2020
14	In Java, Develop state objects for an Owner's Profile	Allison McDonald, Christina Reiss	11/7/2020	11/23/2020
15	In Java, develop State Objects for a Dog's Profile	Allison McDonald	11/7/2020	11/23/2020
16	In Java, develop State Objects for a Park's Profile	Anas Abdulrazzaq	11/7/2020	12/3/2020
17	In Java, design the Calendar Object Model on the back end	Ryan Austin	11/7/2020	12/1/2020
18	Refactor Spring Security to use Password Authentication via SQL Database	Ryan Austin	11/7/2020	11/26/2020
19	On the back-end, develop the Sign-up protocol to the BasicUserService	Christina Reiss	11/7/2020	11/23/2020
20	On the back-end, develop the Sign-up endpoint in the UserController	Christina Reiss	11/7/2020	11/26/2020
21	Determine General Layout for User Interface (GUI)	Ryan Austin, Moe Rantala	11/7/2020	11/18/2020
22	Write Phase I Report	Allison McDonald	11/7/2020	11/25/2020
23	Write Phase II Report	Allison McDonald	11/7/2020	12/3/2020
24	Write Phase III Report	Anas Abdulrazzaq	11/7/2020	12/9/2020
25	For Dog Management Controller, develop all required endpoints	Ryan Austin, Christina Reiss	11/7/2020	12/3/2020
26	Design and implement a Dog Management Service	Christina Reiss	11/7/2020	11/26/2020
27	Design or implement an existing Calendar widget for the Front-End Web App	Moe Rantala	11/7/2020	12/3/2020
28	Develop a Visit Controller with all required endpoints	Ryan Austin	11/7/2020	12/3/2020
29	Develop a Calendar Controller with all Required Endpoints	Ryan Austin	11/7/2020	11/18/2020
30	Develop a Calendar Service that conforms Time Data and Visit Data into a Calendar Object	Ryan Austin	11/7/2020	12/3/2020
31	Integrate Calendar Object Model with the front end and conform to Calendar Widget	Moe Rantala	11/7/2020	12/13/2020
32	Develop the Update Profile Use Case within the UserController and the BasicUserService	Ryan Austin	11/7/2020	12/1/2020

Issue ID	Title	Assignee	Created	Closed
33	Write UML Diagrams for Park/Calendar Services	Ryan Austin, Anas Abdulrazzaq	11/11/2020	11/14/2020
34	Write UML Diagrams for User/Dog Related Services	Allison McDonald, Christina Reiss	11/11/2020	11/18/2020
35	Develop Visual Diagrams for User Interface (Templates)	Moe Rantala	11/11/2020	11/18/2020
36	Setting up Cucumber	Ryan Austin	11/18/2020	11/22/2020
37	On the Front End, Create a Sign-up Form and Page for new Users	Moe Rantala	11/19/2020	11/25/2020
38	Create a Dog Service that handles a Dog's Profile and Information	Allison McDonald	11/19/2020	11/25/2020
39	Adding Multiple Dogs on the UI's Sign-Up Page	Moe Rantala	11/26/2020	12/1/2020
40	Page to Edit/Update a Profile	Moe Rantala	11/26/2020	12/3/2020
41	Research a Widget for Calendar for UI	Moe Rantala	11/26/2020	12/1/2020
42	Implement Endpoint to add a Dog to a User's Profile	Allison McDonald	11/26/2020	11/29/2020
43	Implement ScheduleVisit and UnscheduleVisit in ParkCalendarService	Anas Abdulrazzaq	11/26/2020	12/1/2020
44	Implement AddVisit and RemoveVisit inside of BarkPark class	Anas Abdulrazzaq	11/26/2020	12/1/2020
45	Write E2E test for Signing Up	Ryan Austin	11/26/2020	11/29/2020
46	Write E2E Tests for Login	Allison McDonald	11/26/2020	12/12/2020
47	Write all other E2E Tests for Sign Up	Christina Reiss	11/26/2020	12/7/2020
48	Develop Endpoint and Functionality to Update a Profile	Christina Reiss	11/26/2020	11/29/2020
49	Wire the Calendar Scheduling Endpoints to the ParkCalendarService	Christina Reiss	11/26/2020	11/30/2020
50	Create Endpoints (No Functionality) for Dog management inside of the UserController	Christina Reiss	11/26/2020	12/3/2020
51	Implement Method in Dog Service to Edit Dogs	Allison McDonald	12/3/2020	12/9/2020
52	STRETCH: Implement Functionality to Remove a User (Cancel Account)	Ryan Austin	12/3/2020	12/12/2020
53	Implement Edit Visit in the CalendarController	Christina Reiss	12/3/2020	12/7/2020
54	Web Page: In User Profile Edit, we need to be able to Manage Dogs	Ryan Austin, Moe Rantala	12/3/2020	12/12/2020
55	Web Page: In Calendar Page, We need to be able to access a Calendar for a specific Park	Moe Rantala	12/3/2020	12/7/2020
56	Calendar Component: In the Park's Calendar, we need to be able to Schedule, Unschedule, and Edit Visits	Ryan Austin, Moe Rantala	12/3/2020	12/14/2020

Issue ID	Title	Assignee	Created	Closed
57	Navbar: Make all pages accessible from the Navbar	Ryan Austin, Moe Rantala	12/3/2020	12/8/2020
58	Implement Endpoint for Deleting a Dog	Allison McDonald	12/3/2020	12/9/2020
59	Catch Up Unit Tests to 50% coverage	Allison McDonald, Anas Abdulrazzaq	12/3/2020	12/12/2020
60	Write Final Report	Allison McDonald, Anas Abdulrazzaq	12/9/2020	12/15/2020
61	Ask About The Actual Due Date/Time	Allison McDonald	12/9/2020	12/11/2020
62	Cucumber Tests for Adding a Dog (E2E)	Christina Reiss	12/9/2020	12/13/2020
63	Cucumber Tests for Editing a Dog (E2E)	Christina Reiss	12/9/2020	12/14/2020
64	Cucumber Test for Deleting a Dog (E2E)	Christina Reiss	12/9/2020	12/14/2020
65	Cucumber Test for Viewing Dog (E2E)	Christina Reiss	12/9/2020	12/14/2020
66	Submit Lessons learned to Ally	Ryan Austin, Allison McDonald, Christina Reiss, Anas Abdulrazzaq, Moe Rantala	12/10/2020	12/15/2020
67	Peer Reviews for Every Member of the Team	Ryan Austin, Allison McDonald, Christina Reiss, Anas Abdulrazzaq, Moe Rantala	12/10/2020	12/15/2020

9. Conclusions

9.1 Lessons Learned

9.1.1 Anas Abdulrazzaq

Group one has worked through several significant roadblocks in the development of bark park online website that is designed to allow users can update a calendar to show when they plan to visit. The group had worked together in every software development cycle including project plan, test plan, project design, development, and testing. The development part was split into three phases. The overall goal of the Group One project was a success, and every member are instrumental in finalizing the product. The code is working effectively and as expected. We were able to learn a lot from this lesson. One of the things we learned was that a system like this takes time to develop well. We did however play on each other's strengths which are why our project turned out so well and why we as a team did not have many disagreements. We all brought up new ideas and had others cross-check our ideas with each other to ensure we are producing the best product we can. Limiting scope creep early on helped keep the project on track throughout and the Team Foundation Server worked well for free source control. Overall, I believe that we did an amazing job with our project.

9.1.2 Ryan Austin

I learned some of the best practices in regards to authentication and web server security. Initially, I struggled to integrate username-password authentication into the Spring web server since we were using Angular as a front end. The struggle was that while in development mode, the browser was not holding onto the authentication

credentials required to stay logged onto the server, as it does in production mode. This forced me to use the browser's local storage in order to store credentials that can be reused so that while developing we would not have to login every time that the page reloads. In the future, I would like to setup a reverse proxy on the Spring web server so that the web pages can be proxied from the Angular Node server through the Spring web server and on to the client's browser. This way, the browser will store the credentials and allow me to continue developing without having to store credentials insecurely.

In regards to the production process, I feel that our team was fairly organized in using GitLab for both issue tracking and version control. However, it would have been better to have used the Agile tools that come with Jira or Gitlab so that we could have better estimated workloads and time spent on particular tasks. Without using an Agile issue tracking service, we were not able to accurately predict the time requirements for a lot of tasks which may have made the workload imbalanced for team members.

Our team did an excellent job at establishing roles right off the bat. I feel as if most of the team members were able to contribute according to their strengths. One thing that I learned, however, is that even though roles and swim lanes are great for taking advantage of developers' skill sets, there should also be back-ups for when one stack becomes overwhelmed. For example, the front-end development and the testing parts of our application were falling behind towards Phases 3 and 4. This forced us to pivot and have back-end developers shift to the testing/front-end tasks that were backlogged.

9.1.3 Allison McDonald

I learned I still have much more to learn. There are so many technologies I have yet to experience. During the online computer science program, you create your own club that has one member. You gather a few items that make the club feel more comfortable and you are the club ruler.

This class forced me out of my club. At first, it felt overwhelming. Not only were there new technologies, but I also had a responsibility to a team. Fortunately, I had great team members. We were able to organize the development with the tremendous help and effort from Ryan. I feel my ramp up to using the technologies and the understanding of the system architecture was slower than I would have liked. However, after going through this process and working on this team my knowledge has increased immensely.

9.1.4 Raymond "Moe" Rantala

Biggest lesson learned was about Angular. I opted to work on the front end (exclusively) with Ryan and ended up having to learn the Angular style and TypeScript from scratch, as well as quickly get up to speed on higher level HTML skills than I was used to. Time management was crucial, and I failed at that pretty miserably in the beginning but started to get better with juggling real life/full time job and the project until the last few weeks where I hit my deadlines (with Ryan's expert help) and managed to contribute a set of meaningful and well-crafted front-end features, weekly. Turns out we use a lot of React on the contract I will be working on at the org I currently work for, so angular is a pretty decent base, so I picked something good to do even if it were more difficult than just working on the back end would have been for me. I learned a lot more than I thought I was going to, and that's good stuff.

9.1.5 Christina Reiss

- Having a strong team lead is important.
- Good documentation goes a long way so that code can be understood, and problems can be more quickly addressed.

- Understand and fully utilizing git is important for software projects. Having the ability to stash code and reference old code is important in the development process and I have not had a project that fully implemented git in the way this one did so using git and git lab in a team has been very insightful.
- Test driven development leads to better code and having a balanced test coverage can help test the application without having to adjust too much when changes need to be made.

9.2 Design Strengths

A core aspect that was decided early in development was the requirement for testing. The goal was to complete all End-to-End Automated use case testing and to obtain above 50% test coverage with Unit Tests. The objective of the testing goal was to ensure the application was not brittle and that features, and implementations did not regress over time. The original End-to-End testing plan contained 35 use cases. During development and testing, five use cases were found to be redundant as they were already part of other use cases. One use case was determined to be too time consuming to write a proper automated test given the limited timeframe. However, the functionality of the use case was still verified. A total of 29 use cases were fully automated and passed testing. The completed project also obtained 54% test coverage with Unit Tests. With these results, we consider our testing goal achieved; Therefore, ensuring the future functionality of the application.

9.3 Limitations

There were two major limitations to development – time and experience. With the course consisting of eight weeks and limited development experience amongst most team members, it was imperative to keep the application's functions sensible and to utilize the proper development tools.

9.4 Future Suggestions

The application has plenty of room for growth and enhancement. Additional features such as searching and filtering capabilities at the park and calendar levels would improve functionality and the user's experience. Providing additional park and dog information would allow the user's to be more informed about the atmosphere of their visit. Of course, this type of application would be well-suited for mobile development.