Rocket Launch Tracker

Project Proposal

Group 11

Trevor Staebel

Emory Lu

Ryan Parker

Michael Mengistu

Department of Computer Science and Engineering

Texas A&M University

October 18, 2020

Table of Contents

1 Executive summary

Purpose

The purpose of this project proposal is to request authorization to carry on this project of building a rocket launch tracker application.

Project Description

Rocket launches to space are happening all over the world, all the time, yet many people do not know about them. The amount of worldwide rocket launches are large, so it's not very easy for people to collect information about rocket launches schedules and locations with just one application; rocket launches are also highly weather-sensitive, so schedules of launches becoming out-dated is quite normal. In order to solve these problems, we created this project to design an application helping people to learn about rocket launches, obtain rocket launches updates easily, and schedule a visit accordingly.

This project will design and build an web-based application with a browser interface; it will incorporate at least three relevant and necessary existing web services: Spaceflight News API[1], Launch Library[2], Current Weather API[3], and Mapbox API[4]. Users should be able to access this application easily without using any extensions. In addition, this application will put accessibilities such as color blindness and limited fine-motor movements into consideration.

Project Structure

In order to carry on this project efficiently, we organized it into the following tasks:

1. Create user stories and user study proposals.
2. Implement this project over three "sprint" iterations.
3. Have a complete, working program at the end of each iteration.
4. Hold SCRUM meetings.

Expected Results

A complete application with a browser interface can be expected at the end of this project to solve the rocket launcher information accessing problem. This application should let users search for rocket launches, look up past launch videos, watch live videos of the launch, and look at relevant news articles. In conclusion, just with this application, users can obtain rocket launches information and schedule a visit accordingly without going to many different websites .

Project Timeline

If this project can be approved, our team will start the designing job on 19 October 2020 and use the agile approach throughout this project. The whole process will take approximately 5 weeks. Deliverables can be expected at the end of each "sprint" iteration, the final product can be expected on November 22, 2020, and a product presentation can be expected after November 24, 2020.

Summary

In this project proposal document, we will present a brief introduction to give an overview of this project, 2 detailed interface sketches, a description of project goals and scope, and discussions over potential issues and possible solutions. You can find all further details about this project in the following sections.

2 **Introduction**

2.1 Problem background

Rocket launches to space are happening all over the world, all the time, yet many people do not know about them. Rocket launches are very cool to witness and we believe everyone should witness how awesome launches are. Many launches are cancelled due to inclement weather, so providing a platform for status updates and launch times is a problem.

2.2 Needs statement

Rocket launches happen all the time yet many people either cannot find them or do not know about them. Even still, rocket launches can get delayed due to weather or other factors. A single, unified platform is needed for rocket enthusiasts to receive updates and find rocket launches. For this to be able to happen, we are planning to combine APIs that will give you all the information that you need to go see a rocket launch or to learn more about them.

2.3 Goal and objectives

The goal of this project is to build a web-based application named “Rocket Launch Tracker” which provides easy access to rocket launches relevant information such as launch schedule, locations, weather conditions, travel methods, and passed launches, so users can have a better understanding of rocket launches.

Objectives of this project are:

1. User-friendly interface: This application should be very easy to use. The interface should be concise and offer a reasonable amount of options. Users should be able to access information easily without having confusion.
2. Accessibility considered: This application should be accessible to all users, so we need to consider users with difficulty. Color blindness and limited fine-motor movements will be considered, and we will make adjustments in our design according to these.
3. Immediate updates: In order to work as a travel guide for users, we need to reflect immediate updates on all launch events, so users can adjust travel schedules accordingly if there are delays, location changes, or cancellations.
4. Good APIs usage: This application will use at least three existing web services, and these services need to be relevant and necessary.
5. Efficient search: This application should let users search for rocket launches, look up past launch videos, watch live videos of the launch, and look at relevant news articles.

2.4 Design constraints and feasibility

Our platform needs to provide information on rocket launches, travel information, and weather forecasts. The use of at least three APIs that find rocket launches, best travel information, weather, or any other information we may need is a design constraint. Implementing all the APIs onto a web page seamlessly onto a clean, user friendly interface could be a challenge. The need for accessibility for people with disabilities is also a requirement. We have chosen to implement technology on our project that helps the color blind and those with limited fine motor skills. As for the feasibility of the project, with the help of several well designed APIs and the creation of a good user interface, the main challenge will be combining the usage of each API and pivoting the design after any learned experiences.

3 Proposed work

3.1 Evaluation of alternative solutions

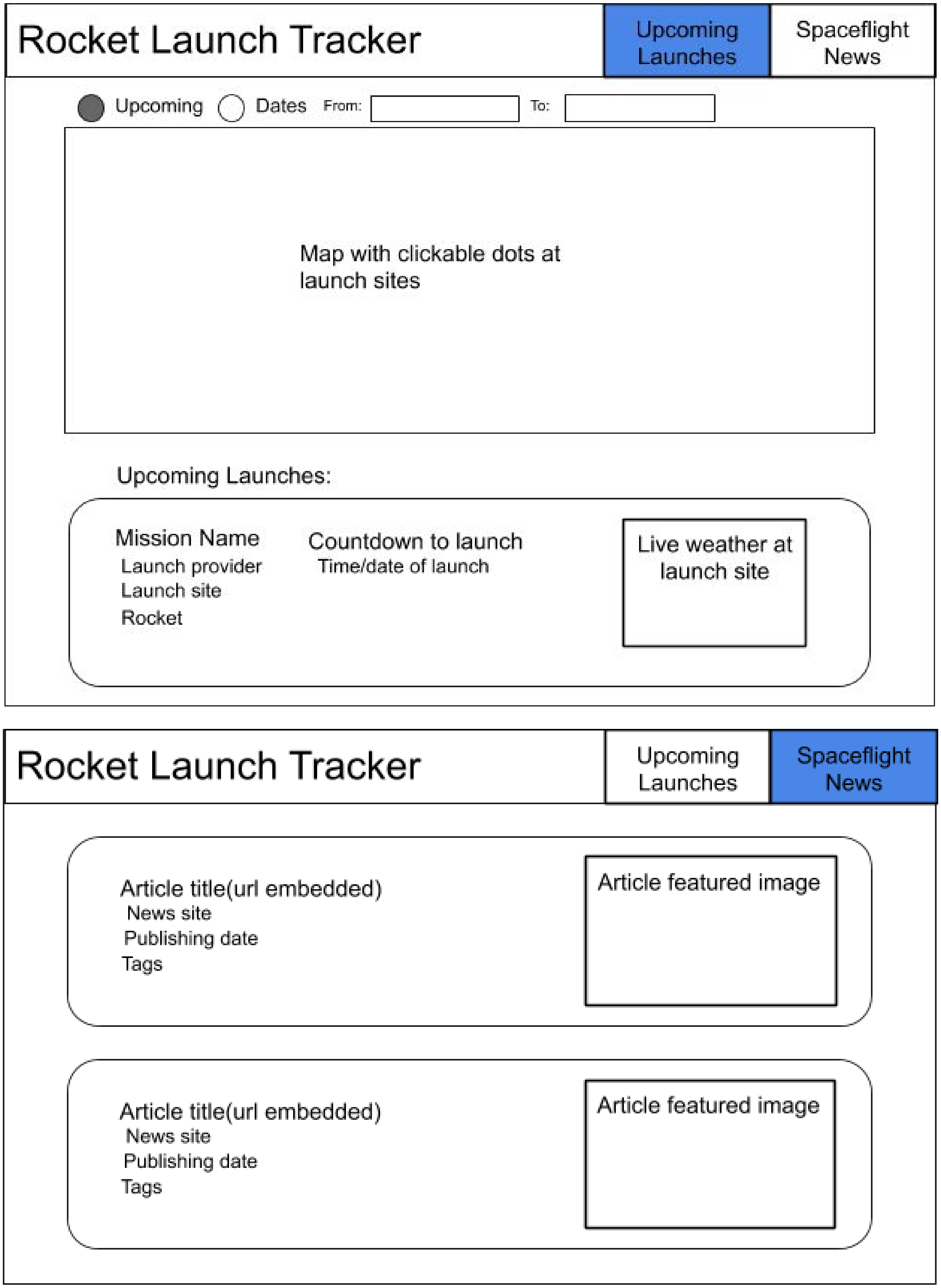
- Alternative solution 1: Delete “watch live videos of the launch” option.

  It’s always a great idea for users to watch live videos of rocket launches; however, this increases the difficulty of implementing. The general workflow of this project is to “glue” at least three existing web services together and form a fully functional application, which means the only way for the application to access data from third party is through a relevant API, and we can’t guarantee to find an API exactly meets our need and provides access to live launches videos. Therefore, if we can’t find a good API for this option, we might just delete it.

- Alternative solution 2: Delete “accessing weather conditions” option.

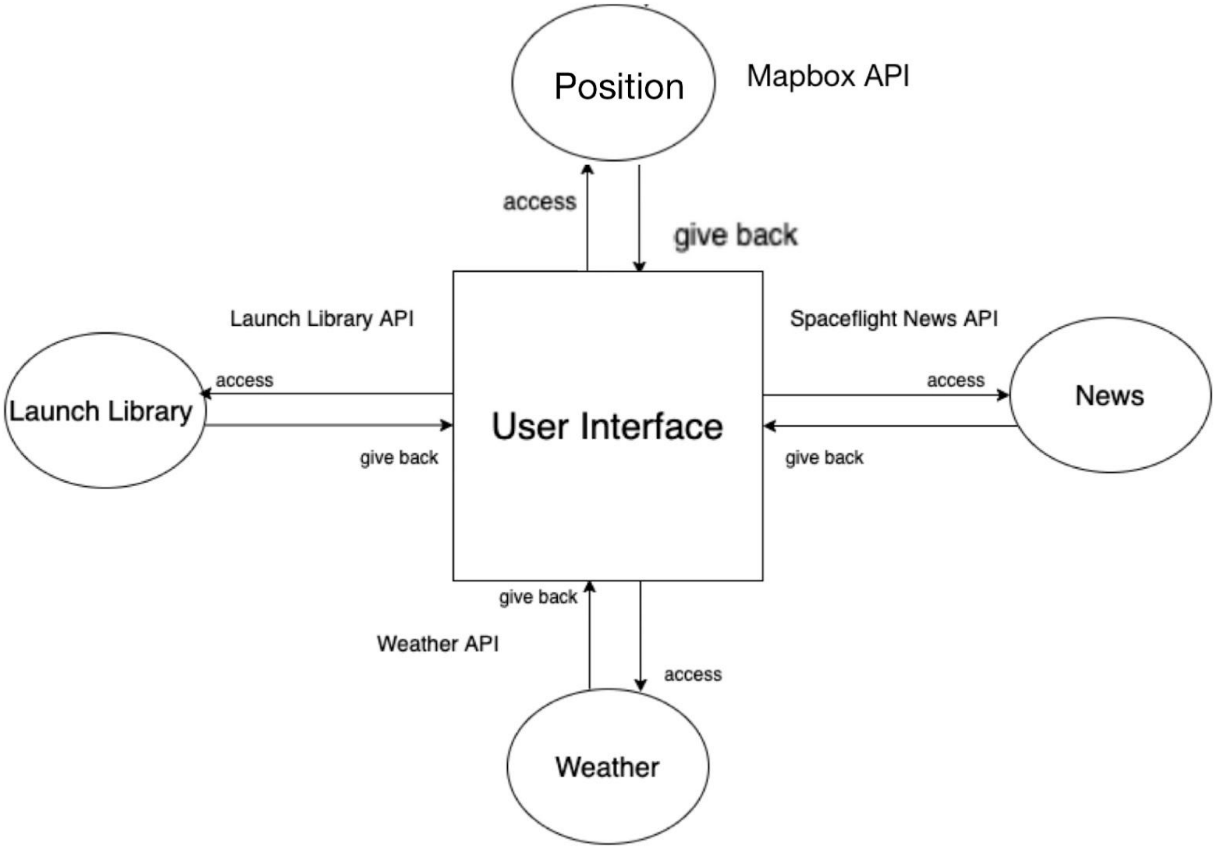
  For users who would like to schedule a visit to watch rocket launches, the most important information probably be the location, schedule, and immediate update of launch status--whether it is canceled or delayed. It would be nice to add a weather forecast functionality in our application, but it is not that necessary since weather applications are so common already. In later stages, if we can find a better functionality, we will use it instead.

3.2 GUI sketch



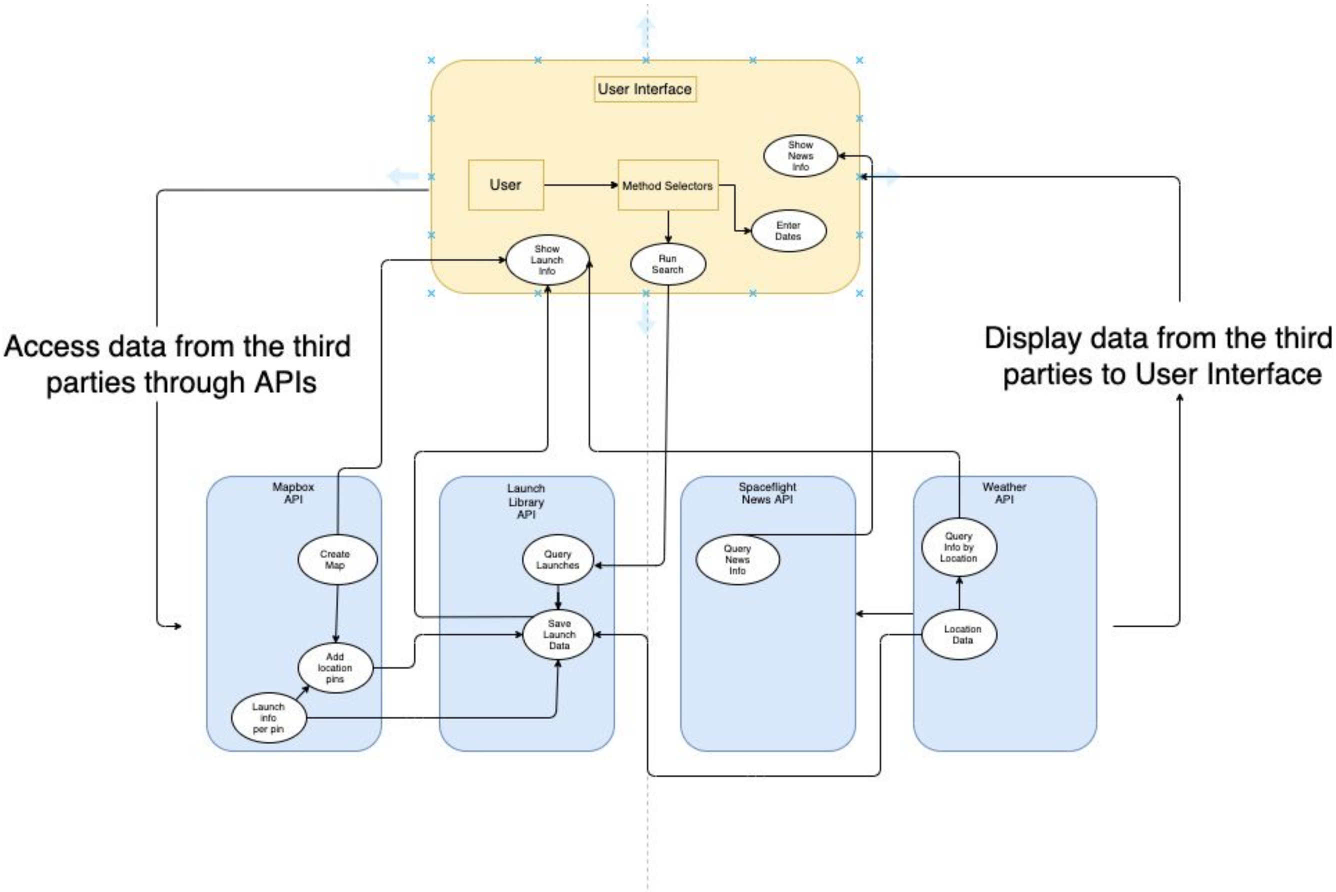
3.3 Design interface sketch

The general workflow of this application is that the application has a browser interface, and this application incorporates four existing services; those services can access data from the third parties and display information according to users' needs.



Specifically, the user interface will be a web page with search functionality for finding rocket launches. This search functionality will include sections for inputting a date range. Results will be displayed as pinpoints on a map and listed below the map. A stretch goal is to have each pinpoint display information of the associated launch when clicked. There will be a second page that lists news related to rocket launches and space. As of now, there will not be any search functionality added to the news page. We plan to embed a map into the webpage with the Mapbox API[4], this API should allow us to have a custom, clickable map that fulfills all of our design specifications.

Look further into the sketch, users can select different methods. The available options are looking for weather information, searching a launch by entering datas, locating launches on a map, and looking for relevant news/articles.



The sketch shows that the location data from the launch library API is used to add pins to the embedded map and also look up the weather information from the Weather API. The Spaceflight News API is used to find news related to rocket launches and the information from this API is added to a separate page from the rocket launch search system, which implements the Launch Library API, Mapbox API, and Weather API. All of which are consolidated onto the user interface after a search is made.

4 Engineering standards

4.1 Project management

Team Strengths

- Trevor Staebel- Can assist in making the user interface with html and css skills, as well as writing the code for an API call to one of the APIs we are using.
- Emory Lu- Can help in writing reports, writing user interface with html and css skills, design interface, coding for API calls, and usability testing.
- Ryan Parker- Plays the role of team manager, design and build user interface with html and css skills, create code for API calls, json parsing, and usability testing
- Michael Mengitsu - Can help with the software design and usability testing.

Management Mechanisms

Holding meetings right after a due date are used to administer and adjust responsibilities as well as brainstorm for the next section of the project if any is required. Throughout the week, updates from each member on their progress with their responsibilities can be shared over Slack. Any last minute development or combining of code can be done a day or two before the due date depending on how much work there is to do. After responsibilities for the current project section are clear, it is mostly up to each team member to do their part before the last meeting and due date. Most meetings will be conducted over zoom, however using the Slack group chat can work for specific scenarios or when the group is too quiet.

Work Schedule

- **10/19 -** Team meeting on Sprint 1 Materials
- **10/25 -** Sprint 1 Materials Due
- **10/26 -** Team meeting on User Stories and User Study Proposal
- **11/1 -** User Stories and User Study Proposal Due
- **11/2 -** Team meeting on Sprint 2 Completion and User Study Report
- **11/8 -** Sprint 2 Completion and User Study Report Due
- **11/9 -** Team meeting on Sprint 3 Completion
- **11/22 -** Sprint 3 Completion Due
- **11/23 -** Team meeting on Retrospection Survey and Report
- **11/24 -** Personal and Team Retrospective Survey and Reports Due
- **11/29 -** Team Meeting on Demo and Presentation
- **11/30 -** Practice for Demo and Presentation
- **12/1 -** Project Demo and Presentation

4.2 Planned Scope

APIs

- Spaceflight News API [1].
- Launch Library API [2].
- Current Weather API [3].
- Mapbox API[4].

Interactions

- Search system for finding rocket launches over specific time period or location
- Map that pinpoints the results from the search, clicking on a point will show more information. Including weather and possibly travel information.
- List of rocket launches and associated information to supplement map
- Separate page dedicated to news about rocket launches

Fallback Goals

- If creation of the map proves to fail or not be up to our expectations, just list out each launch and its associated information.
- If creation of the rocket launch search system proves to take too long, design of the spaceflight news page should be cut out.

Stretch Goals

- Designing of the Spaceflight news after a system for finding upcoming launches is created.
- Showing information on each rocket launch after clicking on a point on the map
- Selecting a launch will provide articles related to that launch

4.3 Product and sprint backlog (1 page)

The entire project can be divided into several sub tasks. These include building the upcoming launches page, building the spaceflight news page, creating the map and its attached capabilities, and attaching the API's to their respective places. A good portion of tasks needed to be completed are designing and building the site itself without any capability of connecting to the API's. This way when we do connect the API we have somewhere to display the information. Knowing this to build our product we will first create the basic site before we move on to connect the API's necessary for displaying information. To begin with our overall project has forty-three tasks which we expect to grow as we discover more tasks that must be completed to achieve a finished product. This means we expect to accomplish around seven tasks per week to begin with.

Our first sprint is a shorter one and will consist of fifteen tasks that are all rather quick.These tasks are the work to create the basic webpage to contain launch information. After building the launch information page the spaceflight news page requires minimal work to also be created. Sprint 2 is where we will get into working with API's and connecting them to the pages. Sprint 3 is some more work adding the map to the page and finishing displaying all the information acquired through the API's

Sprint 1

1. Assemble the basic framework of the launch information page
2. Create navigation bar on launch information page
3. Create box at top center of page to display the map
4. Create input box to select number of launches to display
5. Create checkbox to select upcoming launches
6. Create checkbox to select launches in a date range
7. Create input box for the "from" date
8. Create input box for the "to" date
9. Place label saying "Upcoming launches" next to selectable button for upcoming launches
10. Place label next to date range button that says "Date Range"
11. Place label next to from box saying "From"
12. Place label next to to box saying "to"
13. Assemble basic framework of spaceflight news page
14. Create navigation bar on spaceflight news page
15. Host site on server

Our overall product backlog, backlogs for sprints 2 and 3, and our overall and sprint 1 burndown charts have been attached to the "**Appendices**" section.

5 References

[1] "Spaceflight News API", *spaceflightnewsapi.net*, ver. 1. Nov.27, 2019. [Online]. Available: https://spaceflightnewsapi.net/api/v1/#api-Info-GetInfo. [Accessed: Oct. 14, 2020].

[2] "Launch Library API", *launchlibrary.net*, ver. 1.4.1. [Online]. Available: https://launchlibrary.net/docs/1.4.1/api.html. [Accessed: Oct. 14, 2020].

[3] "Current Weather API", *Weatherbit.io*. [Online]. Available: https://www.weatherbit.io/api/weather-current. [Accessed: Oct. 18, 2020].

[4] "Mapbox API", *docs. mapbox.com.* [Online]. Available: https://docs.mapbox.com/api/. [Accessed: Oct. 18, 2020].

6 Appendices

6.1 Product backlog

Include product backlog as it is defined in this phase. This should be a living document throughout the project and updated approximately 3 times per week.

User stories:

1. As a user I can select to see upcoming launches
2. As a user I can select a range of dates to see launches between
3. As a user I can see a map showing locations of launches
4. As a user I am able to select a location on the map and see upcoming launches from there
5. As a user I am able to select a location on the map and see launches within a date range from that location
6. As a user I can see the weather at the site of an upcoming launch
7. As a user when I click somewhere on the map where there isn't a launch site it shows launches from all sites
8. As a user when I select upcoming launches the map will update to upcoming launches and only upcoming launches will display
9. As a user when I select upcoming launches I will be able to select how many upcoming launches I see
10. As a user when I load the site I will be able to see launches
11. As a user I can easily view everything on the site

Sprints and subtasks:

1. Develop class to display a layout for a launch gotten from the API
2. In launch layout class place the mission name at the top left
3. In launch layout class place the launch provider under mission name
4. In launch layout class place launch site name under launch provider
5. In launch layout class place the type of rocket under the launch site name
6. In launch layout class place box to contain the current weather at the launch site
7. Create countdown clock function for launch layout
8. In launch layout class place the countdown clock in the top center
9. In launch layout class place the launch time and date below the countdown clock
10. Connect map API to launch information page
11. Create box at top center of page to display the map
12. Create input box to select number of launches to display
13. Create checkbox to select upcoming launches
14. Create checkbox to select launches in a date range
15. Create input box for the "from" date
16. Create input box for the "to" date
17. Develop function to create a dot on the map
18. Create function to determine if a dot has been selected
19. Create function to get launch locations and place dots on the map
20. Create function to display launches from selected dot(launch site)
21. Host site on server

22. Create function to process launch API data
23. Create function to determine if upcoming launches or a date range is selected
24. Assemble the basic framework of the launch information page
25. Assemble basic framework of spaceflight news page
26. Place label saying “Upcoming launches” next to selectable button for upcoming launches
27. Place label next to date range button that says “Date Range”
28. Place label next to from box saying “From”
29. Place label next to to box saying “to”
30. Connect weather API to launch layout class
31. Create function to get current weather at a given launch site
32. Create function to display the current weather at launch site
33. Create class to contain article information
34. Create function to retrieve article data from Spaceflight News API
35. Create function to process received article data into article objects
36. Create function to add new articles to the spaceflight news page
37. Create function to add new launches to the upcoming launches page
38. In article class place article title with article url embedded in top left
39. In article class place news site below article title
40. In article class place article published date below news site
41. In article class place article tags below published date
42. In article class display article image on the right side
43. Create navigation bar on spaceflight news page
44. Create navigation bar on launch information page
45. Have links to launch finder and spaceflight news on the navbar
46. Make an API call to Launch Library API.

Sprint 1

1. Assemble the basic framework of the launch information page
2. Create navigation bar on launch information page
3. Have links to launch finder and spaceflight news on the navbar
4. Create box at top center of page to display the map
5. Create input box to select number of launches to display
6. Create checkbox to select upcoming launches
7. Create checkbox to select launches in a date range
8. Create input box for the “from” date
9. Create input box for the “to” date
10. Place label saying “Upcoming launches” next to selectable button for upcoming launches
11. Place label next to date range button that says “Date Range”
12. Place label next to from box saying “From”
13. Place label next to to box saying “to”
14. Assemble basic framework of spaceflight news page
15. Create navigation bar on spaceflight news page
16. Host site on server

Sprint 2

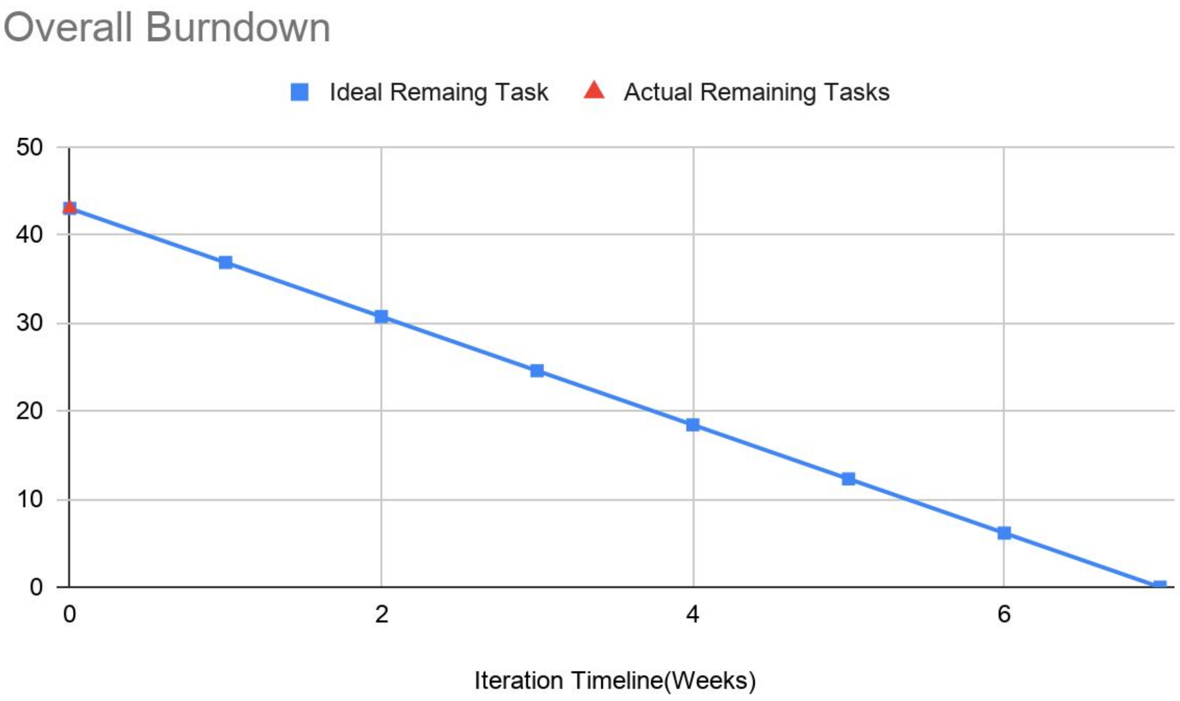
1. Develop class to display a layout for a launch gotten from the API
2. Make an API call to Launch Library API.

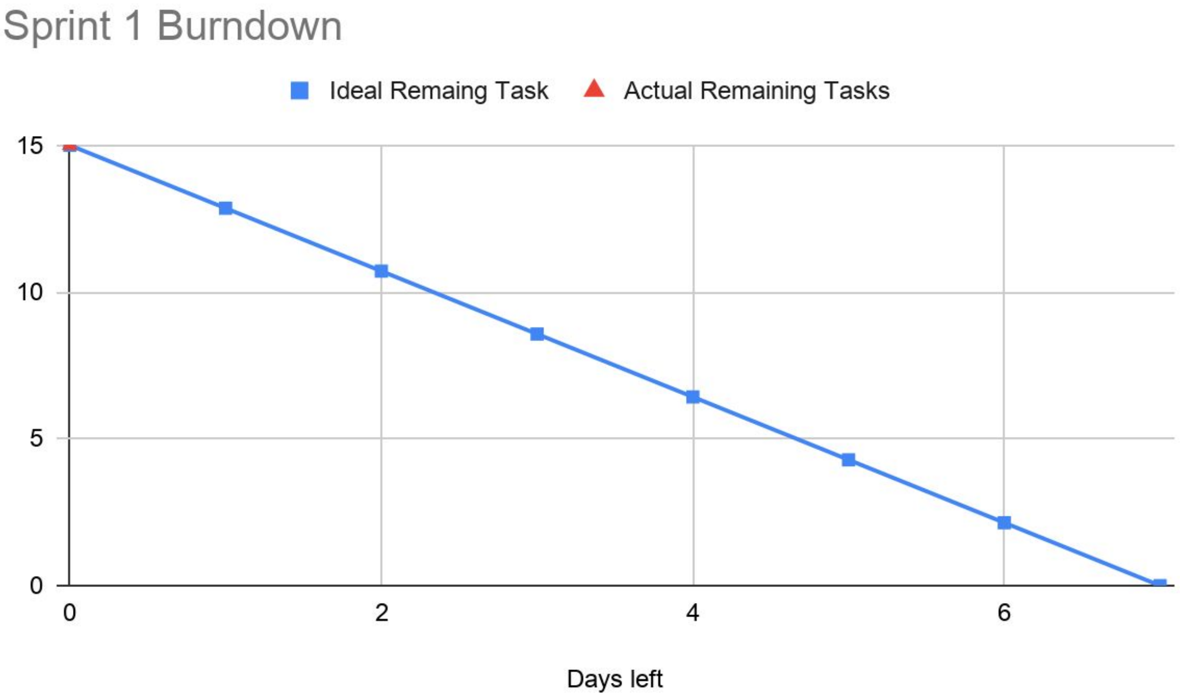
3. In launch layout class place the mission name at the top left
4. In launch layout class place the launch provider under mission name
5. In launch layout class place launch site name under launch provider
6. In launch layout class place the type of rocket under the launch site name
7. In launch layout class place the countdown clock in the top center
8. In launch layout class place the launch time and date below the countdown clock
9. In launch layout class place box to contain the current weather at the launch site
10. Create countdown clock function for launch layout
11. Assemble basic framework of spaceflight news page
12. Create navigation bar on spaceflight news page
13. Create class to contain article information
14. In article class place article title with article url embedded in top left
15. In article class place news site below article title
16. In article class place article published date below news site
17. In article class place article tags below published date
18. In article class display article image on the right side

Sprint 3

1. Connect map API to launch information page
2. Develop function to create a dot on the map
3. Create function to determine if a dot has been selected
4. Create function to get launch locations and place dots on the map
5. Create function to display launches from selected dot(launch site)
6. Create function to process launch API data
7. Create function to determine if upcoming launches or a date range is selected
8. Connect weather API to launch layout class
9. Create function to get current weather at a given launch site
10. Create function to display the current weather at launch site
11. Create class to contain article information
12. Create function to retrieve article data from Spaceflight News API
13. Create function to process received article data into article objects
14. Create function to add new articles to the spaceflight news page
15. Create function to add new launches to the upcoming launches page

6.2 Burn-down chart


Overall Burndown
■ Ideal Remaing Task
▲ Actual Remaining Tasks
Iteration Timeline(Weeks)


Sprint 1 Burndown
■ Ideal Remaing Task
▲ Actual Remaining Tasks
Days left