**Lunas List**

**1. Project Title, and Group Name:**

**Project Title:** Luna’s List

**Group Name:** Software Solution Ltd.

**2. Project Website/Github URL:**

**Github URL:** [**https://github.com/Reiterpallasch/4882-Capstone**](https://github.com/Reiterpallasch/4882-Capstone)

**3. Version:**

**Version:** 4.0

**4. Project Summary:**

**4.1 High-Level Overview:**

Luna’s List is intended to be a platform in which canine lovers may discover and share locations that are dog friendly. In doing so, the platform will allow for the rules of the location to be well laid out, such as if aggressive dogs are allowed, or if a leash need be employed. The platform will also allow for those of the canine affinity to come together in varying ways, whether it be casual or formal. To sum it up simply, one may view it as an initially low-level canine lover’s social network with great potential for expansion. Initially it is to be web based, however, mobile expansion is within its grasp.

**4.2 Goals:**

The goal of this project is to bring information to dog owners in a unified, central location. Dog owners will be able to see tips and tricks on maintaining their dog’s health and behavior. Owners will also be add and see parks, vets, stores and restaurants that allow dogs to come as well as location rules, and a forum to discuss varying topics not covered elsewhere. Admins will be able to view and approved submitted information by users, or be able to add information themselves. Pages will also be able to display maps pinging relevant locations to the page that are near to the user on the page at the time.

**5. Problem Statement:**

**5.1 Original Customer Problem Statement:**

Dogs have been interwoven into our lives at a scale never seen before. They have moved outside of the perimeter of our homes and yards and into our roads, stores and entertainment areas.

More and more questions have flooded businesses, parks, and buildings on when, where and how can I bring my dog?

Lunas List will help answer these questions and more by providing a platform that provides communities with Dog friendly spaces.  It will also bring Dog Owners together in different ways.

**5.2 Expanded Problem Statement:**

Dogs have been interwoven into our lives at a scale never seen before. They have moved outside of the perimeter of our homes and yards and into our roads, stores and entertainment areas. As more dog owners seek to bring their dogs out on more than just walks and car rides: businesses, parks, and buildings have begun to reciprocate by offering themselves as dog friendly places. However, although this is occurring, questions have begun to arrive from these dog owners about when, where, and how they can bring their dogs. These questions that grow more and more make it inconvenient for dog owners to be able to always stay updated on where they can go. This is where Luna’s List comes in. Luna’s List strives to answer the questions that dog owners have by creating a platform that tells its users about upcoming events, dog friendly businesses, and provide tips that will better enable dog owners to prepare them and their dogs to go out to different places. Luna’s List will let dog owners connect with each other and participate in providing information about events and businesses that surround them that are dog friendly and among another range of items will serve to provide: a forum where dog owners can discuss dog habits, doggy trainers that they can go to, an intermediate for adopting dogs or puppies, and other general discussions. Luna’s List will also provide general dog tips such as care, etiquette, and news.

**6. Team Profile:**

**6.1. Team Member Profile:**

Wesley Jones:

My goal is to implement software design and programming into two additional degrees that I posses in Biomedical engineering. I wish to apply lessons learned into building great team foundations, and working towards improving products such as healthcare monitoring devices and prosthetic interfacing. My primary function on this project has been implementing major backend website functions that allow for data addition and display, as well as administrator control. I have also served as Scrum Master, keeping meetings on focus and assisting with team issues that present themselves during the project design and implementation. My major strengths are quickly understanding a problem and working to find a viable solution, as well as being able to quickly pick up a programming language.

Larz Leonard:

My goals in starting a career is to program and eventually to get into game development and I am very interested in programming AI. For this project, my role is a developer and my strengths are programming in Java, Python, HTML, and Ruby, with Ruby being the primary focus behind the coding here. Another aspect of my strengths is brainstorming different ideas for how to tackle a problem to determine what will work and what will not.

Thi Phan:

My strengths revolve around development in Python, Java. I know some about Web development with Django, Ruby on Rails and iOS development with Swift. For this project, I will work on the back end and some front end aspects.

Vinh Tran:

My main understanding in programming is Java and C/C++ with a minor amount of knowledge in SQL. In this project my role serves as a developer, helping to formulate the issues/tasks that the team requires to build a successful product.

Noah Hanks:

My career goals are less focused on programming and more concentrated on other aspects of the information technology field. More specifically, I would like to pursue a career either in systems administration or possibly digital forensics. Lately, my interests have primarily revolved around building out a “homelab” which is a hobby focused on setting up and using enterprise hardware and software in a home network. While my career goals may not be programming focused, I have experience coding in the following languages: Python, C++, SQL, and Ruby on Rails. In this project, I am filling the roles of developer and product owner. As the product owner, I am responsible for handling all communication with the customer and making sure that this information is conveyed correctly to the rest of the team. As a developer, I am responsible for completing assigned tasks on-time and to the best of my ability.

**6.2 Roles and Responsibilities:**

**Scrum Master:**

Wesley Jonesis our scrum master. He is a keeper of scrum process. Wesley is tasked with leading daily scrum and running the scrum planning meeting. In doing so, it is ensured that the topics stay on track and important issues are discussed as necessary.

**Product Owner:**

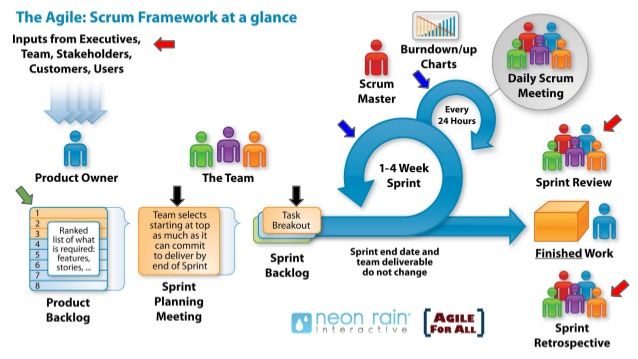
Noah Hanks is Product Owner. He is the key member to communicate with customer and responsible for writing user stories or help others to write user stories.

**Developers:**

Every member has a developer role. Each Team Member is typically in charge of developing a new feature for the website and make sure their codes work.

**7. Development Process:**

**7.1 Agile Scrum Process:**

****

***Figure 1: Agile Scrum Process***

The Agile Scrum method is being implemented to perform this project. Thus far, Sprint planning has consisted of determining what is to be accomplished during the sprint, and scrum meetings used to discuss where the team was in finishing the tasks as well as discuss concerns. We initially began with brainstorming on our product backlog, creating a list of both functional and non-functional user stories that we felt embodied the scope of the project. Once we had our basis, we broke these stories down further into smaller components. Presenting the overall plan to the customer and understanding his needs, we then broke down the current backlog and selected what was needed that would both fit our timeline effectively and match the customer needs. Values to the user stories were determined via a consensus to the effort required for completion. We also decided that we would meet with the customer before and after each sprint, and hold scrum meetings twice a week, with external communication available if needed. Each scrum meeting consisted of ensuring that we understood where we were in regards to completion of tasks, or discuss changes that needed to be implemented to complete a task. We also provided assistance to one another if it was needed. We ended with a retrospective to determine what went well, as well as what we can do better, which is detailed in section 9.

**7.2:**

For our website, we made use of Ruby 2.5.1, Rails 5.2.3, client-side JavaScript, and HTML/CSS. For CSS, we’re additionally using Bootstrap to create a responsive, clean design. We use sqlite3 for database management as it is portable, easy to manage, and straightforward in making changes or pulling information.

**7.3 Communication Method:**

Communication consists of Scrum meetings, Slack channels, Email, and Github. These platforms were felt best to effectively and consistently maintain communication flowing in a manner that all team members are up to date. Slack particularly allowed team members to communicate outside of scrum meeting and implement a system to alert users when a commit was made and pushed to Github.

**8. Glossary of Terms:**

Mobile – Platform description for phones and tablets.

Model - Database table for storing values.

Admin - System administrators capable of reviewing, editing, or deleting added table values.

Routes - File containing paths to pages necessary to reach each page of the website

Controller - File containing primary ruby code that controls how a page functions and calls the associated model for use. It also defines what values are permitted to be added to each database model.

Views - Sub-folders containing website pages. A hierarchical structure allowing for multiple pages of similar name to be implemented and defined in the routes file without conflicting naming schemes.

Bootstrap - An API that allows for graphical enhancement of website pages and layouts.

Gem - A packaged ruby API containing code to allow for functions and features to be implemented.

GUI - Graphical User Interface.

Home - The central location from which the rest of the website structure may be accessed.

Sitemap - A hand drawn description of the website that shows how to reach and page, and the pages primary function.

Index - A page dedicated to displaying all model information. Essentially lists entries in the model.

Registration - The page dedicated to creating a user.

User - A standard person who wishes to have an account on the website and the ability to add information to the models for approval.

UML Diagram - An overarching Database diagram that depicts what each table/model is supposed to contain, and how it is related to another model.

Relation - Describes how a model is related to another such as one to one, one to many, or many to many.

rb - File extension denoting ruby code.

Forum - Implemented forum that allows users to communicate with one another

Map - Google maps API implemented to show users the location of relevant data.

ratyrate - Gem used for rating the locations.

Bug - General code error causing unwanted program behavior.

**9. Results:**

**9.1 Requirements, Analysis, Design:**

**9.1.1 Product Backlog:**

Note that the backlog is now empty as all tasks were completed in Sprint 3. Sprint 4 will consist of further testing and minor polish, thus a backlog is not present within JIRA.

During Sprint 4, we will finalize the look and layout of each page, we will also test all of the features of each page ensuring that they function as intended.

**9.1.2 Functional Requirements:**

**User Interface Functional Requirements:**

Navigation Bar – Located permanently at the top of the website with with links to the following pages: Home, Stores, Events, Parks, Restaurants, Vets, and Login/Register.

Home Page – Landing page for the website containing a welcome message and links to sign in

Stores Page – List of local businesses that have been approved by a site admin to be displayed publicly

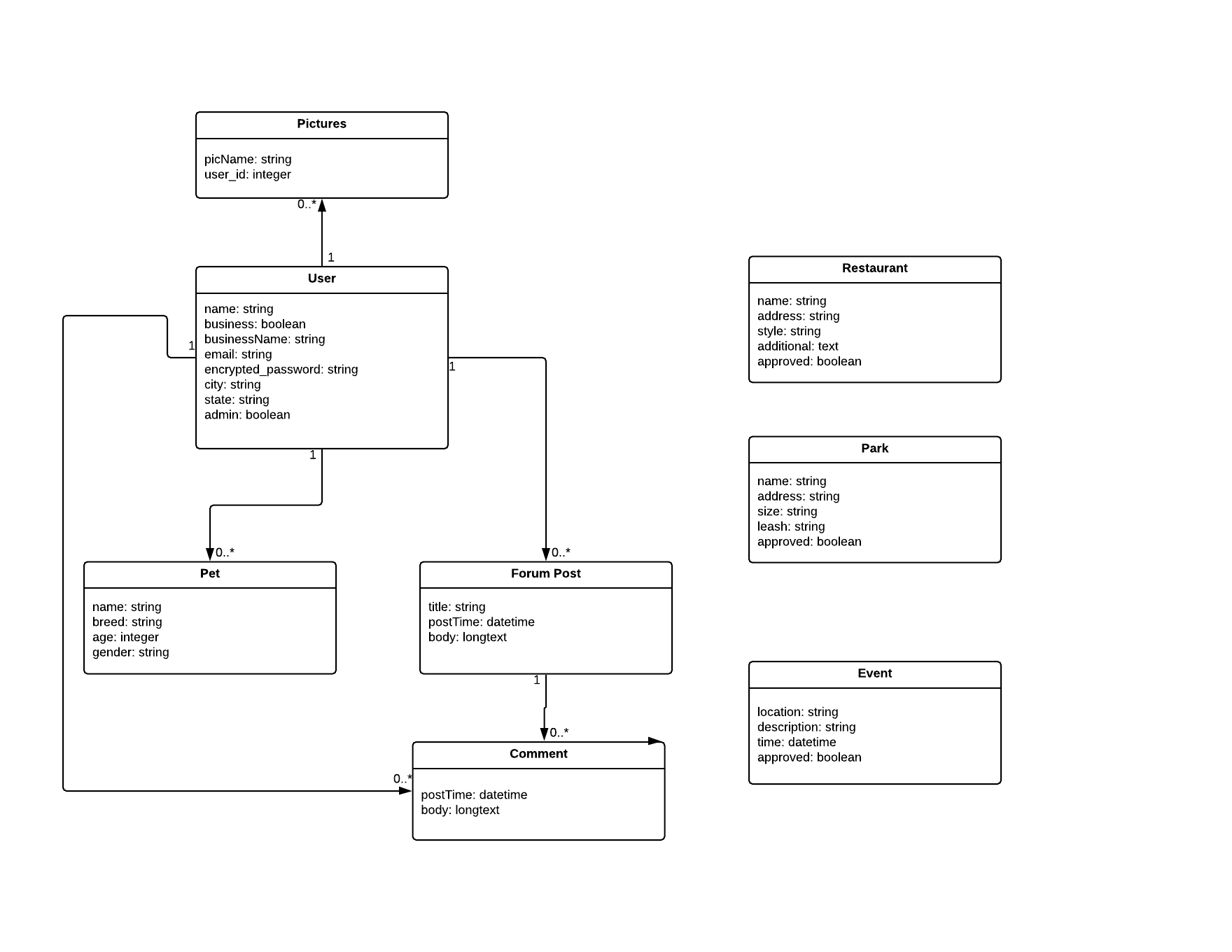
Events Page – List of events in a users area

Restaurants Page – List of approved restaurants in a users area

Vets Page – List of approved veterinarians in a users area

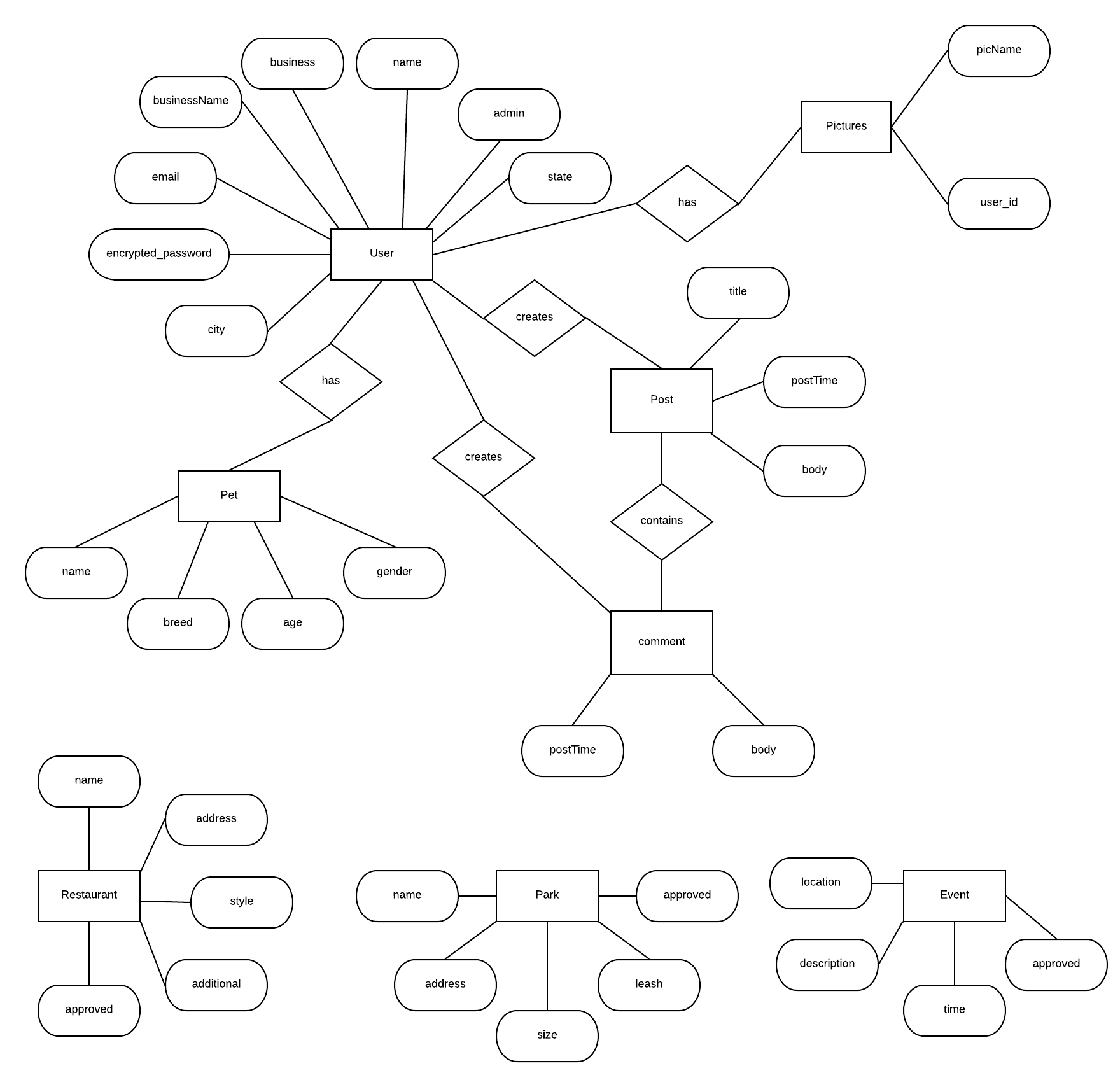
Forum - Forum implemented so users can talk, and Admins can moderate

Map - Pages have a map relevant to the displayed data



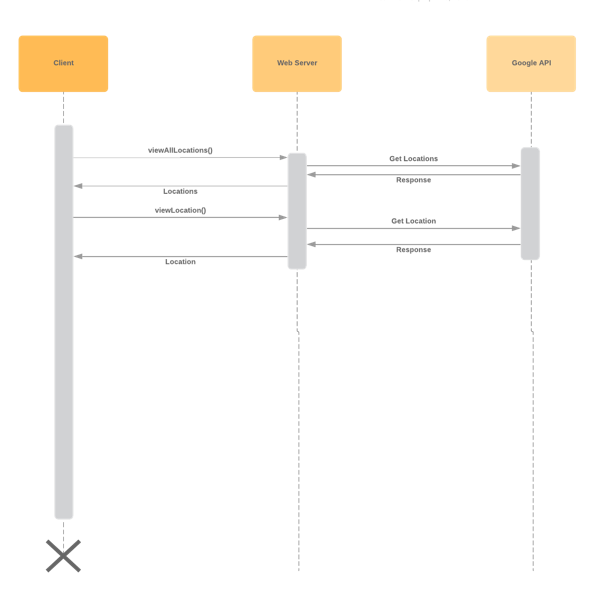
**Figure 2: Class Diagram**

This class diagram outlines the database models for the application as well as how information is stored inside of them.



**Figure 3: Database Schema**

The database currently consists of seven tables to meet the Sprint 1 goals: User, Pet, Post, Comment, Restaurant, Park, and Event. A user can add a pet profile to be displayed with their user profile. A user can also make a forum post or comment on another forum post. They can also view information about approved restaurants, parks, and events in their area.



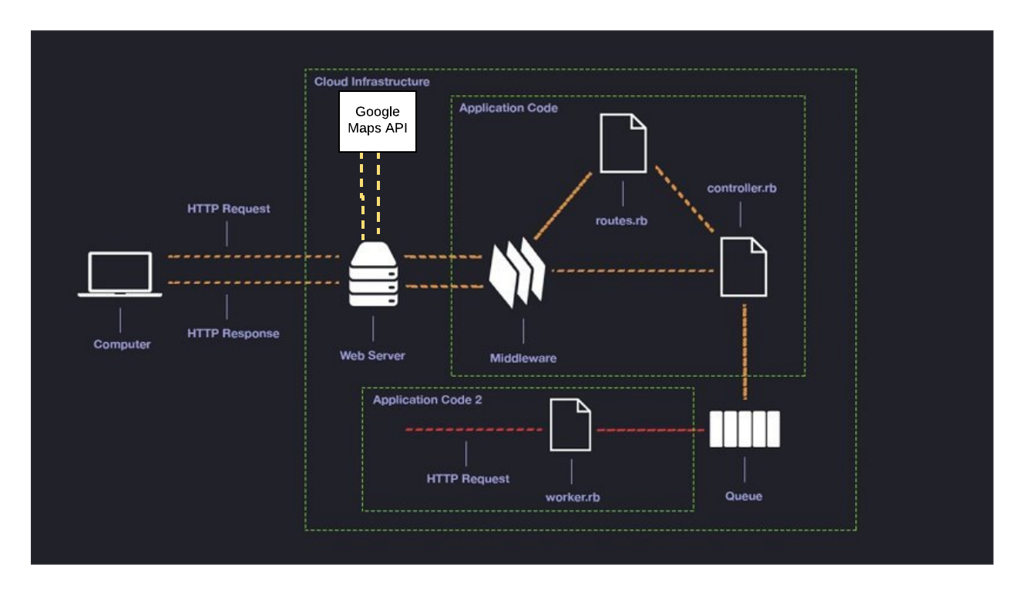
**Figure 4: Google Maps API Sequence Diagram**

This sequence diagram illustrates the interaction between the client, web server, and the Google API. The client can either request to view all pet friendly locations related to the part of the website they are viewing (stores, restaurants, vets, parks), or view one selected locations. In both of these instances, the web server will call the Google API and receive a response with location information to be displayed back to the user.

**9.1.3 Non Functional Requirement / Specification:**

**9.1.3.1 Base System:**

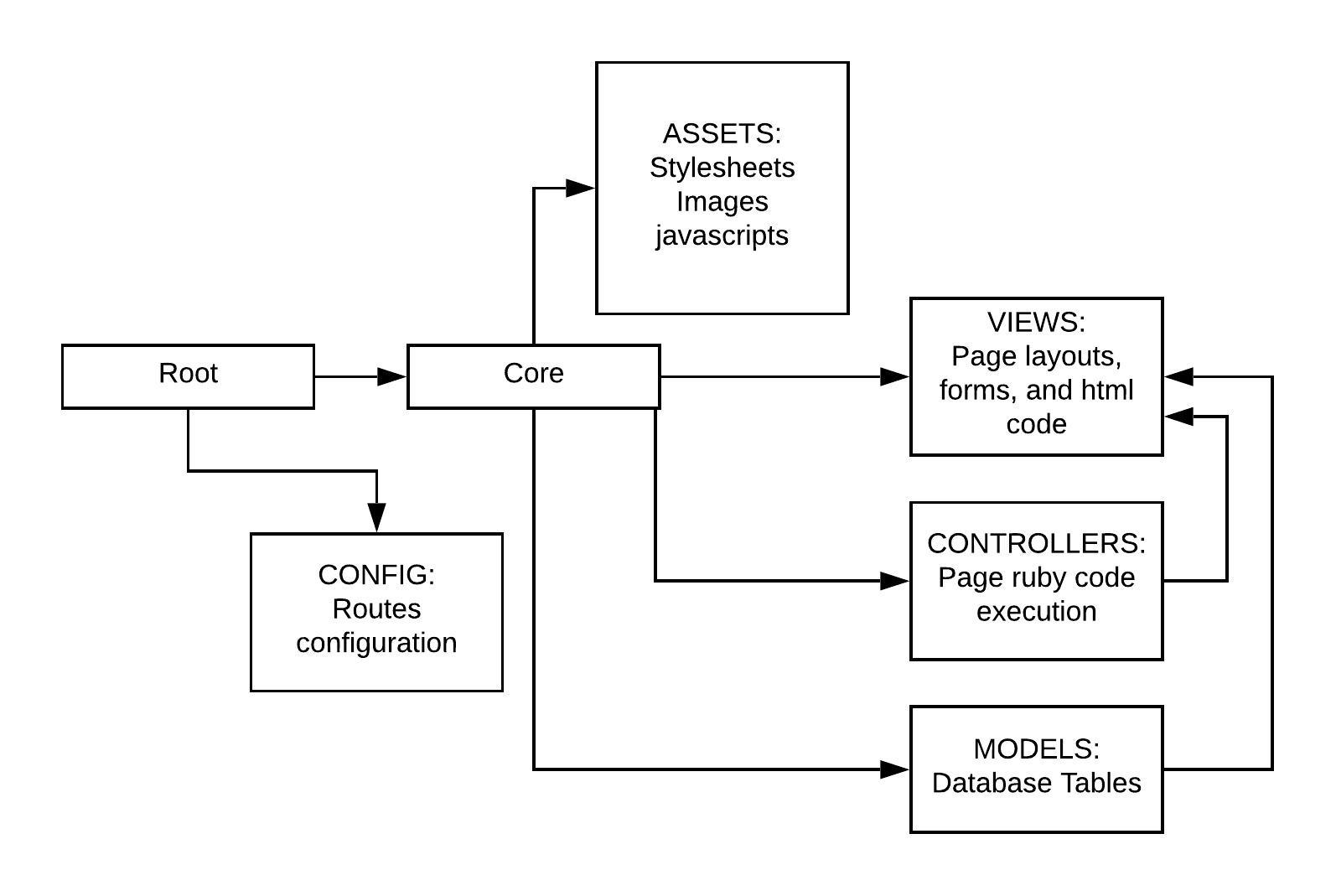
The project is based on a ruby on rails server. We are using Ruby 2.5.1 and Rails 5.2.2 for the most up to date features, with the exception of sqlite3 in that the latest 1.4.0 version is broken, therefore, 1.3.6 needed to be in use.



***Figure 5: Rails Architecture***

**9.1.3.2 Source Code Structure:**

Our source code file structure is as follows:



***Figure 6: Simplified Source Structure***

Within Root, we have our basic ruby installation, as well as included gem file. This is what allows the web server to run. Links on the site require the routes file contained in config to be set up such that they point to the correct destination within core/views. Ruby on Rails is pre-configured to know where to look for items making this setup easier. Within core, assets contains styling files and javascript actions. Views contains the primary items seen when navigating the page such as the home page, or forms that can be filled out. Controllers contains each corresponding views subfolder code execution. Again Rails knows where to look for matching names. Controllers allow for model tables to be called and displayed, or even new ones to be created and edited. Finally, models contains all of the database tables. An exact folder tree can be found in the appendix.

**9.1.3.3 Installation Procedure:**

For our system, there are two options for installation. The first option is to install a pre-configured virtual machine in which the user needs to simply clone the Github repository and navigate to the lunasList folder. From inside, the commands ‘bundle install’, ‘rake DB:migrate’, ‘rake DB:SETUP’, and ‘rails s -b 0.0.0.0’ can be run to have a local version running, reachable by typing ‘localhost:3000’ into a browser navigation bar.

The other option is to download a Debian based variant of Linux, our suggested version being Mint. The commands that follow may vary a little from distribution to distribution and are designed for use Linux Mint 19. After installation the following commands should be run in a terminal to ensure all necessary items needed are installed.

sudo apt-get update

sudo apt-get install ruby-full

sudo gem update

sudo gem update --system

sudo apt-get install ruby-dev zlib1g-dev liblzma-dev

sudo gem install rails

sudo apt-get install libsqlite3-dev

sudo apt-get install nodejs

Once these commands and or variations needed for the Linux installation have been run, the first installation variant instruction can be continued at cloning the Github repository. Video installation instructions are also available at:

Linux Setup - https://youtu.be/gkEwZndLZc0

Ruby on Rails Setup - https://youtu.be/Oh8GrPInDMQ

**9.1.4 Other specific requirements:**

In order to have this project work an active and stable internet connection is needed. For final deployment, a dedicated and port-forwarded server will be the ideal location to host the web app.

Administrators will be required to be put in place as data submitted by users must be reviewed and approved to ensure accuracy.

Ruby 2.5.1

Rails 5.2.3 - run “bundle update”

Any modern web browser can access the website.

The Linux system used for hosting the site will need to ensure g++ is installed in order to install the required gems for the forum.

The owner of the site will need to get a google maps api key and place it into the maps html code. This is needed because each call to the map costs a small fee. For our purposes, the free monthly limit provided by google was not exceeded, however the end customer will need to make note of this.

Many ruby gems require specific versioning. Reference the appendix for a copy of the gemfile.

**9.2 Development planning results:**

**9.2.1 Sprint planning summary:**

**Sprint 1:**

The project began with the team thinking on what core items a website using ruby on rails needed as well as what a user might like to see and what was necessary to be implemented via reading the customer’s initial problem statement, yielding our backlog. We then proceeded to select what user stories we felt were necessary to get the project on its feet. We followed this up with corresponding with the customer to ensure that our selection fell in line with what the customer wanted to see as well in each sprint and ensure proper prioritization. We are reliant on the method of contacting the customer to have final say on what goes in to each sprint.

Thus far, our method seems to be yielding positive results in that during the first sprint, all goals have been met to our satisfaction.

**Sprint 2:**

Sprint 2 had a relatively short planning process. Due to semester time constraints, and the lengthy learning process of sprint 1, we had to hit the ground running, with brainstorming and task value assignment being done quickly. This seems to have had a minor effect on how well stories were estimated, however, the theme of sprint 2 was not lost. Given a hiccup in customer communication, we proceeded with what we felt the overall vision of the project would require as well as suggestions from the sprint 1 demo. This entailed ensuring that all functions that worked with the database were implemented, as well as a map and forum implemented. Minor styling was added into the task as a precursor to sprint 3 such that we could develop a good understanding of how to evaluate the weight of future tasks as well.

Generally, we see the method used for sprint 2 planning as not ideal, but functional. It allows us to proceed with the project, using what we feel the customer was looking for from the beginning. We understand in the real world, this event is a possible occurrence and being able to overcome it is no small feat.

Regardless of hiccups and short time on hand for the planning process, the efforts have yielded results that we as a team agree are acceptable.

**Sprint 3:**

Sprint 3 was unique with its planning process in that initially, we thought we were going to be starting it the day that Sprint 2 ended, hence not much time to discuss it and what it was totally going to entail. We made a slight change however to give us the time needed for discussion and backlog assessment to be done. This did in turn present us with an awkward sprint cycle of less than two weeks. We also knew going in that we would have to cut our Sprint 3 short in order to match the deadline of the school semester. Having the extra days to plan did however prove to give us a more directed backlog and tasklist. Being that Sprint 3 is mostly cleanup and minor feature additions not requested by the customer, we felt that these quirks were not a major impact in the overall development.

Ultimately, our planning process for Sprint 3 is not what would be considered ideal, however, we did tailor it to suit our needs as well as the needs of the customer and the time constraints of the semester.

**Sprint 4:**

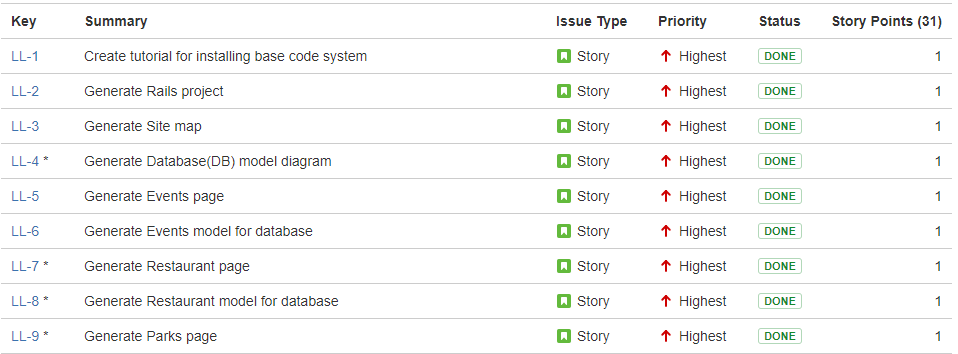
Being that Sprint 4 was the final sprint, we did not have to do much planning. everyone had the goal of testing the product and seeking out bugs, as well as finishing up minor details from the previous sprint. The idea behind this sprint was that as bugs were found, they were then reported within our slack channel to inform others of them. following, the member that found the bug would attempt a fix, and if they could not fix it, another member would then make an attempt.

Sprint 4 had a very different planning feel to it as opposed to the other sprints, however, this comes from the fact that all project tasks had been completed and we were solely working on finish and polish.

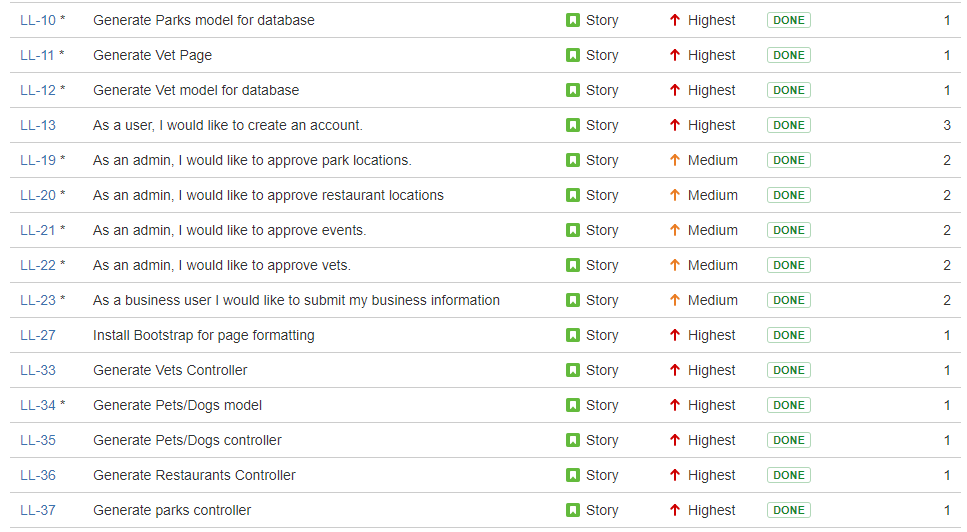
**9.2.2 Roadmap from Sprint 1 to Final Sprint:**

**Sprint 1:**

The focus of Sprint 1 was to set up the backbone of the project. The idea was to make sure that we had a working system that met the most important of the customer needs. We needed to have the databases for dog friendly locations and events in place, as this is the major theme of the site. We also needed for users to have the ability to add information as being that more places become open to the concept of allowing our canine friends to join, they may not immediately make that information available, thus community aid in gaining this information is needed. These items are also the core of what we needed to proceed further in the project. Below is the list of tasks implemented. We did complete all tasks for this sprint.



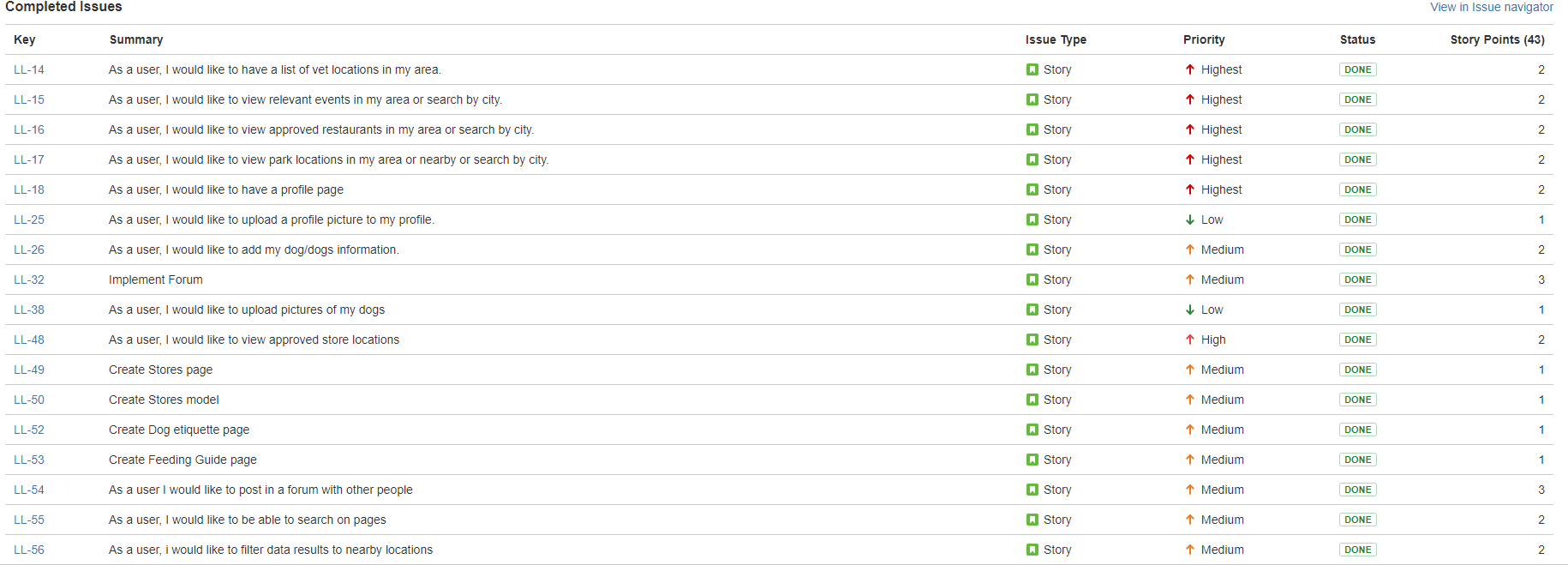
***Figure 7: Sprint 1 Issues***

******

***Figure 8: Sprint 1 Issues Continued***

**Sprint 2:**

The focus of sprint 2 was to implement what we called, “Secondary Features”, meaning they were not necessary to the core purpose of the website. They are items the customer would like to see implemented though. This primarily involved allowing for users to have a user profile page, on which they could also add their pets. Users were also given the ability to both upload a profile picture as well as photos of their pets. Another highly important item was adding in a map, so that locations added could be found easily without having to ever leave the page. This applied to general locations within the index pages, as well as a more precise map located on the locations main information page. We also aimed to add in the capability of filtering the index pages to allow users to search for locations within their city. Finally, a social networking aspect was added to the site in the form of a Forum, where topic boards can be set and users can post new topics, allowing for discussion within them. Below is a list of the major tasks, plus other minor additions. All tasks were completed on schedule.



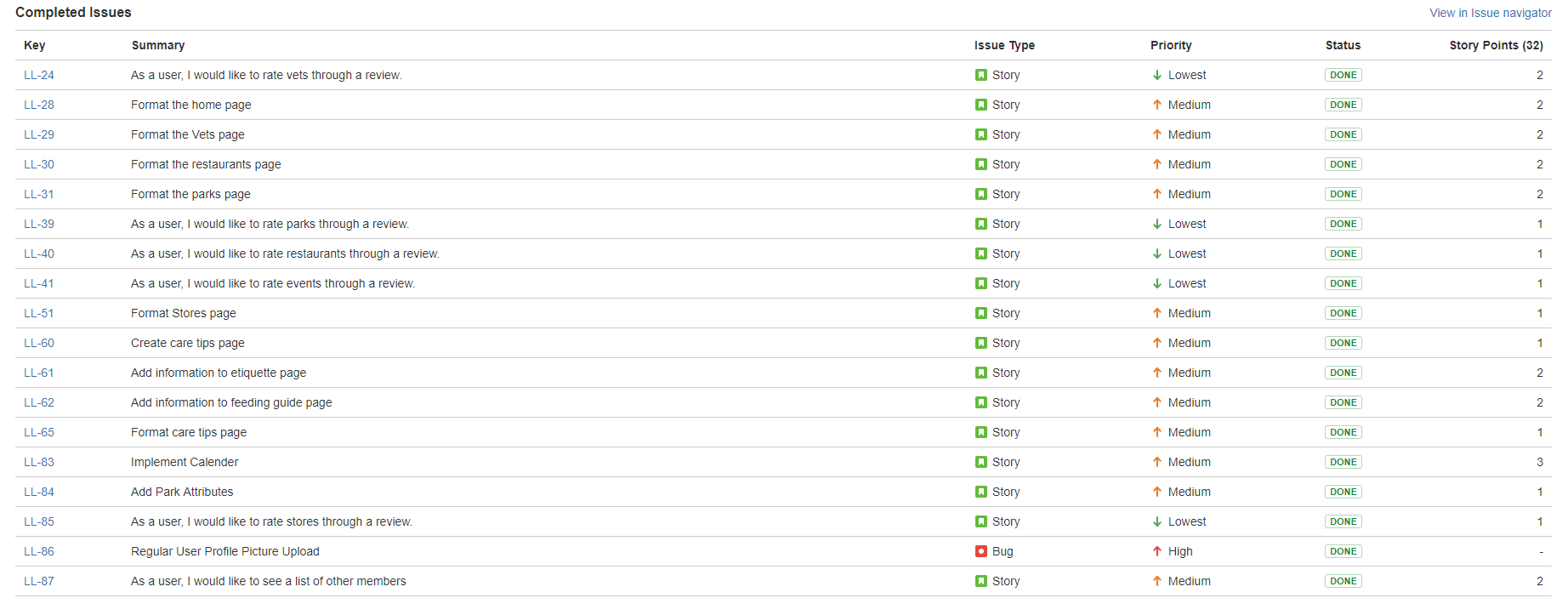
***Figure 9: Sprint 2 Issues***

******

***Figure 10: Sprint 2 Issues Continued***

**Sprint 3:**

The focus of sprint 3 was to polish the viewable pages and make them appealing to the eye, as well as to implement a function that we think would be neat, said function being able to rate locations. Finally everything was run through a final testing phase and any bug that presents itself will be corrected. Do note that Sprint 3 did not have the same workload as Sprint 1 and 2 because we did not encounter too many code issues and were able to wrap up most of what we thought to be relevant user stories with no stories from the previous sprints bleeding into Sprint 3. It is important to note that during the sprint, one task that was adding a calendar for events was removed. This was due to the fact that the calendar made the pages as a whole look less appealing and that the information was already available for the events, thus the removal decision. It was also decided that the user business end would instead be handled within the forum.



***Figure 11: Sprint 3 Issues***

******

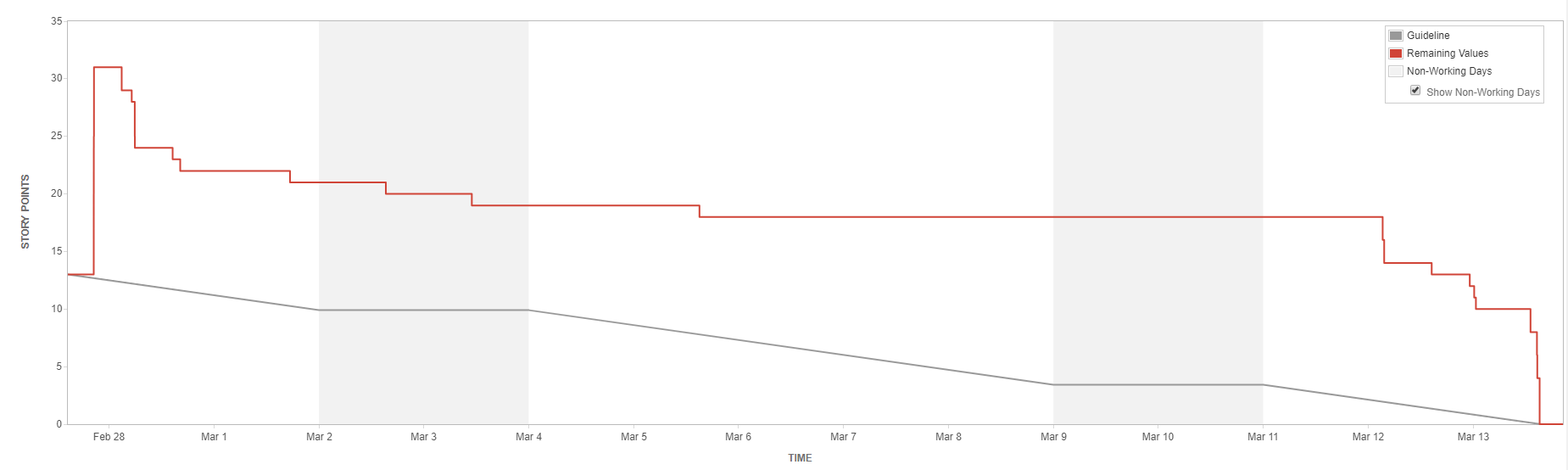
***Figure 12: Sprint 3 Issues Continued***

**Final Sprint - Sprint 4:**

The focus of Sprint 4 was to finish minor polish and bug fixing on the site. We had primarily completed the project at the end of sprint 3, with no more tasks remaining, thus, we were left with product testing. Every member ran the site and looked for bugs that were occurring. We ultimately only found one major bug on the events page. Said bug was causing the formatting of newly created events to become squished, while the events places within the seeds folder for database setup were displaying properly. Ultimately we were able to fix this bug. Other than this bug, we had known about a bug on the forum that requires a page refresh in order to get drop-down text-fields to show. Unfortunately, a fix was not able to be found for this issue, and is what potential future work on the site would be put towards. Another bug that we discovered and were unable to fix involved the rating system. We encountered an issue where when a user rates a location, the rating does not save. We know that we have our code and rails controllers set up as per the instructions for the rating gem, but were unable to solve the issue at this time. Thankfully, the bug does not correspond to a major feature of the site that was requested by the customer. Finally we ensured that the site was functioning as intended, minus the two bugs that we were unable to fix, for the demo. We also made a working video of the site for display at the demo. It should be noted that Sprint 4 did not make use of the JIRA system and as such, neither a burndown chart, nor a velocity chart were created. We also do not have a product backlog at this time as we were able to complete all of our assigned tasks.

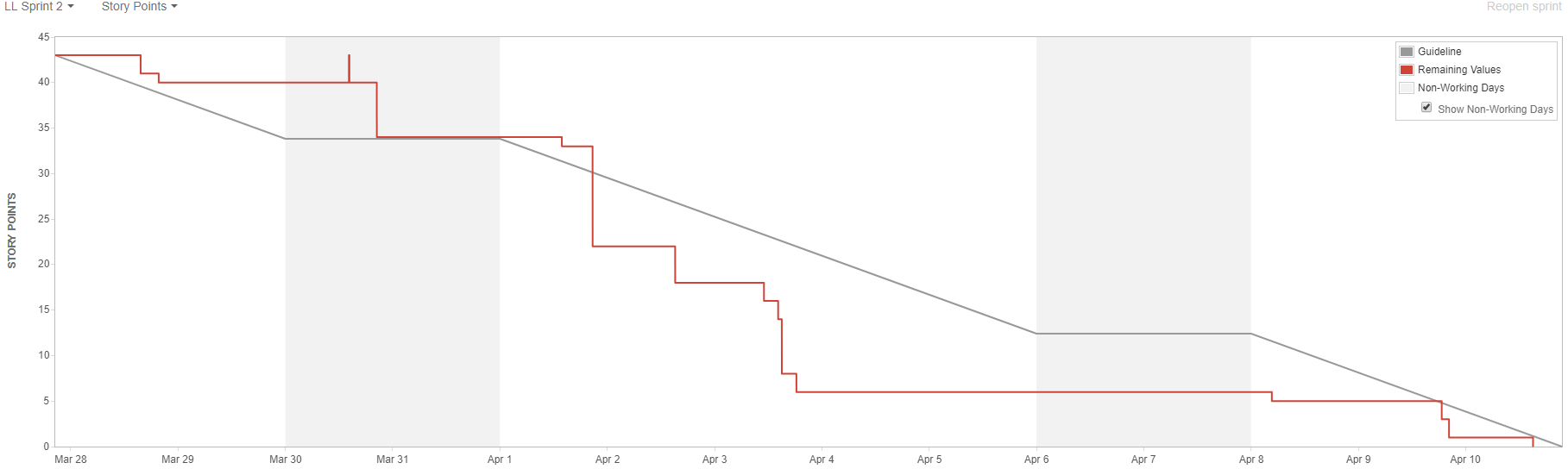
**9.2.3 Sprint Burn Down Chart History:**

**Sprint 1:**

****

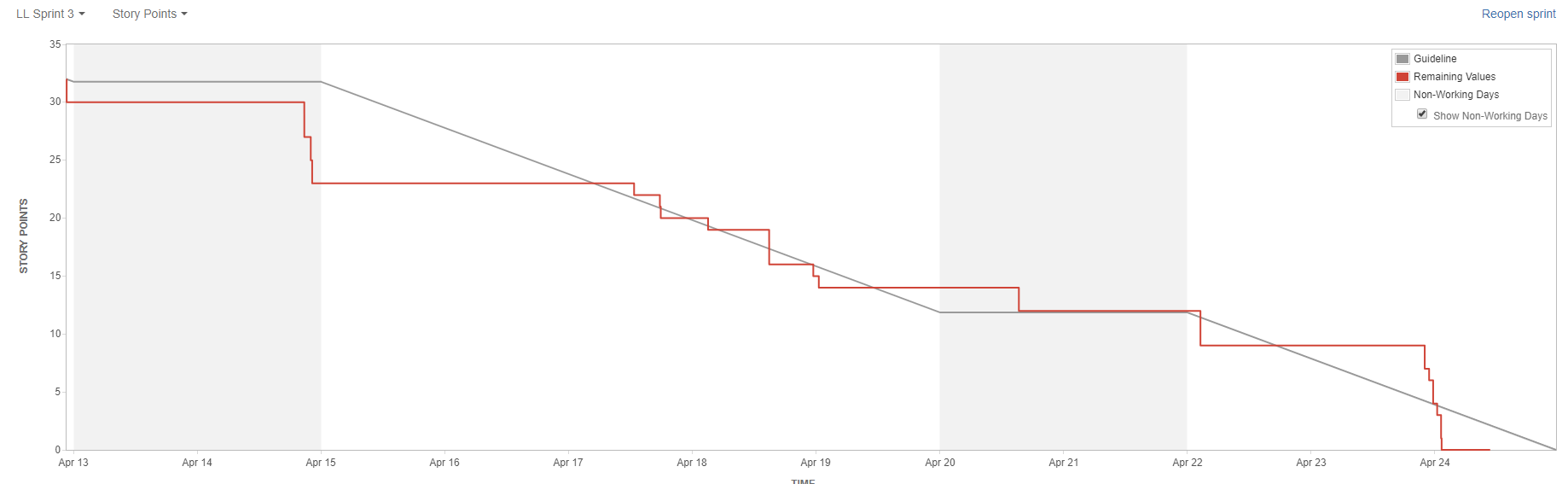
***Figure 13: Sprint 1 Burndown Chart***

**Sprint 2:**

****

***Figure 14: Sprint 2 Burndown Chart***

**Sprint 3:**

******

***Figure 15: Sprint 3 Burndown Chart***

**9.2.3.1 Sprint 1 Burndown Chart Description:**

As one can see from the chart, our sprint goals were met. It will be noticed that there is a long period of inactivity, however, that is due to a JIRA software outage during which issues could not be moved from in progress to complete. Near the end, it can also be seen that a fair amount of work crunching went in to play to complete the remaining tasks. It should also be noted that at the start of the sprint, several key stories were added in after the start of the sprint, hence the large rise of remaining values relative to the guideline.

**9.2.3.2 Sprint 2 Burndown Chart Description:**

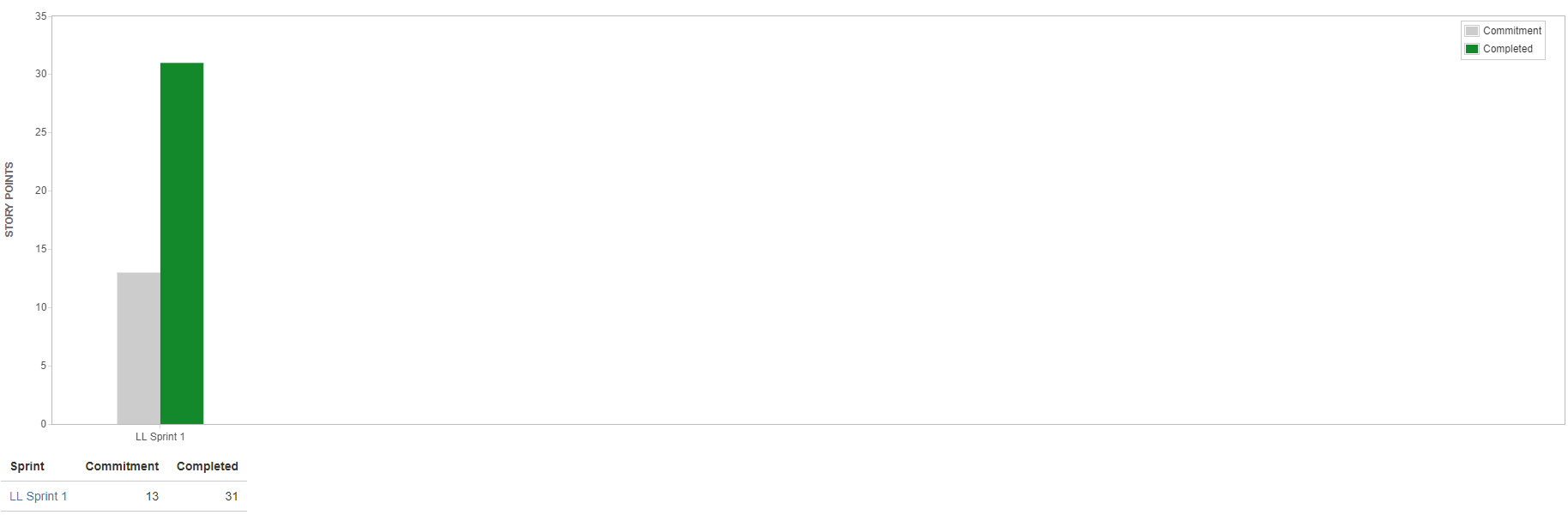
The sprint 2 burndown chart shows steady completion of tasks. We did still have a spike which comes from the addition of a task we did not realize that needed to be addressed. Also notice that a large flatline appears. This comes from a period of working on trying to implement a couple of the stories that were giving us some trouble and took a few more days than had initially been anticipated. The chart does show active work that is getting completed on time. This chart also shows that we have better figured out how to assign user stories to the sprint to better show total user story points and produce a graph closer to the JIRA system estimator.

**9.2.3.3 Sprint 3 Burndown Chart Description:**

The sprint 3 burndown chart shows steady completion of tasks again much like in sprint 2. We had no spikes indicating that we successfully had all needed tasks assigned in order to complete the sprint. Each long section of straight line was simply taking time to complete a task, or log into JIRA and update the tasklist. What we can take from the chart is that each member put forth time to their assigned tasks and completed each one in a timely manner, not encountering any issues that could not be solved. Again this chart shows continued efficiency in assigning user stories. Do note that at the end of the chart we see a final dip. This dip shows that we completed all tasks prior to the official sprint end.

**9.2.4 Sprint velocity and hours:**

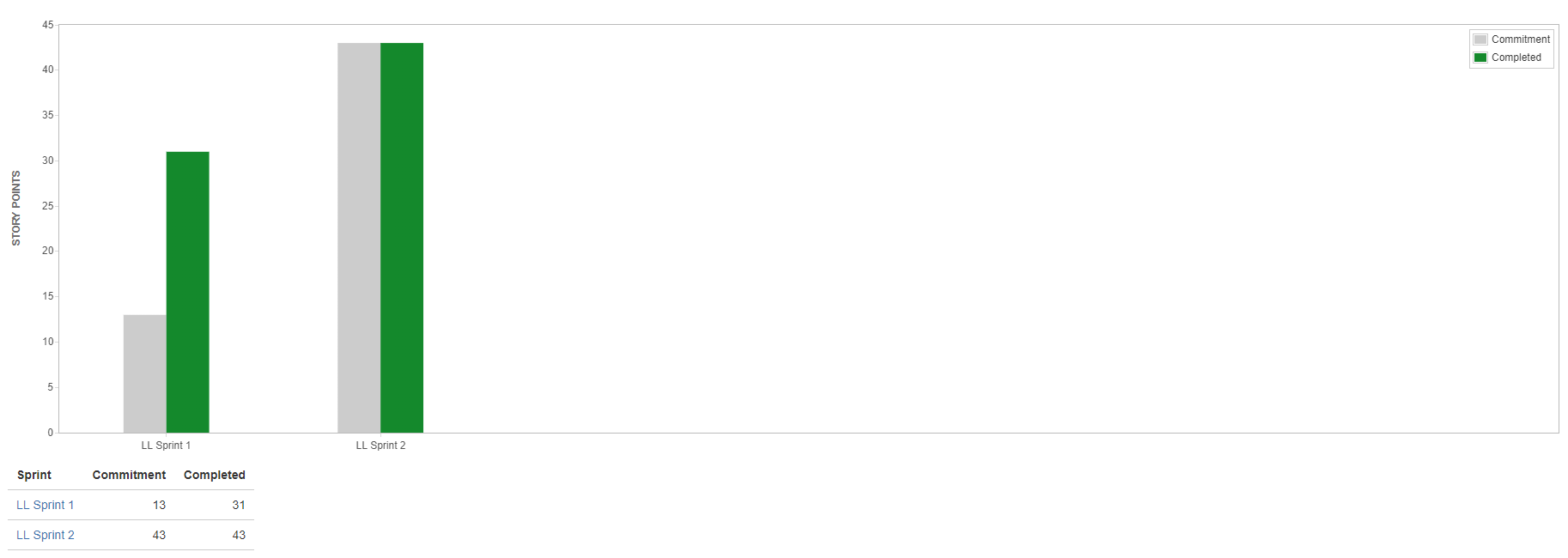
**9.2.4.1 Sprint 1 Velocity Chart:**

****

***Figure 16: Sprint 1 Velocity Chart***

We can see from the velocity chart that we well exceeded the Sprint 1 commitment. This does come from the fact that User Stories were added to the sprint in an ineffective manner as we further discussed the needs of the customer for sprint 1. The chart does otherwise show that we were well adapted and handled the necessary changes to ensure customer happiness and sprint completion.

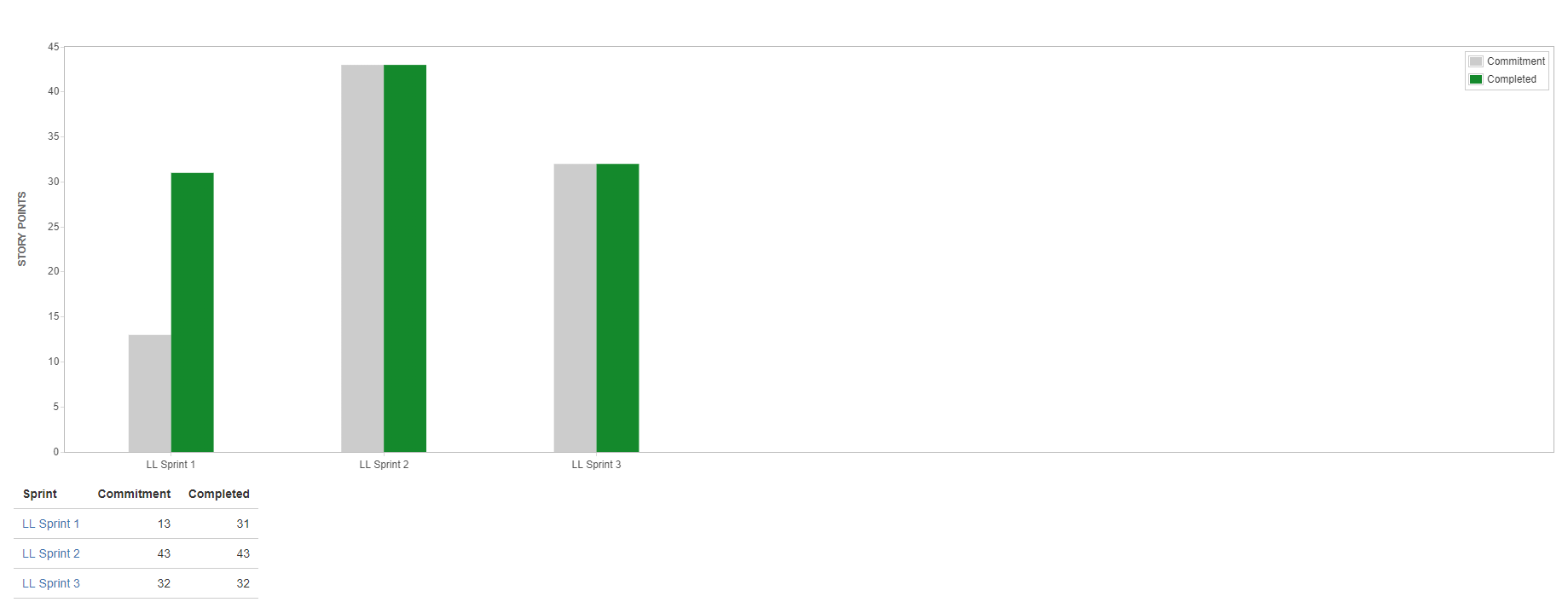
**9.2.4.2 Sprint 2 Velocity Chart:**

****

***Figure 17: Sprint 2 Velocity Chart***

We can see from the velocity chart that Sprint 2 had its commitment met. This falls in line with the fact that all team members pulled together and effectively completed their tasks on time. Do note that in comparing the velocity chart from sprint 2 to that of sprint 1, We can see that sprint 1 had an overestimation, whereas sprint 2 matched the expectation. This comes from the fact that users stories were added quickly as opposed to being added well into the sprint’s lifetime. Overall, the chart shows that we continue to handle each sprint well, and adapt to changes. Arguably sprint 2 could be said to be more effective in that the customer needs did not change. In our case though, this came from a lack of communication regarding user story verification.

**9.2.4.3 Sprint 3 Velocity Chart:**

****

***Figure 18: Sprint 3 Velocity Chart***

We can see from the Sprint 3 velocity chart that expectations were again met. We can again attribute this to the fact that each team member diligently completed their tasks and solved issues that came up with these tasks. We did have a lower total user story score to meet, however, this comes from the fact that sprint 3 entailed finalizing the project as a whole and did not have a large list of tasks. This also shows that our user story estimation was accurate for this sprint based on our feedback from our sprint 2 demo.

**9.3 Risk Tracking Table:**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Severity** | **Likelihood** | **Steps to Avoid** | **Outcome** |
| Installation Incompatibility across different systems | HIGH | HIGH | Create a single known working system, and create an iso of said system | PASS |
| Ruby Gem version issues | HIGH | MEDIUM | Test aspects that the gems control to ensure effectiveness | PASS |
| Users are unable to save added data | MEDIUM | LOW | Ensure ruby code is accurate and perform tests | PASS |
| Admins are unable to edit, approve, or delete additions | MEDIUM | LOW | Ensure ruby code is correct and run tests | PASS |
| Forum cannot be implemented | HIGH | MEDIUM | Admins are able to create forum sections, as well as delete sections. Users are able to create and post into threads. | PASS |
| Map cannot be implemented to show locations | HIGH | HIGH | Pages with maps are visited, and a map showing a marker, or markers is visible | PASS |
| Database search filtering inoperable | MEDIUM | LOW | Users are able to filter results by city on each data page | PASS |
| The pets table does not properly link ownership to users | HIGH | MEDIUM | pets have a user\_id identifier and can be reached from the correct user | PASS |
| Layout does not function | LOW | LOW | each page is visited and is looked at to show that is is displaying properly | PASS |

**9.4 Summary of Retrospectives:**

**9.4.1 Retrospectives:**

**Sprint 1:**

Overall Sprint 1 feels to have gone smoothly. We did encounter a major hiccup with the JIRA software, affecting the overall look of the Sprint Burndown Chart, but said issue was quickly resolved upon notification. We could handle to use more tasks as it seems the User Stories were completed rather quickly. We do need to ensure that we can fix some stories relying on others to be completed, however, in looking back at sprint 1 and its theming, it seems an unavoidable issue at this early state. We have not yet heard back from the customer regarding a demonstration and our planned setup for meetings. All team members seem to have been on track and understood well, what needed to be done. We did notice that we had to add User Stories after starting the sprint, something we need to fix.

**Sprint 2:**

Sprint 2 has both shown what we are doing well, what we have improved upon from the previous sprint, and new issues that have arisen, being detailed here. We were better able to divide tasks in a manner that did not require one task to be completed before the other. We did encounter one situation involving uploading images that did have another task rely on it, but the quantity of these situations was greatly decreased. Regarding assigned tasks, we were able to assign more tasks and better fill our sprint time with productivity. We have found a good balance of available time and sprint task quantity. We do still need to work on customer communication as well as having all tasks pre-assigned before beginning the sprint. We have better learned that a customer is not always going to be available, however, obtaining several forms of customer contact is a better way to ensure that effort to contact a customer has been made. Customer communication has taught us that we will not always have a set in stone sprint goal and that we must work to provide what we feel to be in the customer’s interest, knowing full well that it may have to be scrapped down the road. Communication as a team continues to be a strong point especially in scrum meetings with being able to determine problems and solve them. Finally a new issue we have encountered and unfortunately appears that it will be a continued issue into the final sprint, is not having the time to really discuss the start of the sprint. Constraints of the school semester have made it such that we have to start the next sprint the day sprint 2 ends which is not ideal. What this has taught us is that initial project sprint roadmapping needs to be better implemented such that we have a general idea of what each sprint will fully entail. With this in mind, we can better populate the product backlog to meet each sprints’ needs, and have a far less chance of having to add tasks once a sprint has started. Again, better logging of scrum meetings needs to be worked on. A final lesson learned is that we should expect potential bugs as we near the end of a sprint. It is impossible to think that this does not ever occur and we will have to learn how to make proper adjustments.

**Sprint 3:**

Sprint 3 continued to show us that as a team, we have upheld team communication as well as task completion time. We continue to discuss problems as they come up and work together to find solutions as needed. All major changes are properly discussed and detailed between team members. Task designation and assignment has seen a major improvement with each member having a suitable number of tasks and not relying on another team member’s tasks to be completed. Even though tasks are completed on time, we could still improve on starting tasks more quickly, and moving to a different task if we are having trouble with a particular one. Unfortunately during this sprint, we did not have the opportunity to put this into practice as we did not encounter any issues that prevented any one member form completing a task. We did were unable to improve upon sprint planning and start times which again falls into the same situation with Sprint 2 and needing to meet semester time constraints. We have however learned from this that it is ok to end a sprint early. We will see this issue occur again with Sprint 4, though we have adapted to this issue.

We have improved upon communication with the customer on our end, however, it should be noted that we still received few responses. This in turn led us to still further learn that sometimes a team must proceed strictly with the first available project information, and be prepared to make changes as necessary once communication is re-established. Thankfully, once reaching the end of a project and working strictly on polish, this does not pose as large of a challenge if some changes were to be requested.

A big lesson learned is that sometimes a desired feature is unable to necessarily be implemented without changing the core of the system. in our case, in order to implement the rating system, we had to upgrade our version of rails to the newest version. While this did not pose any problems to our project, it is important to note that an upgrade similar to what we did could in theory break an entire project that relies on specific versioning. That being said, we have learned to better understand version requirements for tools used in building a project.

Another important lesson learned is that sometimes a task that has been planned during the sprint planning process does not always end up turning out to be ideal and is in turn dropped. In our case, we had originally intended to add a calendar to the system, however, due to layouts of the web pages and font sizes, the calendar played only to deter being able to see the important information of the page. We instead chose to pursue displaying the information available in a prettier manner.

Finally, another lesson learned is that as a project comes to a close, the quantity of tasks remaining tend to become sparse. because of this, we need to find ways to apply ourselves during the down-time whether it be through rigorous system testing, or assisting others with their tasks.

**Sprint 4:**

Sprint 4 showed us once again that as a team, we are able to quickly communicate between one another and solve issues that present themselves during a sprint. We were also able to coordinate a loose task system during Sprint 4, said task system being that as bugs were found, a member would then have an added task of attempting to solve the bug. This is a unique way of adding and addressing tasks, but we feel that the way to solve this in the future is to implement more complete bug testing in each sprint as we move along to find bugs faster and add fixing said bugs to the tasklist sooner.

A major lesson learned is that we have to accept the fact that sometimes bugs cannot be solved within a sprint completion time and will either have to be included within the release for further on launch work, or pushed into another sprint as a priority task. This of course happens in the real world and seeing it first hand gives that experience to understand how to handle the scenario.

Given Sprint 4’s short period of one week, there was not enough time to run into many other issues or situations where we learn, but we feel that it has taught us the process of finalizing and pushing a product. This includes extensive testing, ensuring that the system runs across all systems after code has been merged, and making sure that the product is to customer expectation for release.

Finally, regarding communication with the customer, we still encountered problems in trying to contact the customer regarding the sprint. This in turn led us to work off of what we learned from our previous sprint demos, and will set us up to further learn how to handle a scenario where a customer may not approve of the final product. In our case though, given that this is our final sprint and the end of the semester, we do not foresee this being an issue.

**9.4.2 Table of Retrospectives:**

|  |  |  |  |
| --- | --- | --- | --- |
| Sprint # | What did we do well and should carry on doing? | What did we do that we should be avoiding in future? | What were the surprises (unusual items) during the sprint? |
| Sprint 0 | We should carry on with picking the relevant topics to be spoken about during each meeting. | Staying on track during discussion and not diverging should be practiced given a couple of times that it has happened, however, even in this, the talking points of the meeting were still met, thus care needs to be taken. | No real surprises as of yet. |
| Sprint 1 | Discussions of where we are on the project continue to be effective as well as relevant issues being brought up and solved in a timely and effective manner. | Have all user stories ready. Sufficient customer contact. Handle more issues. There did exist task dependency. | User stories added after sprint start. Unexpected JIRA outage. Customer responsiveness. |
| Sprint 2 | Project issue discussion continues to be effective. Task designation to avoid other task completion reliance has been improved.  Quantity of tasks assigned has been improved. | Customer Communication.  Time allocation between sprints.  Scrum meeting logging. | The time we will have between sprint 2 and sprint 3 has thrown us a unique curveball. |
| Sprint 3 | Issue discussion remains a strong. Task designation has seen improvement. | Little customer Communication.  Little time allocation between sprints.  Late task start time. | Needing to upgrade to a newer version of rails was unexpected. |

**9.5 SCRUM Meeting Notes Sample:**

**3/13/2019**

This is a direct pull from our chat client.

WJones [11:39 AM]

I am going to make a video for the demo of the site being navigated and primary sprint 2 features being detailed. Will likely do this tomorrow evening. Did the user images ever get sorted out or is it still causing trouble? lastly, I do have a watermark image made, and am going to try to get it on the site today for sure. Regarding Sprint 2 Team Report, My goal is to begin making necessary edits to that as well tonight.

My other question is where do we stand on the feeding and care pages

Vinh Tran [12:34 PM]

still sifting through some information....as for actually creating the pages....

Thi Phan [1:03 PM]

I am sick. Will not be in class today. Let me know if you need me to do something. I have problem with routes & pets controller. I will try to fix them before Wednesdays.

WJones [1:49 PM]

just a quick preview of the watermark

https://gyazo.com/46e7e73987f4649f45f7928335f93674

Gyazo

Vinh Tran [1:50 PM]

nice

WJones [1:57 PM]

In any case, I figure we try to have everything finished off tomorrow. going ahead and working on sections in the sprint 2 team report

**10. Conclusions:**

**Sprint 1:**

In conclusion, we feel that Sprint 1 went well to what was expected. We obviously have more learning to do and trying other options to solve what did not work well, otherwise we feel that customer expectations have been met as well as exceeding our own.

The theming behind Sprint 1 being environment setup and database configuration, our goal was to set up the base environment, ensure it was stable, and get the core website running. We needed to learn what issues may come from different systems, as well as ensure all basic code problems could be handled.

Thus far, no major requirement changes have been made. Our only change came in to play with desired user story implementation in that we had to go ahead and partially set up the user model in order to ensure that other functions were working.

The lessons learned from Sprint 1 have been simple but highly important. We obviously need to ensure that we are taking on enough work to not be sitting around and that we need to be sure to have all User Stories ready to go before starting the sprint. We also need to ensure that we maintain effective contact with the customer with ample time ahead of sprint close such that we are unsure as to if we will be able to meet for a demonstration in a timely manner. We have learned that our team does work well together and that we should keep up with our effective level of inter-communication. Our meeting scrum meeting agendas have been on point to us completing this sprint as well.

**Sprint 2:**

The Conclusions from Sprint 2 are that we have successfully completed the goal set by our theming and tasks. We did encounter hiccups with bugs during the process, however, these were sorted out and fixed. We both implemented lessons learned from Sprint 1 while also learning new issues such as allotting ample time for sprint preparation.

More in depth on the tasks, our theme was implementing features not necessary to the core of the website, but still requested by the customer, as well as tidying up functions relevant to the core of the website. Of utmost importance were adding in the forum, maps, and ability to upload images. Being able to add profile information as well as pets was also added in, which we felt would be necessary to complete our vision of Sprint 2 in addition to that of the customer’s vision. Albeit, we did not get a firm view on what the customer wanted in totality for Sprint 2.

Again, we have had no major requirement changes, however, we are more prone to this occurrence based on the fact that we did not have a clear cut goal set by the customer for this sprint. Regardless, we find that it will add to our learning experience to have either performed what the customer will find either at or above satisfaction, or to have to make adjustments. We have had some user story deviation in having to adjust how a task was handled. Particularly in implementing the forum. One method did not work, so another had to be used which in turn cause two user stories to be completed by a singular action. While not a major issue at the time, it could theoretically cause a user story to be scrapped in the future should it happen again.

Finally in regards to lessons learned, We do need to take note of bugs coming up that we have little time to fix, or potentially have to push back to another sprint. Ideally the solution to this is to have work completed on the task sooner, but also to not let other tasks fall behind due to another task’s issue. We also need to continue improving customer contact such that we are left to our own machinations in developing the project. This can lead to having not met customer expectations and having to make major overhauls, costing production time.

**Sprint 3:**

The conclusions from Sprint 3 are again that we have successfully met the goals set by the thememing of the sprint and tasks. We did not encounter bugs that were tough to correct. We did however have to make a decision regarding removing a task as it proved to have a negative effect on the visual layout on the web app. Lessons from previous sprints continued to be employed as well as new lessons learned.

Moving in to the theming and tasks, Sprint 3 was primarily focused on the layout of the webapp and implementing minor additions that we felt would be useful to the site. Tasks falling under the layout of the site were approached in similar manners of making the page look appealing i.e. information placed in logical locations. An example being on the profile page. The users information and photo are placed at the top alongside one another, with their pets below. Layout comprised the major portion of the sprint. Another minor addition was the ability of an admin to view and edit a user, and even setting them as admin, while not allowing basic users to affect others. Admins received slightly different views on each page allowing for greater control of the content such as making a user an admin, or approving and deleting information. Another feature added was the ability to rate locations. All location data has the ability to be rated using a star system. Finally, we were original going to include a calendar for the events page, however, it proved to damage the styling of the page in a manner that we felt it necessary to drop the task. We instead chose to format the page such that event data could be easily seen, accomplishing the same basic effect of having a calendar. This in tandem with the upcoming events section on the home page will act as a suitable replacement for the calendar. Finally, bugs in the search and image upload were successfully corrected.

Regarding the requirements, we again had no changes. We do not foresee any changes regardless of the communication challenge, given that we have striven to follow what we were given during the Sprint 2 demo. We did again have to find an alternative implementation method regarding the calendar, detailed in the above paragraph, however it still meets the requirements.

Regarding lessons learned, we do need to continue working on sprint planning and allowing for time to be allotted to allow for greater user story preparation and planning. We have learned that systems require upgrades from initially implemented versions in order to allow for sub-technologies to be implemented. While this did not affect us negatively in Sprint 3, care must be taken to fully understand version requirements in future tasks or projects. Customer communication will always be an area for improvement and each iteration continues to teach us how to approach this. We also learned that tasks can have negative effects and need to be changed or dropped as is the case with our task of implementing a calendar. In these scenarios we feel it necessary to try and find suitable alternatives and modifying a task as opposed to completely dropping the task or idea.

**Sprint 4:**

The conclusions from Sprint 4 are that we once again work well as a team with respect to communication and understanding how to approach problems as they present themselves. Ultimately this led to us having what we feel to be a successful final product that matches the customer expectations. We did encounter two last minute bugs that we were unfortunately unable to solve, and one bug that we were able to fix. These two encountered bugs that we could not solve at this time are not detrimental to the function of the project, with one bug, the forum drop-down text field, having a workaround.

With theming and tasks, Sprint 4 was focused on finalizing the project, meaning we worked on the last bit of the layout and fixing bugs that we encountered. We ended up not having to make any major changes to the layout of the site as we had completed these tasks within Sprint 3. With that in mind, we focused on testing the site extensively, searching for bugs. We did encounter three bugs. One of the bugs was related to the layout on the events page involving newly added items being squished down, however, we were able to solve this bug. Within testing, two other bugs found involved ratings for events not saving, and the forum not always showing a drop-down text field. We were unable to fix these issues. The forum drop-down text field issue is able to be worked around with a page refresh. These actions completed the theming of Sprint 4. Lastly, a representation video was created for use at the large-scale demo.

Regarding lessons learned, we obviously still have to work on customer communication. This is a never ending topic in the real world. There will always be room to improve and learn as no one customer will ever be the same.

Second, we learned that it is unreasonable to expect that a project will complete taskless or bugless. In our case we did finish all user stories relevant to the project, however we did encounter some bugs, even some that could not be solved.

**11. Sources:**

**Thredded:** [**https://thredded.org/**](https://thredded.org/)

**Bootstrap:** [**https://github.com/twbs/bootstrap-rubygem**](https://github.com/twbs/bootstrap-rubygem)

**Google Maps:** [**https://developers.google.com/maps/documentation/**](https://developers.google.com/maps/documentation/)

**Ruby on Rails:** [**https://rubyonrails.org/**](https://rubyonrails.org/)

**Geocoder:** [**https://github.com/alexreisner/geocoder**](https://github.com/alexreisner/geocoder)

**12. Appendix:**

**12.1 SCRUM meeting:**

**1:**

Ruby or Javascript potential. We will use something everyone is comfortable with.

We should focus on the process and always note what works versus what didn’t. Obviously project completion with a working product is important, but the process is just as important to learn.

Ensure everyone understands the uploaded documents for reports.

Decide on the team name to be unique.

**2:**

Look at the uploaded projects so that we can begin determining what everyone wants to do. Be sure to pick a primary and backup projects that are liked.

Begin learning JIRA and play with the sandbox. Be sure to sign in to JIRA and comment on the checklist for participation. Go through the JIRA tutorial.

Consider what each role is going to require.

**3:**

Make note that Sprint 0 will be coming to an end soon, so be sure to have the JIRA tutorial completed, and play around in the sandbox.

Establish who is responsible for what parts.

Decide on a project by the end of the meeting.

**4:**

**3/11/2019**

This scrum meeting consisted mostly of code bug smashing

Set up remote computer such that users experiencing issues have a known working system to finalize tasks from.

Begin getting ready to merge code, and have a coding freeze at the end of Wednesday.

**5:**

**3/13/2019**

Finalize sections of code and work out any major bugs involved in adding or creating table entries.

Perform testing to ensure that both user and admin can see the correct pages and items relevant to their security status.

Merge pull requests from github and fix any merge issues.

re-perform testing upon completion of merges.

Reach out to the customer and ensure that the desired meeting time can be met.

**6:**

No problems with tasks thus far

**7:**

Issue with code in profile pictures discussed and looked at

continue trying to fix, otherwise other tasks continue to be completed

**8:**

Begin prepping for demo, ensure everyone understands their sections

Regarding pictures issue, try new model to hold images.

**9:**

**3/13/2019**

This is a direct pull from our chat client.

WJones [11:39 AM]

I am going to make a video for the demo of the site being navigated and primary sprint 2 features being detailed. Will likely do this tomorrow evening. Did the user images ever get sorted out or is it still causing trouble? lastly, I do have a watermark image made, and am going to try to get it on the site today for sure. Regarding Sprint 2 Team Report, My goal is to begin making necessary edits to that as well tonight.

My other question is where do we stand on the feeding and care pages

Vinh Tran [12:34 PM]

still sifting through some information....as for actually creating the pages....

Thi Phan [1:03 PM]

I am sick. Will not be in class today. Let me know if you need me to do something. I have problem with routes & pets controller. I will try to fix them before Wednesdays.

WJones [1:49 PM]

just a quick preview of the watermark

https://gyazo.com/46e7e73987f4649f45f7928335f93674

Gyazo

Vinh Tran [1:50 PM]

nice

WJones [1:57 PM]

In any case, I figure we try to have everything finished off tomorrow. going ahead and working on sections in the sprint 2 team report

**10:**(This was the same for each meeting during Sprint 3)

Continue completing user tasks. If anyone has issues, bring them up in the slack channel.

**12.2 Customer Client Discussions:**

**Email 1:**

Mr. Rivera,

We will be the team working on your project "Luna's List". We have a few questions we would like to ask regarding the project.

* What is your preferred way of communicating (email, phone, skype, etc.)?
* Is the primary vision for the project mobile or web based? (As a group we have very limited mobile development experience so web-based would be preferred)
* Do you have any specific vision for the UI design of the site or will that be left to us to decide?
* Do you have a full problem statement already created or will that need to be generated

Thanks in advance!

**Email 2:**

Noah,

Thanks for reaching out. I am excited about working with you and the team!

My preferred way of communication is all 3.  First email for anything quick. Then skype in order to communicate effectively. Finally for anything urgent phone would be best.

We could absolutely do web based and move to mobile after the MVP.

I will leave it to the group on UI vision.  Some elements that are important will be ease of use and interesting as well as organized and visually appealing.

The full problem statement is here:

Dogs have been interwoven into our lives at a scale never seen before. They have moved outside of the perimeter of our homes and yards and into our roads, stores and entertainment areas.

More and more questions have flooded businesses, parks, and buildings on when, where and how can I bring my dog?

Lunas List will help answer these questions and more by providing a platform that provides communities with Dog friendly spaces.  It will also bring Dog Owners together in different ways.

Noah can I ask for a quick bio or resume of the group who is working on this project. And let me know if you have any questions. Also, is there milestone dates that are already established?

My number is 281.513.1354.

Thank you in advance,

Victor M. Rivera

**Email 3:**

Mr. Rivera,

Thanks for your response. As requested, here is a quick bio of the team members:

Thi Phan: Python, Java, Swift, Ruby on Rails

Wesley Jones: Python, Java, Ruby on Rails

Larz Leonard: Java, Python, Ruby on Rails

Vinh Tran: Java, C/C++, SQL

Noah Hanks: Python, C++, SQL, Ruby on Rails

Since most of us have experience with Ruby on Rails, it would probably be the best platform for us to base the project on. In regards to your question about milestones, the dates are not established yet but there will be a total of 4 sprints. If you have any more questions please feel free to ask.

Noah Hanks

**Email 4:**

Yes I can

On Feb 12, 2019, at 9:56 AM, Noah Wilson Hanks (nhanks) <nhanks@memphis.edu> wrote:

Mr. Rivera,

We would like to set up a skype meeting with you to discuss the project further. Will you be available to meet tomorrow (Feb 12) at 4:00 PM?

Noah Hanks

**Email 5:**

I can.

What will the agenda be?

Víctor

On Feb 18, 2019, at 4:11 PM, Noah Wilson Hanks (nhanks) <nhanks@memphis.edu> wrote:

Mr. Rivera,

I just wanted to confirm that you are still able to meet with us again on Wednesday (Feb 20) at 4:00PM.

Noah Hanks

**Email 6:**

Noah,

I will be en rite to the airport. Please call me on my cell phone 281.513.1354.

Victor

On Feb 18, 2019, at 4:21 PM, Noah Wilson Hanks (nhanks) <nhanks@memphis.edu> wrote:

The agenda for the meeting will be to discuss the current user stories that we already have for sprint one and to ensure that we are on the same page as far as priorities go.

**Email 7:**

Yes let’s reschedule for Friday. Send over what you have and I can also review them in the plane.

Thank you,

Victor

On Feb 20, 2019, at 4:58 PM, Noah Wilson Hanks (nhanks) <nhanks@memphis.edu> wrote:

Mr. Rivera,

It would be best if you were able to view the list of user stories as we discuss them. Would it be possible to reschedule the meeting for Friday (Feb 22) at 2:00PM?

Noah Hanks

**Email 8:**

Noah and Team,

I apologize for the confusion on my end. I thought it was at 4 pm.

We can do email. Or if you want to call me my number is 281.513.1354.

Victor

**Email 9:**

Mr. Rivera,

I never heard back with you on skype so I wanted to confirm over email. Will you be available to meet with us to discuss the results of sprint 1 this Monday (Mar 18) at 4:00PM?

Noah Hanks

**Email 10:**

Mr. Rivera,

In class today, we found out that Dr. Yu would like to be present for the project demonstration. Would it be possible to reschedule for another time this week when he is available? We would like to tentatively move the date to Wednesday at 4PM, dependent on Dr. Yu's availability. I apologize for any inconvenience this may cause.

Additionally, we will be starting sprint 2 soon. If there is anything specific you would like to see for sprint 2 we can discuss that after the project demonstration as well.

Noah Hanks

**Email 11:**

Hi Noah,

I can do early on Wednesday, bit in the afternoon I will be on a flight until 5:30 pm.

Are there other times?

Victor

**Email 12:**

After speaking with Dr. Yu, it seems like the next available time will be next Monday (Mar 25) at 4PM. Would this work for you?

Noah

**Email 13:**

Mr. Rivera,

Our project demo with Dr. Yu is set for Monday (Mar 25) at 4PM. I wanted to check and see if you would be available during this time. If not, we can schedule another time to demo sprint 1 with you separately.

Additionally, if you would like access to the github for the project, I can get an invite sent out to you. I just need your github username.

Noah

**Email 14:**

Mr. Rivera,

Attached is our current project backlog. The bottom section was added very recently and hasn't had descriptions written yet but this will be revised likely in the next few days. Please feel free to take a look and add any comments you think would be helpful. Additionally, if there are any user stories in the list that you would like us to try and cover in sprint 2, please let us know. Thanks!

Noah Hanks

**Email 15:**

Hello,

Our sprint 2 demo is scheduled for this Wednesday (April 10) at 4PM. If you are able to join a skype call at this time, that would be great. If not, we can work out another time to do so. Thanks!

Noah Hanks

**Email 16:**

Sounds good, I'll be there.

**12.3 Source Folder Tree:**

+---lunasList

| | .gitignore

| | .ruby-version

| | config.ru

| | Gemfile

| | Gemfile.lock

| | package.json

| | Rakefile

| | README.md

| +---app

| | +---assets

| | | +---config

| | | | manifest.js

| | | +---images

| | | | .keep

| | | | defaultProfile.jpeg

| | | | LunasList.jpg

| | | | LunasList2.jpg

| | | | LunasList2.png

| | | | pawprint.jpg

| | | | pawprint2.jpg

| | | +---javascripts

| | | | | application.js

| | | | | cable.js

| | | | | events.coffee

| | | | | forums.coffee

| | | | | parks.coffee

| | | | | posts.coffee

| | | | | restaurants.coffee

| | | | | stores.coffee

| | | | | topics.coffee

| | | | | users.coffee

| | | | | vets.coffee

| | | | \---channels

| | | | .keep

| | | \---stylesheets

| | | application.scss

| | | events.scss

| | | forums.scss

| | | parks.scss

| | | posts.scss

| | | restaurants.scss

| | | stores.scss

| | | topics.scss

| | | users.scss

| | | vets.scss

| | +---channels

| | | \---application\_cable

| | | channel.rb

| | | connection.rb

| | +---controllers

| | | | application\_controller.rb

| | | | events\_controller.rb

| | | | pages\_controller.rb

| | | | parks\_controller.rb

| | | | pets\_controller.rb

| | | | pictures\_controller.rb

| | | | restaurants\_controller.rb

| | | | stores\_controller.rb

| | | | users\_controller.rb

| | | | vets\_controller.rb

| | | \---concerns

| | | .keep

| | +---helpers

| | | application\_helper.rb

| | | events\_helper.rb

| | | forums\_helper.rb

| | | parks\_helper.rb

| | | posts\_helper.rb

| | | restaurants\_helper.rb

| | | stores\_helper.rb

| | | topics\_helper.rb

| | | users\_helper.rb

| | | vets\_helper.rb

| | +---jobs

| | | application\_job.rb

| | +---mailers

| | | application\_mailer.rb

| | +---models

| | | | application\_record.rb

| | | | avatar.rb

| | | | event.rb

| | | | forum.rb

| | | | park.rb

| | | | pet.rb

| | | | picture.rb

| | | | post.rb

| | | | restaurant.rb

| | | | store.rb

| | | | topic.rb

| | | | user.rb

| | | | vet.rb

| | | \---concerns

| | | .keep

| | \---views

| | +---events

| | | edit.html.erb

| | | index.html.erb

| | | new.html.erb

| | | show.html.erb

| | | \_form.html.erb

| | +---layouts

| | | application.html.erb

| | | application.scss

| | | mailer.html.erb

| | | mailer.text.erb

| | +---pages

| | | etiquettetips.html.erb

| | | feedingtips.html.erb

| | | home.html.erb

| | +---parks

| | | edit.html.erb

| | | index.html.erb

| | | new.html.erb

| | | show.html.erb

| | | \_form.html.erb

| | +---pets

| | | edit.html.erb

| | | index.html.erb

| | | new.html.erb

| | | show.html.erb

| | | \_form.html.erb

| | +---pictures

| | | edit.html.erb

| | | index.html.erb

| | | new.html.erb

| | | show.html.erb

| | | \_form.html.erb

| | +---restaurants

| | | edit.html.erb

| | | index.html.erb

| | | new.html.erb

| | | show.html.erb

| | | \_form.html.erb

| | +---stores

| | | edit.html.erb

| | | index.html.erb

| | | new.html.erb

| | | show.html.erb

| | | \_form.html.erb

| | +---users

| | | | edit.html.erb

| | | | edit2.html.erb

| | | | index.html.erb

| | | | show.html.erb

| | | | \_form.html.erb

| | | +---confirmations

| | | | new.html.erb

| | | +---mailer

| | | | confirmation\_instructions.html.erb

| | | | email\_changed.html.erb

| | | | password\_change.html.erb

| | | | reset\_password\_instructions.html.erb

| | | | unlock\_instructions.html.erb

| | | +---passwords

| | | | edit.html.erb

| | | | new.html.erb

| | | +---registrations

| | | | edit.html.erb

| | | | new.html.erb

| | | +---sessions

| | | | new.html.erb

| | | +---shared

| | | | \_error\_messages.html.erb

| | | | \_links.html.erb

| | | \---unlocks

| | | new.html.erb

| | \---vets

| | edit.html.erb

| | index.html.erb

| | new.html.erb

| | show.html.erb

| | \_form.html.erb

| +---bin

| | bundle

| | rails

| | rake

| | setup

| | spring

| | update

| | yarn

| +---config

| | | application.rb

| | | boot.rb

| | | cable.yml

| | | credentials.yml.enc

| | | database.yml

| | | environment.rb

| | | puma.rb

| | | routes.rb

| | | spring.rb

| | | storage.yml

| | +---environments

| | | development.rb

| | | production.rb

| | | test.rb

| | +---initializers

| | | application\_controller\_renderer.rb

| | | assets.rb

| | | backtrace\_silencers.rb

| | | content\_security\_policy.rb

| | | cookies\_serializer.rb

| | | devise.rb

| | | filter\_parameter\_logging.rb

| | | inflections.rb

| | | mime\_types.rb

| | | thredded.rb

| | | wrap\_parameters.rb

| | \---locales

| | devise.en.yml

| | en.yml

| +---db

| | | schema.rb

| | | seeds.rb

| | \---migrate

| | 20190401174605\_create\_active\_storage\_tables.active\_storage.rb

| | 20190417173843\_add\_trails\_to\_parks.rb

| | 20190417173902\_add\_gated\_to\_parks.rb

| | 20190417173934\_add\_waterbodies\_to\_parks.rb

| +---lib

| | +---assets

| | | .keep

| | \---tasks

| | .keep

| | auto\_annotate\_models.rake

| +---log

| | .keep

| +---public

| | | 404.html

| | | 422.html

| | | 500.html

| | | apple-touch-icon-precomposed.png

| | | apple-touch-icon.png

| | | favicon.ico

| | | robots.txt

| | \---stylesheets

| | application.css

| +---storage

| | .keep

| +---test

| | | application\_system\_test\_case.rb

| | | test\_helper.rb

| | +---controllers

| | | .keep

| | | events\_controller\_test.rb

| | | forums\_controller\_test.rb

| | | parks\_controller\_test.rb

| | | posts\_controller\_test.rb

| | | restaurants\_controller\_test.rb

| | | stores\_controller\_test.rb

| | | topics\_controller\_test.rb

| | | users\_controller\_test.rb

| | | vets\_controller\_test.rb

| | +---fixtures

| | | | .keep

| | | | events.yml

| | | | forums.yml

| | | | parks.yml

| | | | pets.yml

| | | | pictures.yml

| | | | posts.yml

| | | | restaurants.yml

| | | | stores.yml

| | | | topics.yml

| | | | users.yml

| | | | vets.yml

| | | \---files

| | | .keep

| | +---functional

| | | forums\_controller\_test.rb

| | | posts\_controller\_test.rb

| | | topics\_controller\_test.rb

| | +---helpers

| | | .keep

| | +---integration

| | | .keep

| | +---mailers

| | | .keep

| | +---models

| | | .keep

| | | event\_test.rb

| | | forum\_test.rb

| | | park\_test.rb

| | | pet\_test.rb

| | | picture\_test.rb

| | | post\_test.rb

| | | restaurant\_test.rb

| | | store\_test.rb

| | | topic\_test.rb

| | | user\_test.rb

| | | vet\_test.rb

| | \---system

| | .keep

| +---tmp

| | .keep

| \---vendor

| .keep

+---Ruby on Rails Setup Guide

| 2019-02-19 13-00-59.flv

| 2019-02-19 22-37-54.flv

| Linux terminal commands.txt

| Preconfigured VM Setup.txt

| Youtube Video Links.txt