

CS 320 Course Project Final Report

for

Animal Database

Prepared by

Group Name: Team 20

|  |  |  |
| --- | --- | --- |
| Vinh Duong | 11613642 | vinh.duong@wsu.edu |
| Seth Lanante | 11458512 | seth.lanante@wsu.edu |
| Yevgeniy Diriyenko | 11679399 | Yevgeniy.diriyenko@wsu.edu |
| John Stevenson | 11639605 | john.stevenson@wsu.edu |

|  |  |
| --- | --- |
|  |  |
| Date: | December 16, 2020 |
|  |  |
|  |  |
|  |  |

Content

Contents ii

1 Introduction 1

1.1 Project Overview 1

1.2 Definitions, Acronyms and Abbreviations 1

1.3 References and Acknowledgments 1

2 Design 2

2.1 System Modeling 2

2.2 Interface Design 2

3 Implementation 3

3.1 Development Environment 3

3.2 Task Distribution 3

3.3 Challenges 3

4 Testing 4

4.1 Testing Plan 4

4.2 Tests for Functional Requirements 4

4.3 Tests for Non-functional Requirements 4

4.4 Hardware and Software Requirements 4

5 Analysis 5

6 Conclusion 6

Appendix A - Group Log 7

# Introduction

## Project Overview

This project is an animal database that a user can either make an account to log in to interact with the animals in the database. When logged in the user will have a page to create an animal and the user can view animals in the database. The logged in user will also be able to play with and feed the animals.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| SRS | Software Requirements Specification |
| Database | Where the information for animals will be stored |
| UML | Unified Modeling Language, used to create visuals of how systems works.. |
| Account | How the user’s information will be stored. |

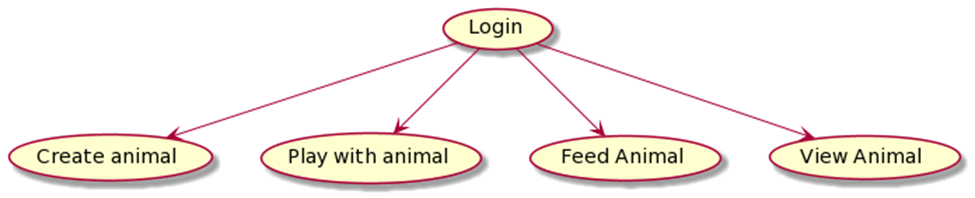
## References and Acknowledgments

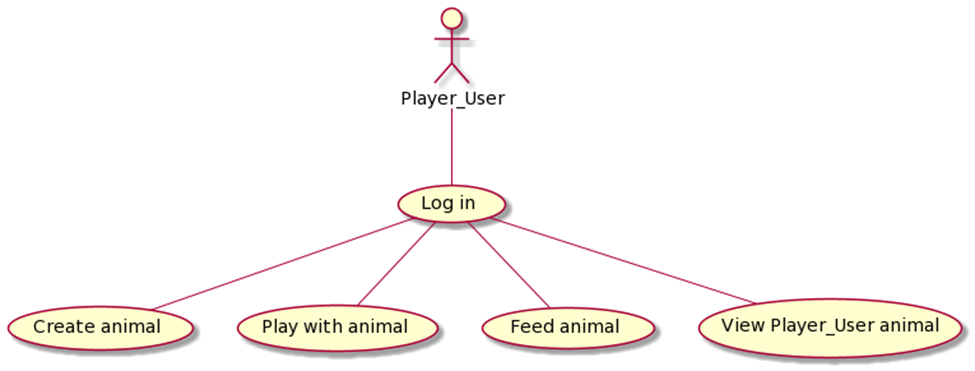
Used the Lecture 13 – System Modeling Part 2 slides for the definitions of the diagrams. Other than that, we did not cite or paraphrase any other source to the best of our knowledge for this documentation.

# Design

## System Modeling

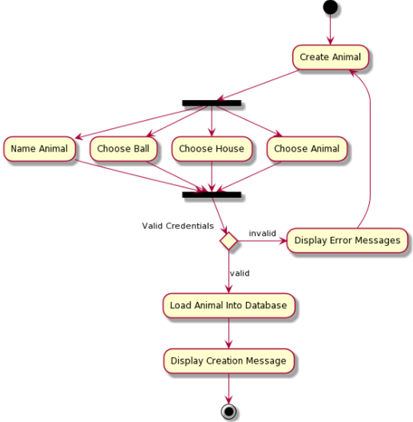
(Context Diagram)

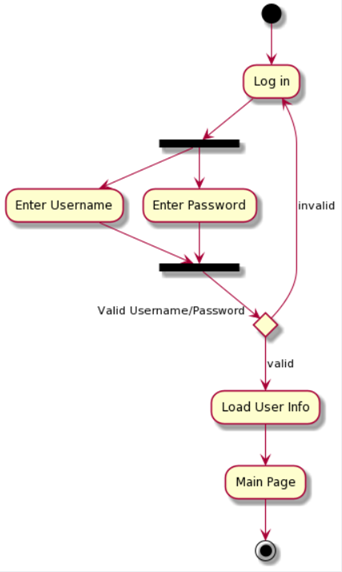
(Data Flow Diagram)

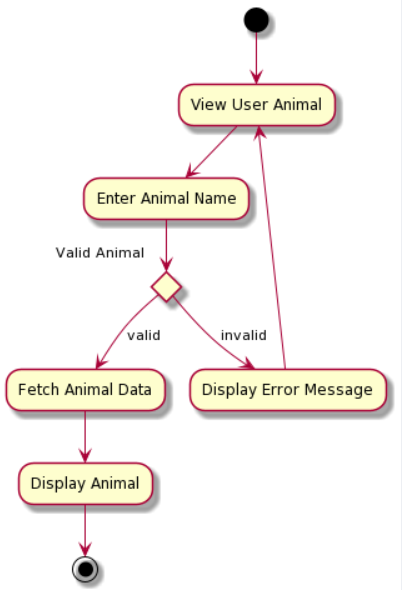


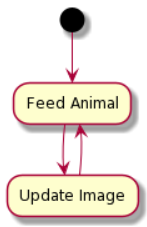
(Use Case Diagram)

Activity Diagrams

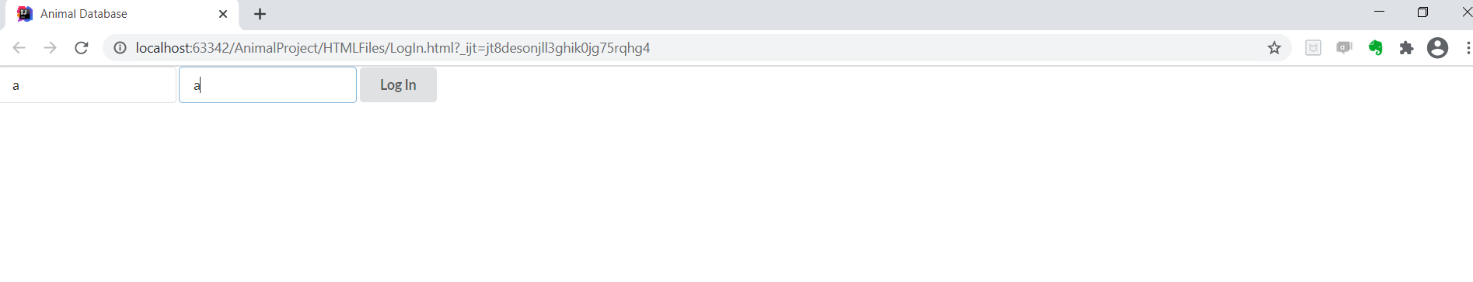


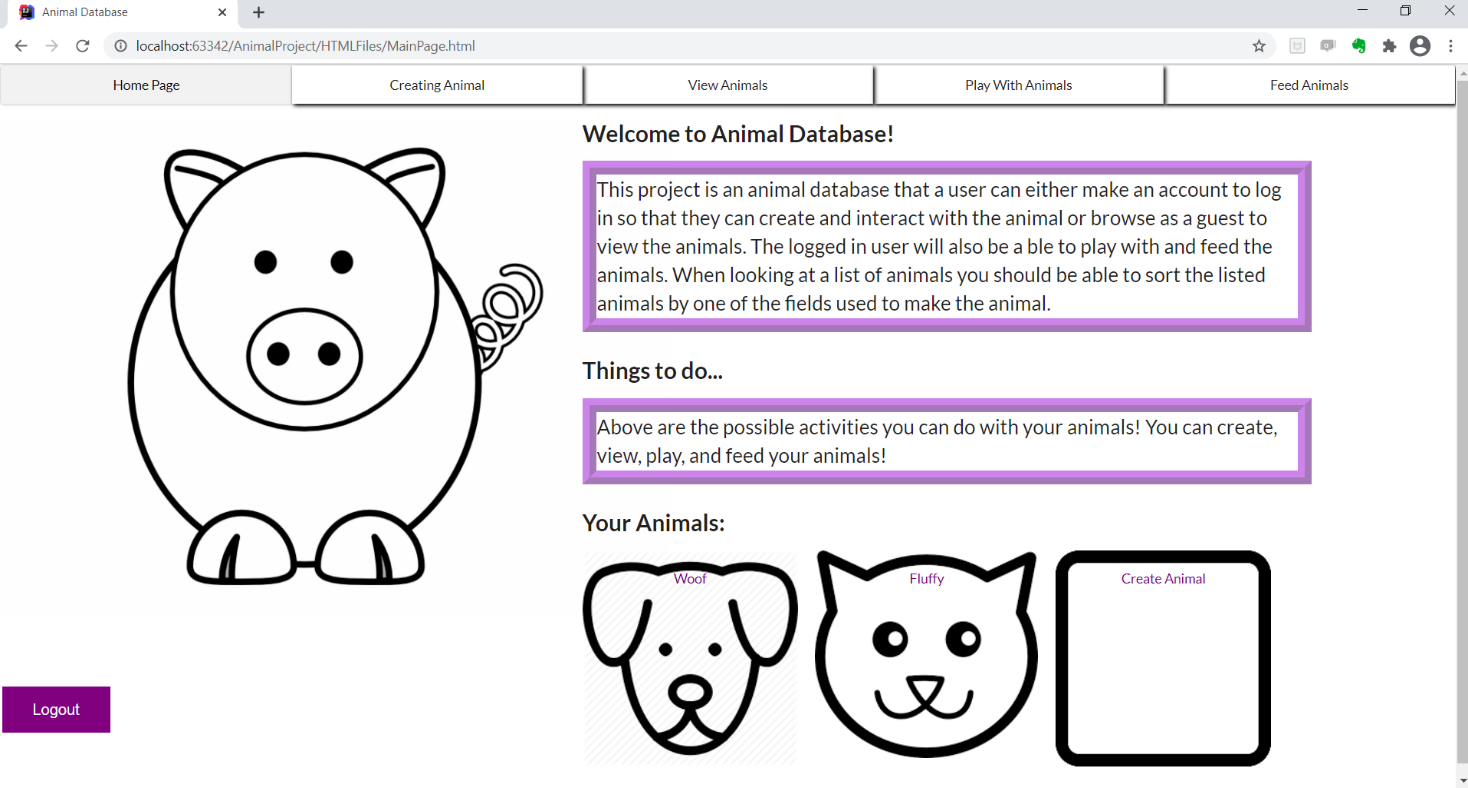


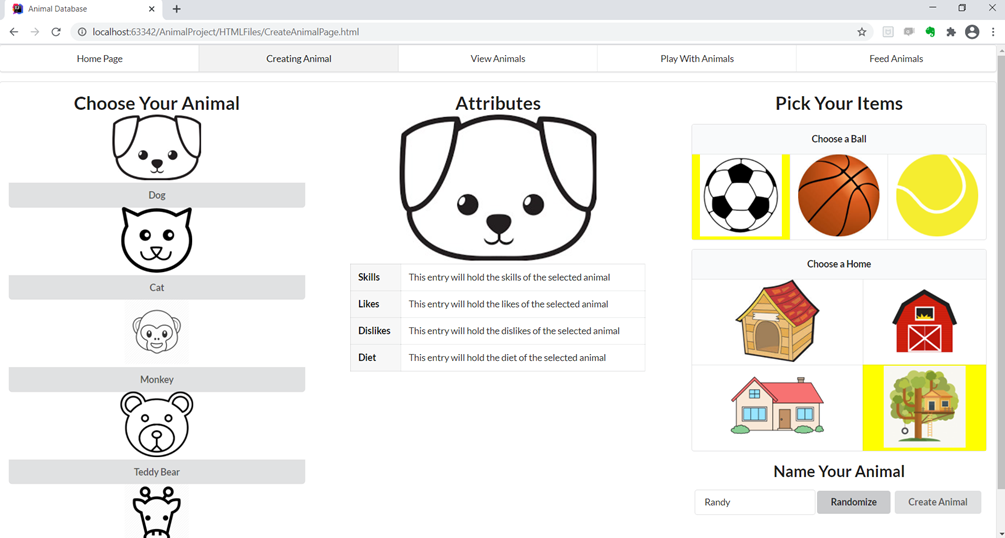


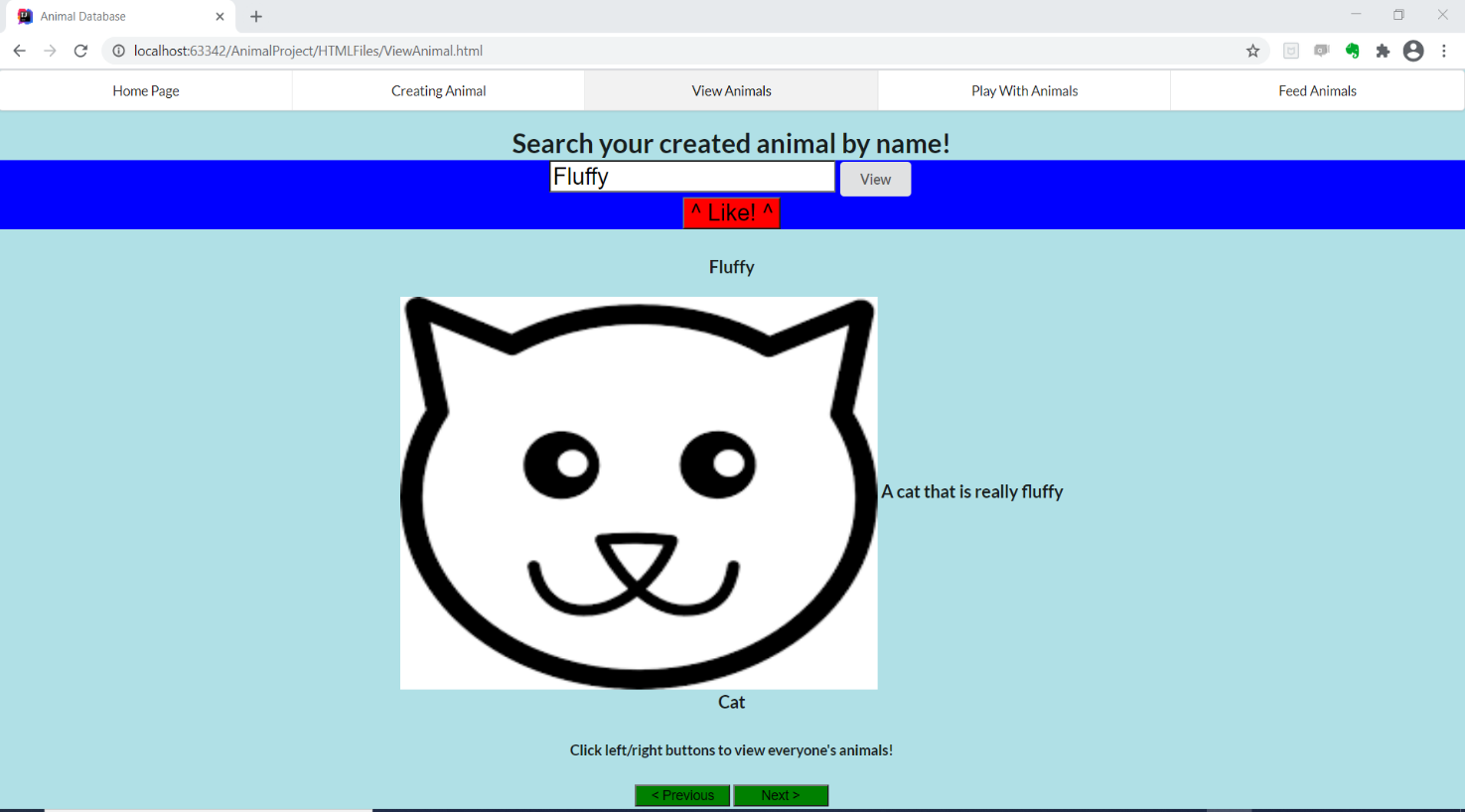


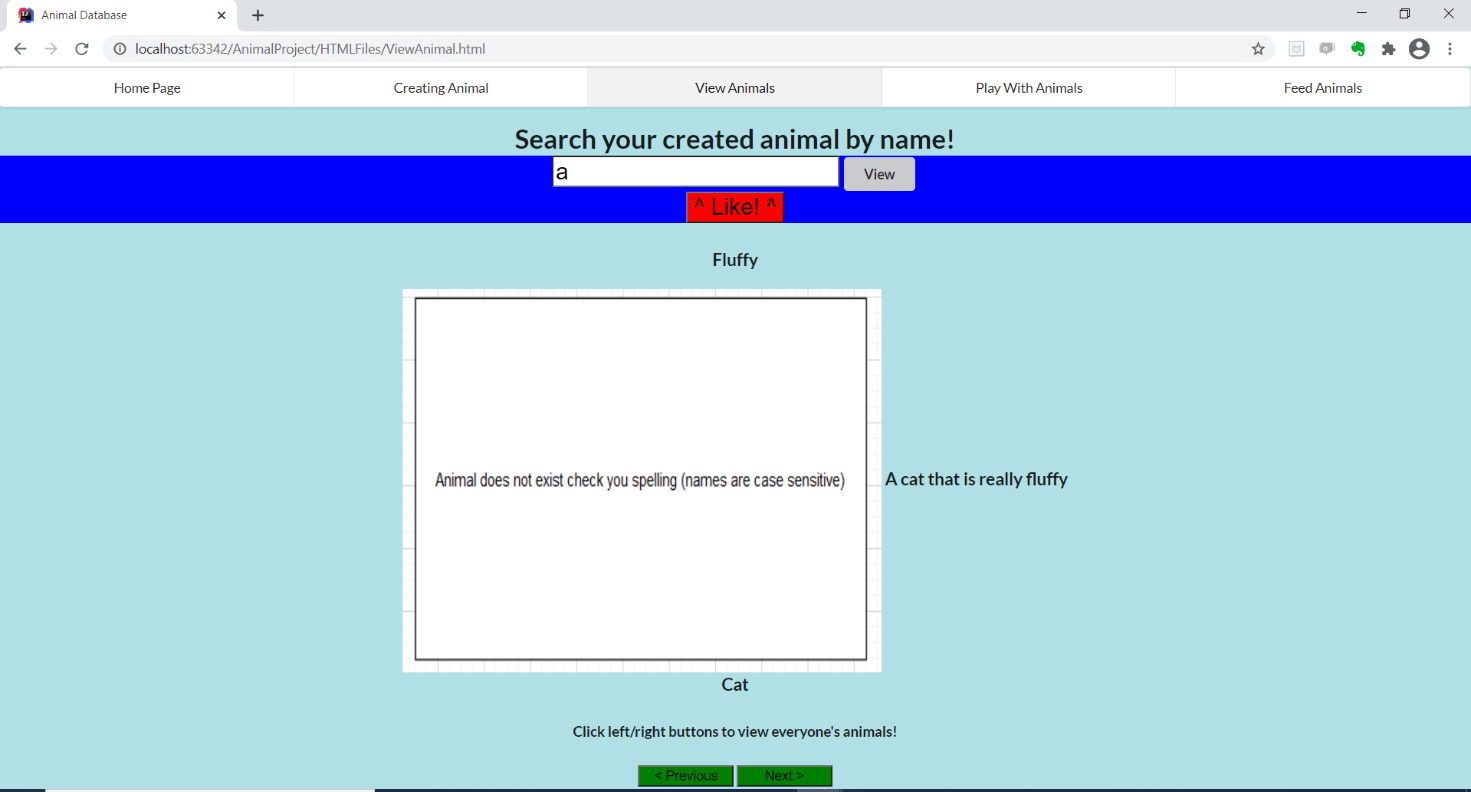
## Interface Design

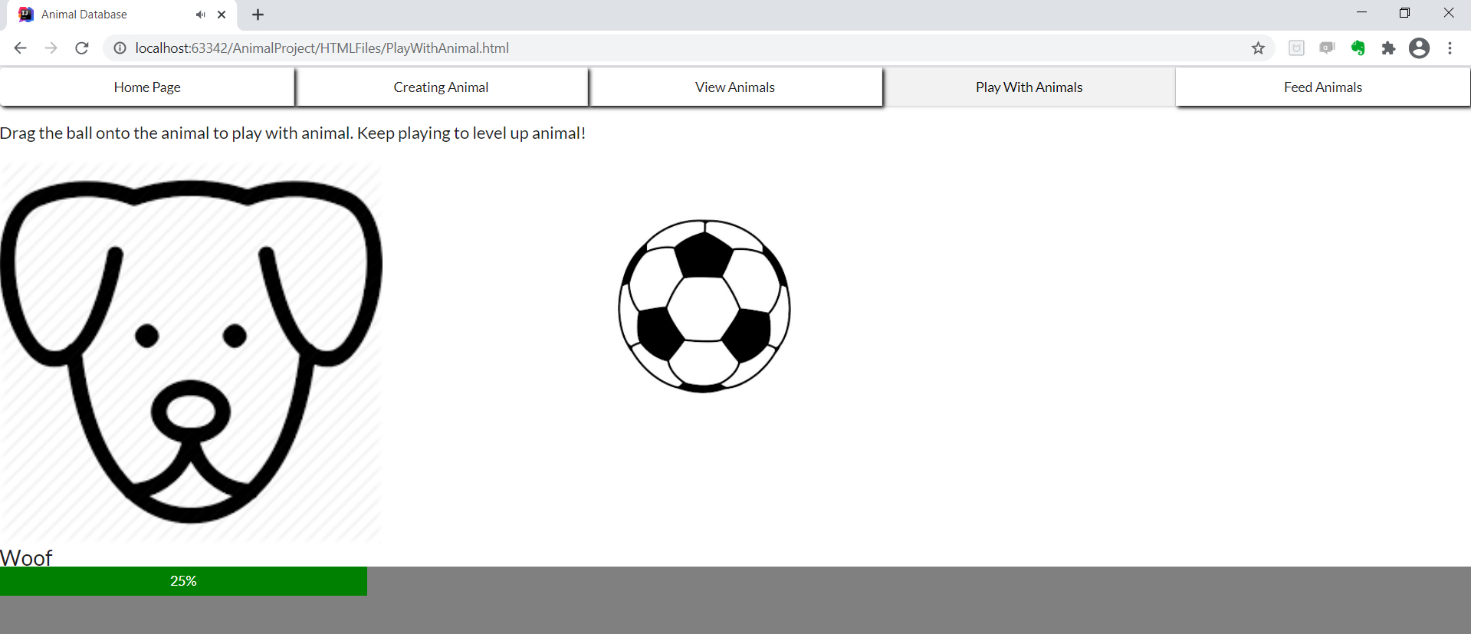
Login Page Screenshot:

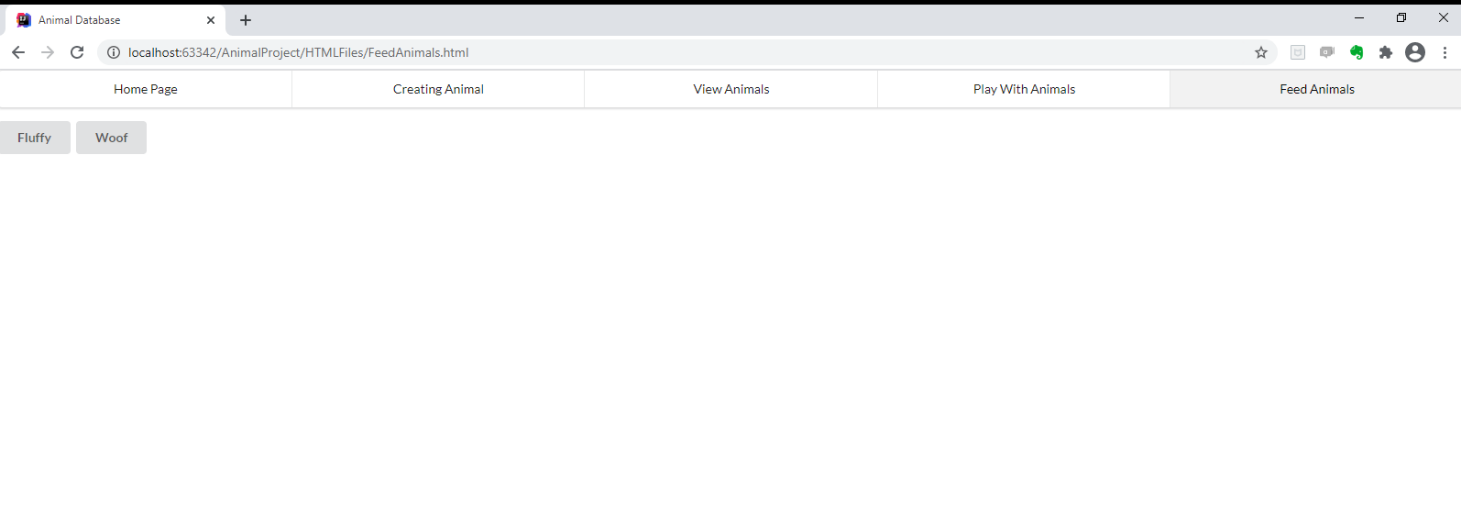
Home Page screenshot:

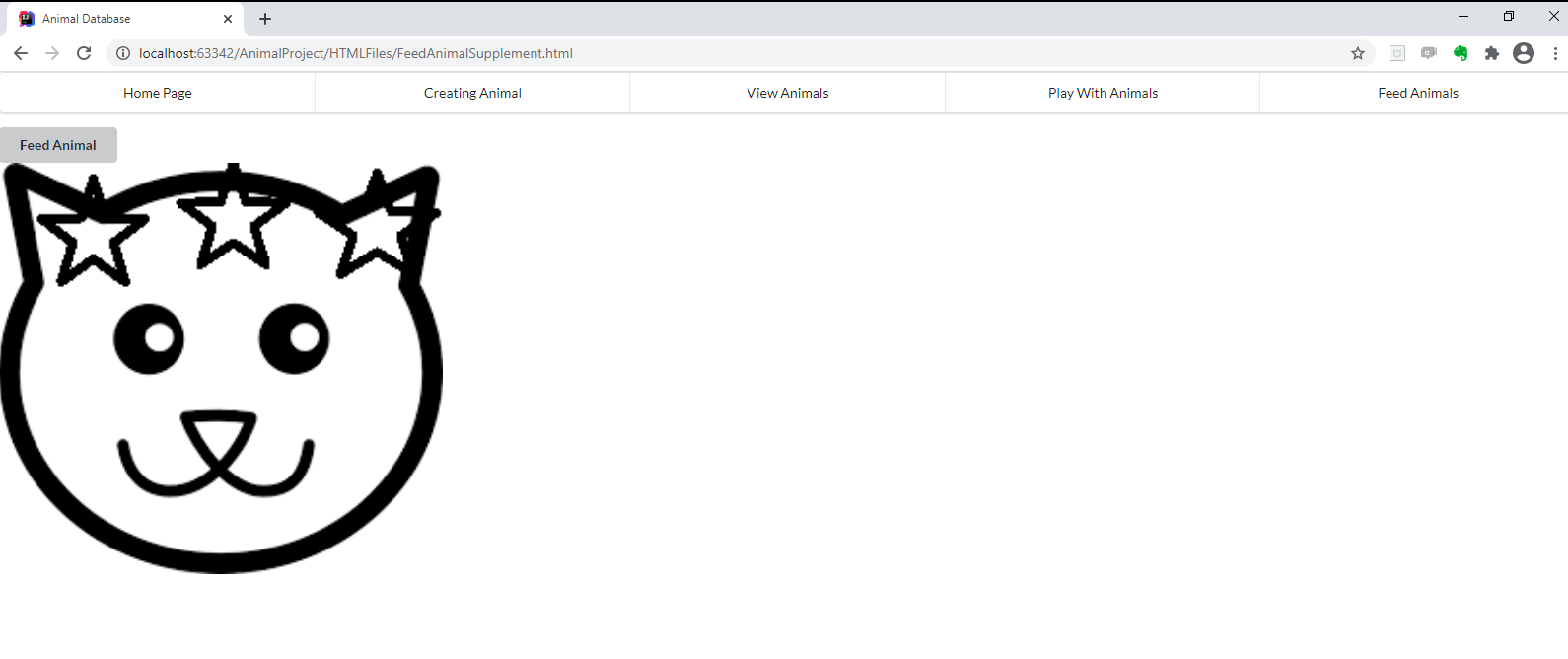
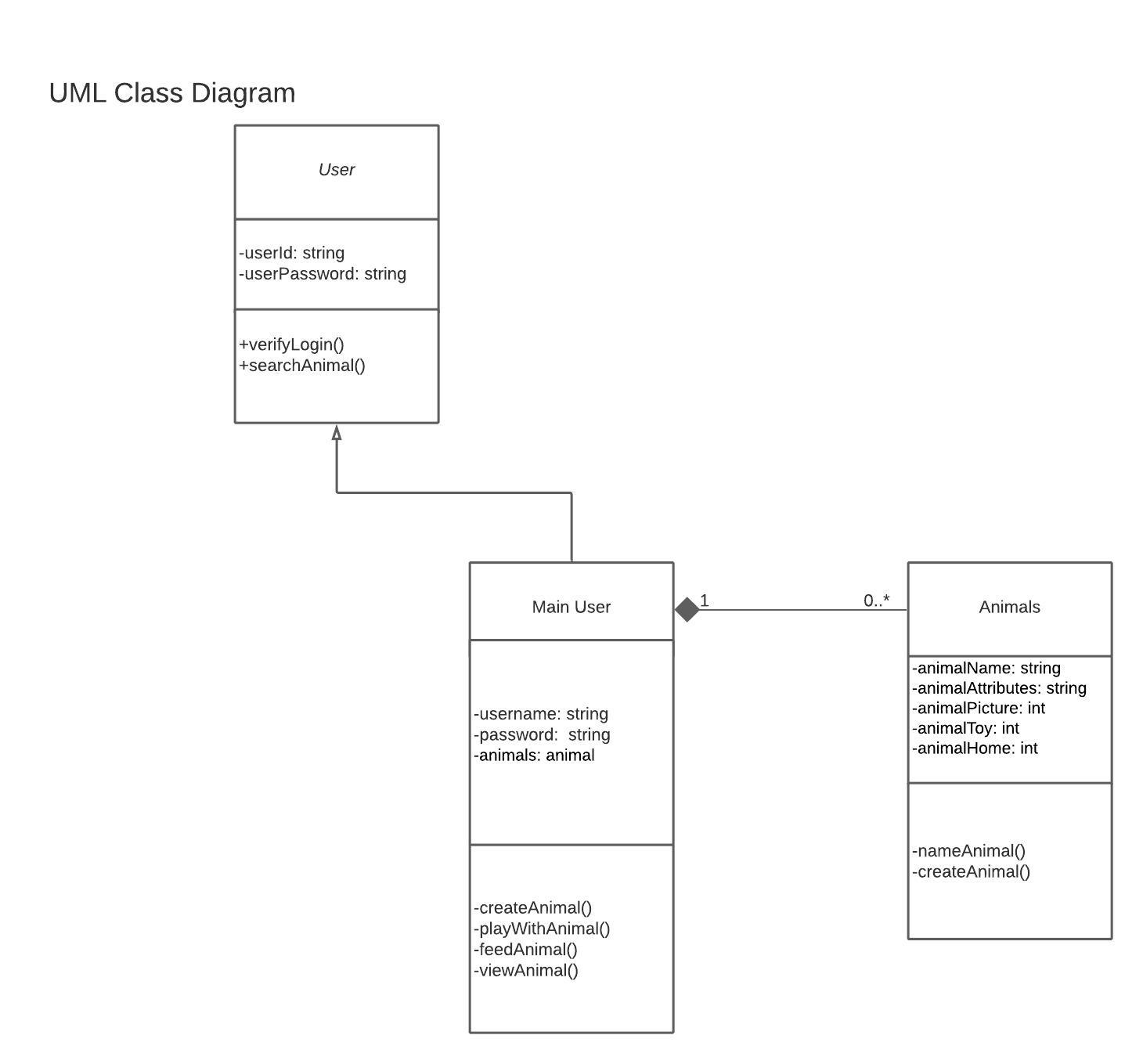
Create Animal Screenshot:

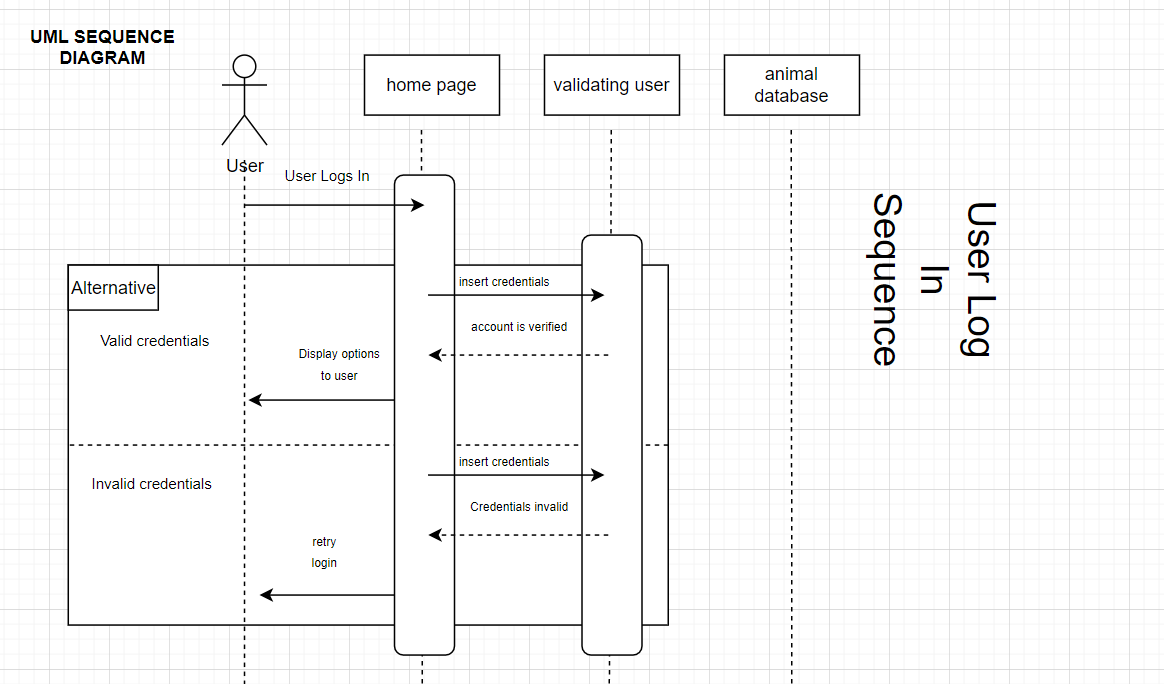
View Animal Screenshots:

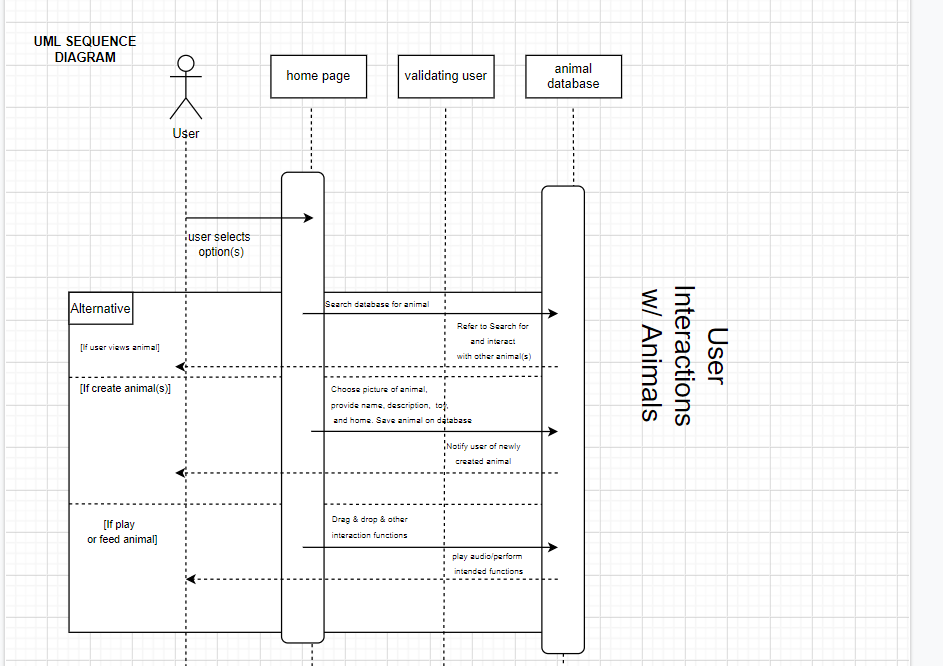


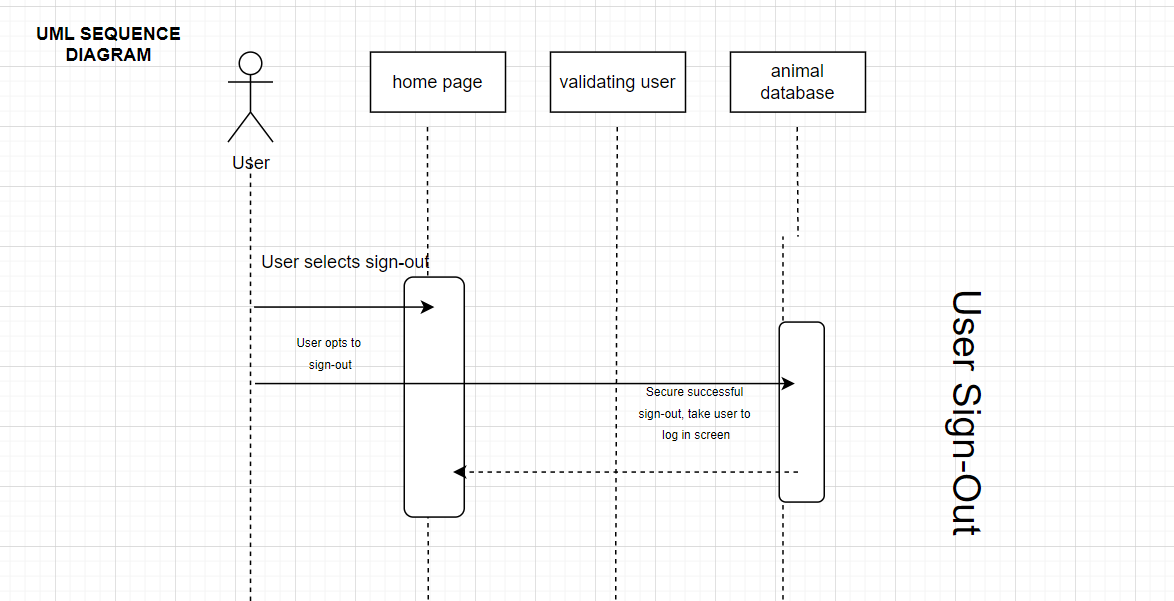
Play With Animal Screenshot:

Feed Animal Screenshot:









# Implementation

## Development Environment

The programming languages we used was JavaScript and HTML. We used the IntelliJ IDE with ESLint. For HTML framework we used some of frameworks from Semanticsui.

## Task Distribution

Vinh: Did most of the original SRS document and edited for final submission. Did section one of milestone 2. Did most of the login page and did most of Feed Animal page for the animal database. Worked on the Final Report document.

*Design and implementation of Create Animal Page -> John Stevenson*

*Implementation of animal database and temporary storage -> John Stevenson*

*Design of activity diagrams -> John Stevenson*

*Design and implementation of Main page and Play with Animal page -> Seth Lanante*

*Initial layout of software testing -> Seth Lanante*

*Co-design of class diagrams -> Seth Lanante*

*Implementation of final ideas and editing for original SRS doc -> Yevgeniy Diriyenko*

*Design and implantation of View Animal page -> Yevgeniy Diriyenko*

*Design of sequence diagrams -> Yevgeniy Diriyenko*

*Co-design of class diagrams -> Yevgeniy Diriyenko*

*Designed data structure with tracking system for viewing -> Yevgeniy Diriyenko*

## Challenges

Two of our group members were not able to view the final versions of the play with animal and feed animal pages. One of them was able to fix this by switching from Google Chrome to Microsoft Edge when running the html code. We think this may be linked to how cookies are handled in the browser, but we aren’t certain.

Finding a way to store animals took a bit of researching. Most methods we found were either difficult to implement or didn’t quite fit our requirements. For instance, using a file to store long-term information proved to be impossible when running in a browser. We settled on localStorage and sessionStorage as long-term and temporary storage for the app. This allowed us to not only keep track of information across sessions, but we could load session-specific data into temporary storage to easily interface with our stored data (load a user’s animals into temporary storage).

# Testing

## <*This section is a summary of your testing report>*

## Testing Plan

<Describe your testing plan for the project.

TODO: Give a list of items or functions you want to test, and also a schedule for performing the testing. >

The testing plan for this project is to mainly test the web pages to make sure that everything is displaying properly and that available buttons function properly. The testing happens when we run the web page by making sure that any changes made does not affect what we already have working. Tests are done after most commits to the github repo.

## Tests for Functional Requirements

<Describe your test results for the functional requirements.

TODO: Provide a list of use cases or functions you have tested, as well as the testing results (whether or not the system passed the tests).>

For the log in page we tested by making sure that it will only proceed to the main page if the correct credentials was entered, which it did. For edge case test we put a limit of 20 characters for the username and 30 characters for the password so that any long strings will not be accepted. Additionally, we tested to make sure that empty strings are not accepted either. The result of this is if any string that is not a valid username and password is entered then the log in page will just reload.

For the feed animal testing we made sure that it was only displaying the animals that belonged to the current logged in user so we tested it by logging in with both user and seeing if feed animal display the correct animal names. We also tested to make sure that when we click on the animal that it displays the correct picture for the animal type and that as we click the feed button it progresses through the right sequence of photos. This is done for every animal and after testing it shows the correct results.

For the create animal page, we tested that the program would not create an animal if any and/or all fields were not selected. Input for the name should be tested for length extremes as no code prevents erratic behavior with an unexpected response.

For the database, we tested that the data pushed can be retained both temporarily and for long-term storage. The database retains information in long-term storage across sessions and deletes all temporary storage when a session is ended.

For View Animal we tested empty or incorrect strings to find animal(s). Then we tested to verify that the animal searched was the same animal that was found. After we tested the previous and next buttons to make sure no errors occur, and the animals are viewed in the correct order.

## Tests for Non-functional Requirements

<Similar to the Section 4.2, but this section is for the non-functional requirements. >

For Non-functional Requirements the only tests we had was for the speed of loading each page. We can click through every page after logging in and seeing if it takes too long to load. All the pages we have seen loads near instantly so it meets the general speed requirements.

## Hardware and Software Requirements

<Describe the hardware and software requirements for performing the tests. >

The hardware and software requirements we used for this project was a laptop that has google chrome installed and have enough memory and resources to run the animal database.

# Analysis

<In this Section you need to analyze the effort that has been put on this project.

TODO: Describe how many hours (approximately) each team member spent on the project, for each milestone, which milestone took the most effort and why. >

Vinh Duong: 2-3 hours for milestone 1. Less than an hour for milestone 2. 3-5 hours for milestone 3. Milestone 3 took the most effort because it took a lot of time to code and figure out how to do.

Seth Lanante: 1-2 hours for Milestone 1. 3-4 hours for milestone 2. 3-4 hours for Milestone 3. Learning how to work with data storage and testing html and javascript files was a difficulty for me, Milestone 3 definitely took most of my effort.

John Stevenson: Less than 1 hour for milestone 1. 2 hours for milestone 2. 10-13 hours for milestone 3. Milestone 3 took the most effort because it took a lot of trial and error to design the create animal page and code in html. It also took a lot to code the database/data retention and load in a user’s animals upon login.

Yevgeniy Diriyenko: 1-1.5 hours for milestone 1. 3-4 hours for milestone 2. 10-14 hours for milestone 3. Milestone 3 was the most difficult first of all due to the fact that I have never had any experience with html, secondly the entire portion of it required cooperation and editing somewhat of each others pages. So we had to rely on each others work throughout the entire portion. As well as having to format the data that I was presenting in the page in such a way that it made sense to the user, despite having zero experience in building or formatting a website. So in essence everything was learned and implemented on the spot.

# Conclusion

<Conclude the document with what you have learned through working on the project.>

We learned how to do formal specifications of a document. How to work as a coding team using the GitHub desktop app as version control. We learned how to work with JavaScript and HTML to build functional webpages.

Appendix A - Group Log

< Describe how frequently the group members meet during the semester, and how effective the communication is. This is optional for one-person projects.>

|  |  |
| --- | --- |
| 10/10/20 | Went over Git hub and tried to get the GitHub desktop app working. Talked about the SRS document. Agreed on a new meeting time. |
| 10/16/20 | Catch up on trying to get the git hub accounts working and worked on the SRS document. |
| 10/24/20 | Had to change communication method for a group member and added them to the git hub repository. |
| 11/6/20 | Final gathering before submitting SRS document, Final revision, edits, and submission. |
| 11/13/20 | Meeting to discuss how we will do project milestone 2 and then assigning the roles for the document. |
| 11/19/20 | Checkup about project milestone 2. |
| 11/21/20 | Finished the project milestone 2 document. |
| 12/01/20 | Try to set up the IntelliJ folder and get a basic web page working. |
| 12/04/20 | Member: Vinh and John. Do a quick check in on what we have so far and plans for moving forward. |
| 12/05/20 | Members: Vinh, John and Yev. Looked at what we have done so far and decided on what to do next. |
| 12/09/20 | Members: Vinh, Seth, Yev. Go over what we have and what we need to do. Specifically different use cases (what different users will do), log in page, how we will connect everything from page to page. |
| 12/11/20 | Members: Vinh, John and Seth. Review what we have and worked on the project during the zoom call. |
| 12/13/20 | Members: Vinh and Yev. Talked about the requirements Yev needs to do for the view animal page. |
| 12/14/20 | All member present. Group discussion on what we have and what we need to do. |
| 12/15/20 | Afternoon meeting Vinh, Yev, and John are present: John talked about an easier way to pass data around the web pages.  Night meeting all members present: Worked on making the presentation to present to the teacher. |
| 12/16/20 | Morning meeting all members present: Make sure everything is working for the presentation.  Night meeting all members present: Finalize everything and turn it in. |