Openspace and Climate Reanalyzer Interface
System Requirements Specification

Table of Contents

1. Introduction

This project, titled *Climate Reanalyzer, is made available to planetariums through OpenSpace and other software platforms* is an encompassing capstone project for the University of Maine Computer Science students on V-Model Violets (formerly known as Team Violet). This project will be developed in collaboration with Dr. Sean Birkel, a Research Assistant Professor at the Climate Change Institute, and Sean Laatsch, the Director of the Versant Planetarium at the University of Maine. This project will also be collaborating with Dr. Carter Emmart and Micah Acinapura from the OpenSpace team. The goal of this project is to create an image tile server consisting of data from the Climate Reanalyzer, which can be used by OpenSpace and in planetariums or other research settings.

1.1 Purpose of This Document

The purpose of this System Requirements Specification (SRS) document is to define and describe the requirements (functional and non-functional) for the Openspace and Climate Reanalyzer Interface. This document will serve as a reference for our team, the clients, and the teaching staff for the Capstone class, outlining the functionality of the system, possible constraints, and expectations regarding performance, security, and design. The SRS document is intended for:

- Capstone Team Members: To lay out the project requirements to ensure the client's needs are met
- Clients: To ensure that the project is on the right track
- University of Maine Faculty: To assess the progress of the project

1.2. References

The table below lists the sources being used for this project be used currently:

Title	Author(s)	Publisher / Organization	Date	Link / Citation
Climate Reanalyzer	Dr. Sean Birkel	Climate Change Institute, University of Maine	Ongoing	https://climatereanalyzer.org/
OpenSpace	OpenSpace Team	American Museum of Natural History	Ongoing	https://www.openspaceproject.com/
Apache HTTPD Server Documentation	Apache Software Foundation	Open Source	1997-Current	https://httpd.apache.org/

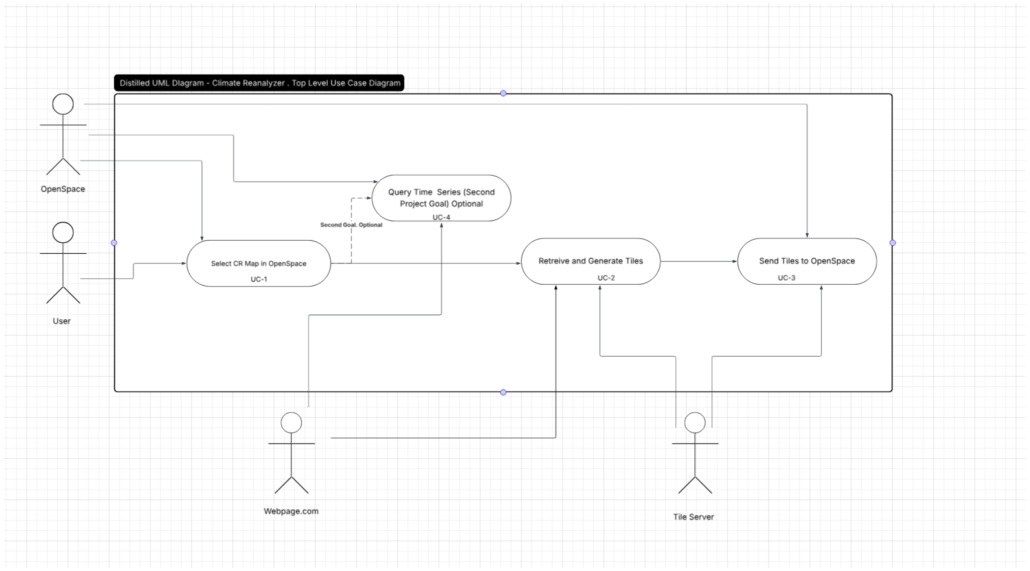
Software Engineering (10th Edition)	Ian Sommerville	Pearson Education	2015	Sommerville, I. (2015). Software engineering (10th ed.). Pearson Education.

1.3. Purpose of the Product

The Climate Reanalyzer OpenSpace interface project is set out to make the data gathered and used on the University of Maine's Climate Reanalyzer available to use in OpenSpace planetarium software. This allows for coverage of data that OpenSpace may not have readily available yet. The incorporation of the Climate Reanalyzer into the OpenSpace system will allow for immersive and interactive displays of historical climate data across the globe in a planetarium setting. This product will allow for data such as temperature, climate change data, atmospheric pressure, and much more to be overlaid onto a model of Earth that OpenSpace has integrated into their software. The product will address both missing data or inconsistencies in existing data to provide a more accurate model of Earth in its current state and in previous generations. In conclusion, the product will help enhance science education for teaching and learning.

1.4. Product Scope

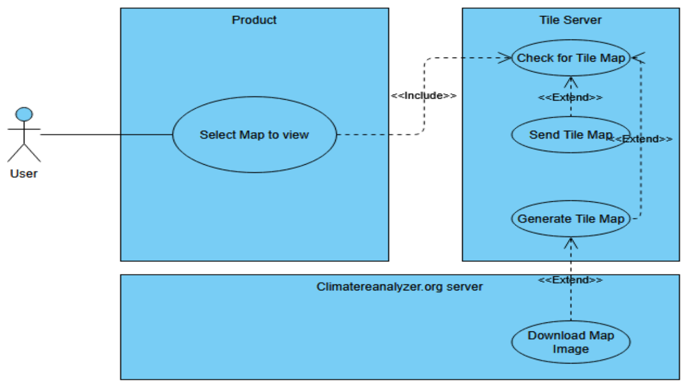
The Climate Reanalyzer (CR) system will provide a connection between the Climate Analyzer web platform and the OpenSpace visualization software that is used in planetariums. Our primary goal is to make global climate and weather data from the CR visual on display for planetariums and other large digital environments.


Distilled UML Diagram - Climate Reanalyzer . Top Level Use Case Diagram
OpenSpace
User
Query Time Series (Second Project Goal) Optional
UC-4
Second Goal. Optional
Select CR Map in OpenSpace
UC-1
Retreive and Generate Tiles
UC-2
Send Tiles to OpenSpace
UC-3
Webpage.com
Tile Server

This Diagram shows the overall scope of the system and how it will connect to the actors. The Interface will allow users to select and load climate maps from the webpage. The system will get that information, convert the map data into tiles that can be sent to OpenSpace for viewing. The optional feature UC-4 will let the system show time series data from the CR which is the second project goal. This diagram shows the inside system boundary and what's the external.

2. Functional Requirements

Use Case #1: Map-Tile Conversion



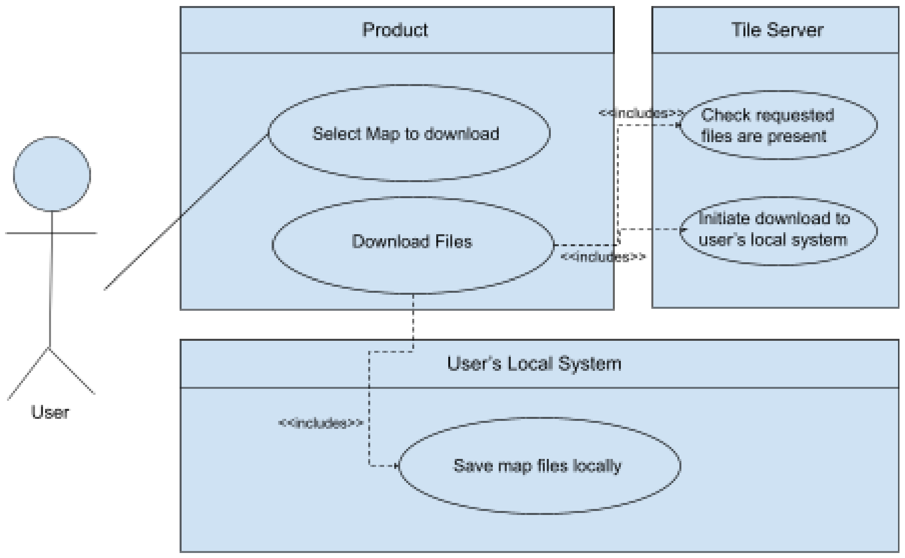
Number	1	
Name	Map - Tile Conversion	
Summary	The system shall allow Openspace to display a tile map generated from a 2x1 map image when requested.	
Priority	5	
Preconditions	A user must request a map image	
Postconditions	The map image will be displayed within Openspace.	
Primary Actor	User	
Secondary Actors	Climatereanalyzer.org image data, Tile Server, Openspace, system	
Trigger	When the user selects a map to load.	
Main Scenario	**Step**	**Action**
	1	The user requests a new map.
	2	The System checks if that map is already available.
	3	The system sends requested tile data to openspace for display.
Extensions	**Step**	**Branching Action**
	2a	If not already available, The system downloads the map image.
	2b	The system calls for Gdal and Apache to generate a tile map

	2c	The system stores the tile map on the tile server.
Open Issues	Should the System be automatically downloading and storing all data from Climate Reanalyzer? Should the system only download new maps to the server only if the map is requested directly?	

Test for Functional Requirement #1

To test this requirement, we will write some test code that will simulate a call from a user for a specific map. If the map is sent correctly, or if it is downloaded and stored correctly on the server. The test will pass, otherwise fail.

Use Case #2: Downloading and Saving Tile Files



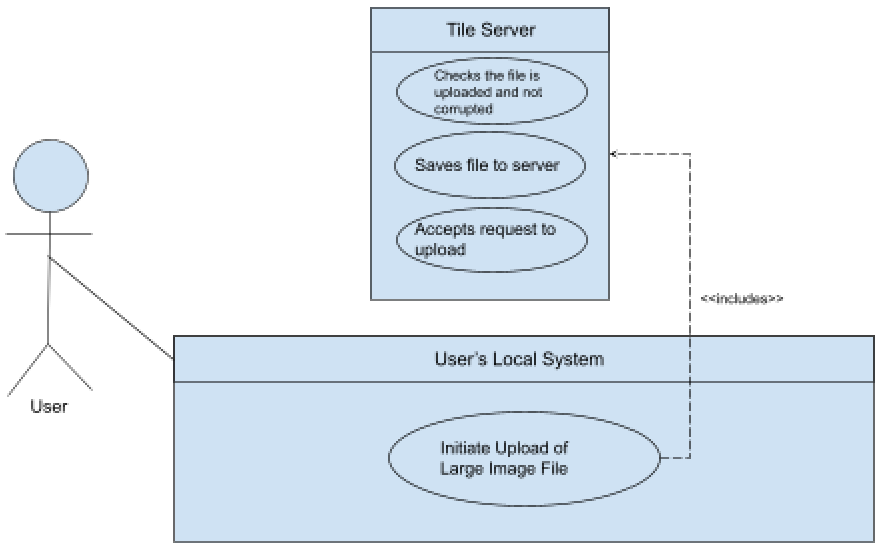
Number	2
Name	Download and Save Tile Files
Summary	The system should be able to download and save files containing tiles, allowing them to be used in the Openspace simulation on demand.
Priority	2
Preconditions	● The system has a connection to the internet. ● There are files present in the tile server ● There is sufficient space on the user's system for the download
Postconditions	● The selected tiles are successfully saved to local storage. ● The system can access and load the saved tiles on demand without needing to redownload them.
Primary Actor	User

Secondary Actors	Tile server, OpenSpace system	
Trigger	When the user initiates the request to download tiles to use them offline.	
Main Scenario	**Step**	**Action**
	1	The user selects the tiles to download.
	2	The System connects to the tile server
	3	The system downloads the files from the tile server and saves them locally to the user's device.
Extensions	**Step**	**Branching Action**
	2a	If the download fails due to the connection to the tile server, the system will alert the user
	3a	If the download fails (due to connection loss), the system pauses and will alert the user
	3a	If the download fails due to storage issues, the system alerts the user to the insufficient space and prompts them to clear space or choose a different storage location.
Open Issues	If the download fails, should the system automatically attempt to resume the download once connection has been restored, or should it wait for another prompt from the user?	

Test for Functional Requirement #2

To test this requirement, a request will be sent to the system to download a specific set of tiles for later use in the OpenSpace simulation. Once the request is made, the system should connect to the remote server, retrieve the requested tile files, and store them in the designated local directory. After the download process completes, the storage location will be checked to verify that the tile files exist and are accessible. If the files exist and are accessible, the test passes, otherwise, the test fails.

Use Case #3: Admins Uploading Large Data Files

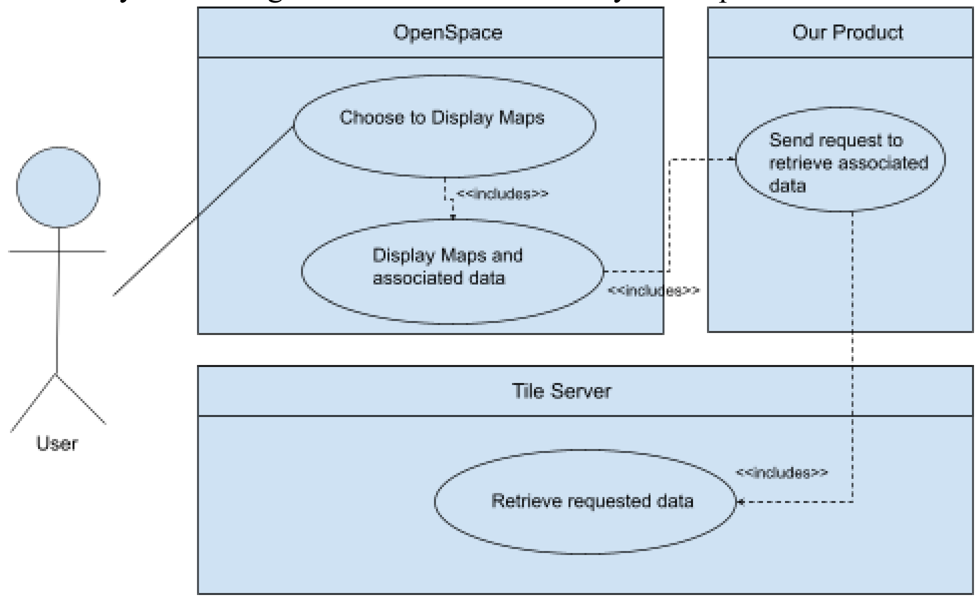

Tile Server
Checks the file is uploaded and not corrupted
Saves file to server
Accepts request to upload
<<includes>>
User
User's Local System
Initiate Upload of Large Image File

Number	3	
Name	Admins Uploading Large Data Files	
Summary	The system shall allow administrators to upload large source images to be used by the tile server.	
Priority	4	
Preconditions	● The system has a connection to the internet. ● There are files present on the local machine ● There is sufficient space on the server to upload	
Postconditions	● The selected tiles are successfully saved on the server. ● The system can access and load the uploaded image.	
Primary Actor	Admin	
Secondary Actors	Tile server	
Trigger	When the user initiates the request to download tiles to use them offline.	
Main Scenario	**Step**	**Action**
	1	The admin finds a large image file to upload
	2	The admin connects to the server
	3	The admin uploads the file to the server
Extensions	**Step**	**Branching Action**
	2a	If the upload fails due to the connection to the tile server, the system will alert the user
	3a	If the upload fails (due to connection loss), the system pauses and will alert the user
	3a	If the upload fails due to storage issues, the server alerts the admin to the insufficient space.
Open Issues	If the upload fails should the server save the upload progress or should it delete progress until there is sufficient space?	

Test for Functional Requirement #3

To test this requirement, we will attempt to upload an image to the server. Then the server will be checked to see if the images are present. If they are, the test passes. If the images are not present, the test fails.

Use Case #4: The System listing available Climate Reanalyzer Maps

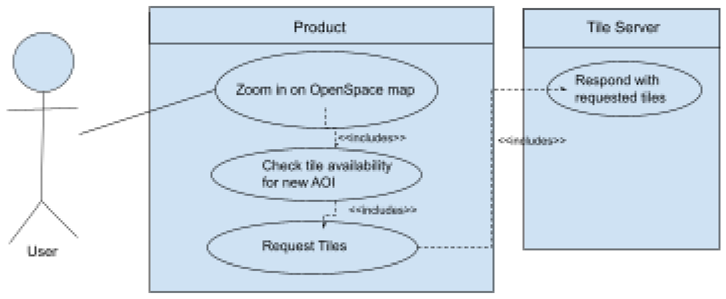


Number	4	
Name	List Available CR Maps	
Summary	The system shall expose a list of available Climate Reanalyzer maps with the layer name, date/time, resolution. So the presenters can pick one in OpenSpace	
Priority	4	
Preconditions	● The tile server is reachable ● CR list is available	
Postconditions	● Map layer is selected ● Meta data is cached for retrieval.	
Primary Actor	User	
Secondary Actors	Tile server, OpenSpace, ClimateReanalyzer.org	
Trigger	The user opens the “Select map” .	
Main Scenario	**Step**	**Action**
	1	User requests list of maps
	2	System fetches map catalog
	3	System returns a list, user selects one.
Extensions	**Step**	**Branching Action**
	2a	If CR is not reachable. The system will serve the last known local catalog.
Open Issues	Is this possible? Can it auto refresh.	

Test for Functional Requirement #4

To test this requirement, we will create a mock CR and serve it a catalog. We will then confirm that the program lists accurate data and selection returns the correct layer ID. If it does so, the test passes, otherwise the test fails.

Use Case #5: Multiple tile levels



Number	5	
Name	Multiple tile levels	
Summary	The system shall provide inner (zoomed) layers of tiles upon request	
Priority	4	
Preconditions	● The server must successfully connect with Openspace ● The map must be loaded in Openspace ● The user must zoom in to trigger new tile requests	
Postconditions	● Tile layers will be provided while there is sufficient resolution in the CR image	
Primary Actor	User	
Secondary Actors	Tile server, OpenSpace, Climatereanalyzer.org	
Trigger	When the user interacts with the OpenSpace interface and zooms in	
Main Scenario	**Step**	**Action**
	1	The user zooms in on the map in Openspace
	2	Openspace requests the necessary tiles from the tile server
	3	The tile server responds with the requested image tiles
Extensions	**Step**	**Branching Action**
	2a	If there are more tile levels that can be loaded, openspace requests these from the tile server
	2b	If no further tile levels remain, openspace will not load zoomed data
Open Issues	Should the system store all of the tile levels individually, or calculate image cropping and transformation upon request of a new tile level?	

Test for Functional Requirement #5

To test this requirement, we will visit server URLs specifying the inner tile layers that should be available and verify that an image is returned. This should call the server and request a tile, which the server should recognize and identify as an available layer, with which it will respond. If the image is displayed at the URL, the test passes, otherwise, the test fails.

3. Non-Functional Requirements

Non-Functional Requirement #1

The application should be able to run smoothly while being used by up to 1000 users simultaneously.

Priority - 3

Test: Run a stress test of many uses at once over a short period of time. If everything works as planned it will pass, otherwise fail.

Non-Functional Requirement #2

The application should be able to download maps and tiles from the database for offline use.

Priority - 2

Test: Attempt to download multiple maps and open them from Openspace. If the maps are downloaded and opened successfully it will pass, otherwise fail.

Non-Functional Requirement #3

The application should have an intuitive and self-explanatory user interface so it is user-friendly.

Priority - 5

Test: Perform usability testing with a handful of various people and gauge their understanding of the program's operation through observation and post-test questions. If the vast majority of people (at least 85%) can understand with little assistance from the proctor, the program passes, otherwise, it fails.

Non-Functional Requirement #4

The application should be able to run both with and without an internet connection.

Priority - 2

Test: Test that the base functions of the application works both when connected to wifi and with no wifi or service connection.

Non-Functional Requirement #5

The application should be able to handle errors such as corrupted files or missing data gracefully.

Priority - 3

Test: Provide corrupted data to the system and confirm that the exceptions are handled correctly.

Non-Functional Requirement #6

The application's codebase should be clean and easy to understand to allow for maintainability

Priority - 1

Test: Read through all code to determine readability. You can also provide code reviews to ensure simplicity and good code quality

Non-Functional Requirement #7

The application should work with the target OpenSpace versions and run on the ARCSIM-provided Linux VM.

Priority - 4

Test: Install ARMSIM Vm validate it works with OpenSpace.

Non-Functional Requirement #8

The tile service must be ready within 20 seconds after the process starts on the ARCSIM VM.

Priority - 3

Test: Restart the service, measure the time it takes to start.

Non-Functional Requirement #9

All source code shall be publicly accessible on GitHub

Priority - 3

Test: Verify that any GitHub user is able to access the repository holding the source code

Non-Functional Requirement #10

The application shall be available 99% of the time

Priority - 4

Test: Log tile server up time for a week through requests at regular intervals and calculate if availability is 99% or better

4. User Interface

See User Interface Design Document for *Openspace and Climate Reanalyzer Interface*

5. Deliverables

The following deliverable items will be supplied as described below to the clients, Dr. Sean Birkel and Shawn Laastch, throughout the development of the specified software. Most deliverable items will be made available as both a hard copy and a digital pdf.

Item	Format	Date of Delivery
Systems Requirement Specification	-Document Hard Copy -Document Digital Copy	10/29/2025
System Design Document	-Document Hard Copy -Document Digital Copy	11/17/2025
User Interface Design Document	-Document Hard Copy -Document Digital Copy	12/3/2025
User Manual	-Document Hard Copy -Document Digital Copy	5/8/2026
Administrator Manual	-Document Hard Copy -Document Digital Copy	5/8/2026
Copies of all Biweekly Status Reports	-Document Hard Copy -Document Digital Copy	Every two weeks- Mondays starting 10/20/2025
All source code	-GitHub Repository	5/8/2026
The executable program	-Web Application -Open Space Interface	5/8/2026

6. Open Issues

The following are issues that have been identified and not yet concluded. These issues will be addressed later in the development process.

- Tile server availability
- Function to process images from climate reanalyzer into tiles
- Function to select desired tiles based on area of interest and available data
- WMS file to store data for each map to be loaded into Openspace
- Openspace asset files for each map to be loaded
- JS webpage that can be opened within Openspace to browse CR catalog

Appendix A – Agreement Between Customer and Contractor

The team at V-Model Violets and our clients, Dr. Sean Birkel, Shawn Laatsch, and the team at The Openspace Project, all agree that the product to be delivered will be a web-based platform that interfaces with Openspace to display tiled map data gathered from Climatereanalyzer.org. The product will be delivered on May eighth, which is after our COS 497 ends next semester. The product is being created for free as part of our capstone coursework, and while we will work diligently, the end product may not include all the features discussed.

Our team at V-Model Violets will continue to monitor the product in our free time, but after the completion of the course, the product will officially be completed and delivered as-is. However, as the product being created will be open source after completion, maintenance of the software will fall into the hands of anyone who wishes to contribute or improve on the work we complete. Our names and contact information will stay present on the site if we need to be contacted, and we would all be willing to support fixes that need to be done.

Signed
V-Model Violets Team:

Emily Scott:____ **Date:**_____/_____/_____

Brianna Gannet :____ **Date:**_____/_____/_____

Brennon Tiensivu: Brennon Tiensivu **Date:** 10/18/2025

Hunter Savarese:____ **Date:**_____/_____/_____

Matthew Keast: ____ **Date:**_____/_____/_____

Clients:

Dr. Sean Birkel:____ **Date:**_____/_____/_____

Shawn Laatsch:____ **Date:**_____/_____/_____

Dr. Carter Emmart:____ **Date:**_____/_____/_____

Micah Acinapura:____ **Date:**_____/_____/_____

Appendix B – Team Review Sign-off

We, the team at V-Model Violets, all agree and sign off on all the information provided in this document. The document has been reviewed by all members, and we agree on its content, and it is formatted in a way we all agree on. No one in the team has any major issues with the document or its layout, but they will have the chance to list some minor things they disagree with.

Signed
V-Model Violets Team:

Emily Scott:____ Date:_____/_____/_____

Comments:____

Brianna Gannet :______________________________________ Date:_____/_____/_____

Comments:____

Brennon Tiensivu: Brennon Tiensivu **Date:** 10/18/2025

Comments: N/A

Hunter Savarese:______________________________________ Date:_____/_____/_____

Comments:____

Matthew Keast: ______________________________________ Date:_____/_____/_____

Comments:____

Appendix C – Document Contributions

Emily Scott's contributions: Added 1 Functional Requirement and associated diagrams and tests. Added 2 Non-Functional Requirements and associated tests. Completed the content and formatting of the appendices. Wrote an outline of the document for internal use and assigned team roles for the document's creation. 20% Contribution.

Brianna Gannet's Contributions: Added 1 Functional Requirement and associated diagrams and tests. Added 2 Non-Functional Requirements and associated tests. Worked as an overall proofreader and Editor for the team. As Client Liaison, provided input on each section to make sure it was what the client wanted. 20% Contribution.

Brennon Tiensivu's Contributions: Added 1 Functional Requirement and associated diagrams and tests. Added 2 Non-Functional Requirements and associated tests. Completed the text for the sections from the introductions to 1.3. Created the team GitHub, which is linked in the document's references. 20% Contribution.

Hunter Savarese's Contributions: Added 1 Functional Requirement and associated diagrams and tests. Added 2 Non-Functional Requirements and associated tests. Completed section 1.4 and its associated diagram. 20% Contribution.

Matthew Keast's Contributions: Added 1 Functional Requirement and associated diagrams and tests. Added 2 Non-Functional Requirements and associated tests. Completed the text for sections 4, 5, and 6. 20% Contribution.