Software Requirements Specification

for

MoonTrek: Telescope Augmented Reality

Version 1 approved

Prepared by:

Nadir Abdusemed

Jackson Bentley

Jesus Cruz

Youssef Elzein

Derek Guevara

Joe Hineno

Rich Ho

Owen Ramirez

Salman Sheikh

Alex Sherzai

Table of Contents

Revision History

Name	Date	Reasons for changes	Version
First Draft	11/30/2022	Template Creation	1.0
Update	12/1/2022	Update document from previous years	1.1
Fall Final Date	12/7/2022	Completed documentation with updated information	1.3

1. Introduction

1.1 Purpose

The purpose of the software requirements specification document is to give an informative description of the MoonTrek: Telescope Augmented Reality, hosted by Jet Propulsion Laboratory. This document is intended to inform the audience of the software, including its purpose and features, the application's interface, what the system will do, and the constraints it will operate on.

1.2 Intended Audience and Reading Suggestions

The software requirement specification document is proposed to the general audience and users who are involved in the development of MoonTrek: Telescope Augmented Reality development. The intended audience can be software developers, project advisors, liaisons, team managers, or whoever is involved with the development of MoonTrek for upcoming years. This document does not need to be read in a specific sequential manner. Users are encouraged to jump to any section they find relevant.

- **Introduction**
  - This section offers a summary of the MoonTrek project, including purpose, scope definitions/abbreviations, and references.
- **Overall Description**
  - This section provides readers with explanations of general facts that affect the software and its requirements. This section will give a high-level description of the requirements.
- **External Interface Requirements**
  - This section offers details on how the MoonTrek software interacts with any external interfaces.
- **Requirements Specifications**
  - This section contains all of the necessary software requirements with enough detail to allow designers to accurately design the software to satisfy those requirements and to allow testers of the software to verify that all requirements have been satisfied. The requirements should include a description of every input to the system, every output, and all functions performed by the system in response to an input or output.
- **Other Nonfunctional Requirements**
  - This section specifies any numerical or statistical requirements imposed on the software such as the number of terminals to be supported, the number of simultaneous users to be supported, and the amount and type of information to be handled.
- **Other Requirements**
  - This section discusses any legal and/or ethical issues involved in the project.

1.3   Product Scope

In this section:

- MoonTrek will allow the user to utilize their telescope to analyze their images of the Moon. MoonTrek will process the user's image and then provide annotations on the uploaded image.
- The software is delivered in a user-friendly format that is designed to give the user a smooth experience. Users can view their annotated images on both laptops and smartphones. Users can also connect their telescopes and have real-time data overlay.

1.4   Definitions, Acronyms, and Abbreviations

SRS   - Software Requirements Specification

JPL   - Jet Propulsion Laboratory

UI   - User Interface

MTTAR - MoonTrek: Telescope Augmented Reality

AR - Augmented Reality

1.5   References

"ASCOM - Standards for Astronomy." Ascom-Standards.org,

ascom-standards.org/. Accessed 10 Dec. 2021.

Nasa.gov, 2016,

trek.nasa.gov/tiles/apidoc/trekAPI.html?body=moon.

Accessed 10 Dec. 2021.

"OpenCV: OpenCV-Python Tutorials." Docs.opencv.org,

docs.opencv.org/master/d6/d00/tutorial_py_root.html.

"Three.js – JavaScript 3D Library." Threejs.org, 2019,

threejs.org/.

"Closest-Point API | Planet-Vector API" JPL,

http://54.157.167.17:5000

2. Overall Description

2.1 System Analysis

    A. The goal of this project is to create a web application that allows users to see the features of the moon. A user will upload their own picture of the moon and the application will pinpoint different points of interest such as craters as landing sites.

    B. A major issue was comparing the user's picture to an actual picture of the moon.

    C. In order to compare and match the picture given and the original picture of the moon, we first turn the pictures to black and white. Then we use SIFT to find common key points and match them. We also have a file with coordinates of the different moon's features and points of interest.

2.2 Product Perspective

The MoonTrek telescope will use JPL's data from its Moon Trek portal. The user will also need a telescope or a device that can capture a high-quality picture of the moon.

The application will do something that has not been done before, which is to display an image of the Moon that a user enters and display different points of interest of the Moon based on the picture that is uploaded.

The application will need a timestamp of when the image was uploaded and based on the timestamp, it will replicate the position of the Sun and Earth relative to the Moon.

2.3 Product Functions

- Display an image that the user takes from their telescope or device.
- The application will flatten out the image and change the color/dimensions.
- Compare the user's image to MoonTrek's moon
- The application will display and pinpoint features and points of interest of the moon.
- A 3D model including the Moon, Earth, and Sun will be displayed.

2.4 User Classes and Characteristics

The user classes that belong to Moon Trek include but are not limited to students, professors, and general users. This software is mainly intended for those who seek access to various pieces of interesting/informative Moon data.

- Images from the telescope are routed to their smartphone or laptop.

- These routed images will be annotated with lunar features such as landing sites, craters, and maria.
- Users will have the ability to explore their moon capture with easily accessible data from the comfort of their own devices.

2.5 Operating Environment

The application lives in a web-based application that is accessible from any computer or smartphone. The application will run on a Linux server that is part of JPL.

We are using Vue and Python for this application.

2.6 Design and Implementation Constraints

Security: We must keep JPL's data secure. For this reason, we cannot expose the code that has their information.

Runtime: Using software like Unity would be helpful, but such software is very large and heavy. Therefore, we must abstain from using software that could be helpful but is too large.

Time - We have a limited time constraint. This project has been going on for years, and we currently have less than six months to complete it.

Data - We don't have enough pictures of the moon in order to test our application.

2.7 User Documentation

Within the Moon Trek telescope, there will be a "Manuals" button that when pressed will display details on how to use the application. The point of these tutorials will be to provide step-by-step examples to make the experience of using the application easier.

2.8 Assumptions and Dependencies

This software will require the user to have a valid internet connection. We can also assume that the user will have a reliable and up-to-date computer.

2.9 Apportioning of Requirements

- Website where people from around the world can upload their pictures of the moon.

- Connectivity of a telescope to a mobile application implementation.
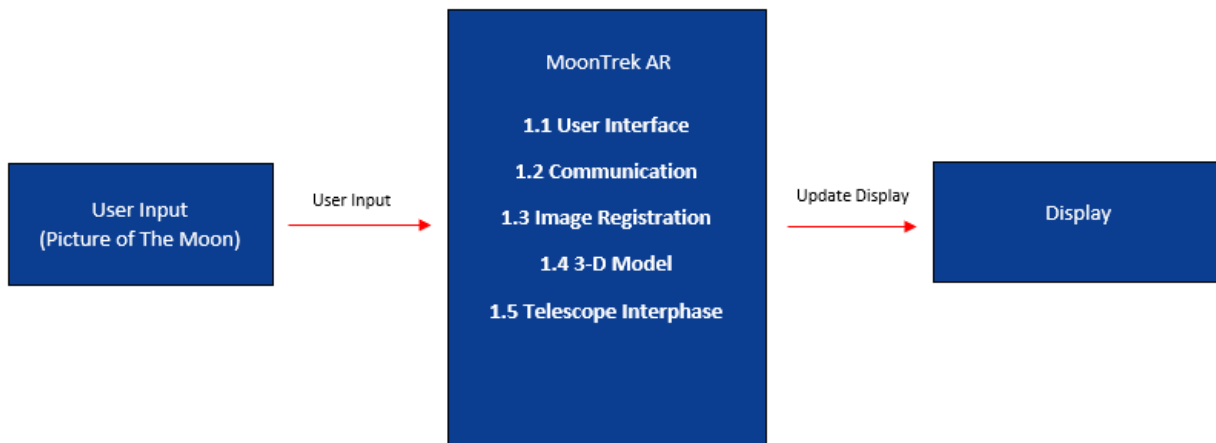
3.   External Interface Requirements

Figure 1: MoonTrek Data Flow Diagram 0 (DFD 0)

3.1   User Interfaces

A. **Initialization**
  i.   **Request:** The user will request the Moon Trek website and the initial landing page will be displayed.
  ii. **Response:** Moon Trek website will be displayed to the user.
B. **Web Home Page**
  i.   **Logo:** Clicking on it will take the user back to the home page.
  ii. **Navigation bar:** Displays the different links for different types of pages.
  iii. **Choose file / Upload:** This gives the user the functionality to upload an image.
C. **Moon Trek Services:**
  i.   **Moon image:** The website will display images from coordinates provided in the image/telescope and by connecting to our model.
  ii. **Moon information services:** The website will provide statistical data like location, vector, and nearest point information from image/telescope coordinates.
  iii. **Moon overlay services:** The website will prove utilities to overlay informative images onto previous Moon images.

3.2   Hardware Interfaces

In the future the application will integrate with a telescope to capture images directly.

3.3   Software Interfaces

- **Express.js API Source:** <u>Express.jshttps://expressjs.com</u>
  - ○ Back-end web application framework for building RESTful APIs with Node.js.
- **Vue.js As Source:** <u>https://vuejs.org/</u>
  - ○   open-source model–view–ViewModel front-end JavaScript framework for building user interfaces and single-page applications.

- **Node.js, version 19.2.0 Source:** <u>https://nodejs.org/en/</u>
  - ○   runs on the V8 JavaScript Engine and executes JavaScript code outside a web browser.
- **Opencv-contrib-python, version 3.4.2.17 or 4.5.0. Source:** <u>https://opencv.org/</u>
  - ○   Allows the implementation, execution, and testing of the circle detection program.
- **Threejs, version r123. Source:** <u>https://threejs.org/</u>
  - ○   Implement the back-end 3D environment representing the Moon, Earth, and Sun.
- **Numpy, version 1.20.1. Source:** <u>https://numpy.org/</u>
  - ○   Numerical computer tools for our calculations.
- **Pillow, version 8.1.2. Source:** <u>https://pillow.readthedocs.io/en/stable</u>/
  - ○   Library to allow us to work with images.
- **Moon Trek API:**
  - ○   Executed through API calls listed below:
  - ○   Nearest Point
    - ■   Find the nearest point on a target body to an observing body.
  - ○   Planet Vector Search
    - ■   Find the xyz positions of planets relative to one observer/planet.
  - ○   Nomenclature Call
    - ■   Query nomenclature for a given region on the Moon.
  - ○   Latitudinal to Rectangular
    - ■ Transform latitudinal coordinates to rectangular space.

3.4   Communications Interfaces

- **Moon Trek application portal**
  - The application shall require an internet connection to access the web portal
- **Moon Trek project image registration**
  - The application shall receive user images via image submission form in the program interface
- **Web Browser**
  - The application shall run on these web browsers: Microsoft Edge, Firefox, Chrome.
- **HTTPS communication standard**
  - The application shall use HTTPS to maintain the privacy of submitted data.
  - The application shall use HTTPS to maintain the safety of recorded data
- **Issues**
  - No communication security issues detected, so far.
  - No encryption issues detected, so far.

4.   Requirements Specification

4.1   Functional Requirements

1. The application shall display a 3D model, along with distance ratios, for the Moon model
2. The application shall take in a Moon visual captured by a telescope.
3. The application shall query and display different data from the MoonTrek API and return specific details such as craters, landing sites, and maria.
4. The application shall be able to overlay layers and change opacity for visualization of a 3D Moon model.
5. The application should allow users to obtain accurate longitude and latitude for any pixel on the Moon's 2D surface.
6. The application should allow users to have the ability to explore their moon capture with easily accessible data from the comfort of their own devices.

4.2   External Interface Requirements

- The application shall interact with the telescope provided by Cal State LA

4.3   Logical Database Requirements

The MoonTrek application database contains references to images uploaded by users. Each Moon layer will be distinguished through the images in the database. These layers include "Maria", "Craters", and "Landing sites" and information that is hardcoded within the application. Each item's coordinates shall be specified correctly, the application will then display a mark on the location.

4.4   Design Constraints

- **Standard limitation**
- **Hardware limitation**

System must have access to the internet, and the application must run on a browser within a smartphone or laptop.

5.   Other Nonfunctional Requirements

5.1   Performance Requirements

This software only handles one image at a time. The necessary calculations are performed on the back-end, and this will be based on the user's wait time. The user's wait time is determined by how good their internet connection speed is when they upload their image to our server. In the United States, on average, the internet speed is 40 megabits per second. This equates to 5 megabytes per second of upload speed. The user's average image size should be between 3 to 15 megabytes. So, the average wait time will be between 0.6 seconds to 3 seconds per image.

5.2   Safety Requirements

It is extremely important for users to understand that the images they upload will be stored on a database on a server. With that, all users should also understand the inherent risks of having their data compromised. There is the possibility of a breach occurring, which means that their data will be exposed and taken advantage of. So, all users should be wary and fully understand the risks and consequences of sharing content on the internet.

This is an application that uses users' lunar images, so that requires users to go outdoors to take pictures of the moon. As COVID-19 restrictions have eased, it is still a safety precaution that users should take. Users should follow safety precautions such as wearing a mask in areas that require them. Another safety precaution that users should keep in mind is where they are taking pictures. Users should be full of where they plan to take pictures. Since there are no requirements or prerequisites that the user needs to follow, they would have to make smart decisions based on their judgment of whether a certain area that they plan on taking a picture of is safe to do so or not.

5.3   Security Requirements

There are no security or privacy issues that are needed to use our software. There is no need for the user to provide any login or any information about the user in general.

5.4   Software Quality Attributes

The following quality characteristics are required by the software to properly use and are required by both the users and developers:

- Availability:
    - The MoonTrek software shall be available 24 hours a day, 7 days a week on the web application
- Flexibility:
    - The MoonTrek software shall perform correctly for any lunar image that is uploaded by any user.
- Reliability:
    - The MoonTrek software shall perform various geometric calculations that is necessary throughout the process to accurately display information on the user's image of the Moon.
- Responsiveness
    - The MoonTrek software shall load and display the necessary features to the user within a couple of seconds.
- Simplicity:
    - The MoonTrek software shall include a tutorial for the user to follow and can be read at any time to understand certain features of the software.
- Usability:
    - The MoonTrek software shall be capable of being used effectively, efficiently, and safely.

5.5   Business Rules

As of now, there are none.

6.   Legal and Ethical Considerations

When considering designing and implementing this web application there are three broad legal and ethical considerations we must take into account, which are privacy, copyright, and data security. In order to ensure privacy we will only maintain the minimum amount of data in order to have a functional application. The only data which will be uploaded from the users will be the pixel data and metadata regarding the location and time the image was taken at. Additionally, this data will not be stored on ours or JPL's servers unless there is a violation of our terms of use, in which case some data will be stored as a reference. Additionally, data will not be packaged and shared with any third parties. The users must also agree to

only upload images that are their own property so as to not violate any copyright laws. If copyright is violated we must implement methods that will remove this data from being processed. Methods should also be implemented to prevent repeat offenders from accessing the application. The security of JPL's data and the user's data must also be maintained, as such methods to compartmentalize and keep the data security must be implemented

Appendix A: Glossary

SIFT - Scale-Invariant Feature Transform: Feature detection algorithm that helps locate local features and key points in an image. This is used for image registration and to match two different pictures.

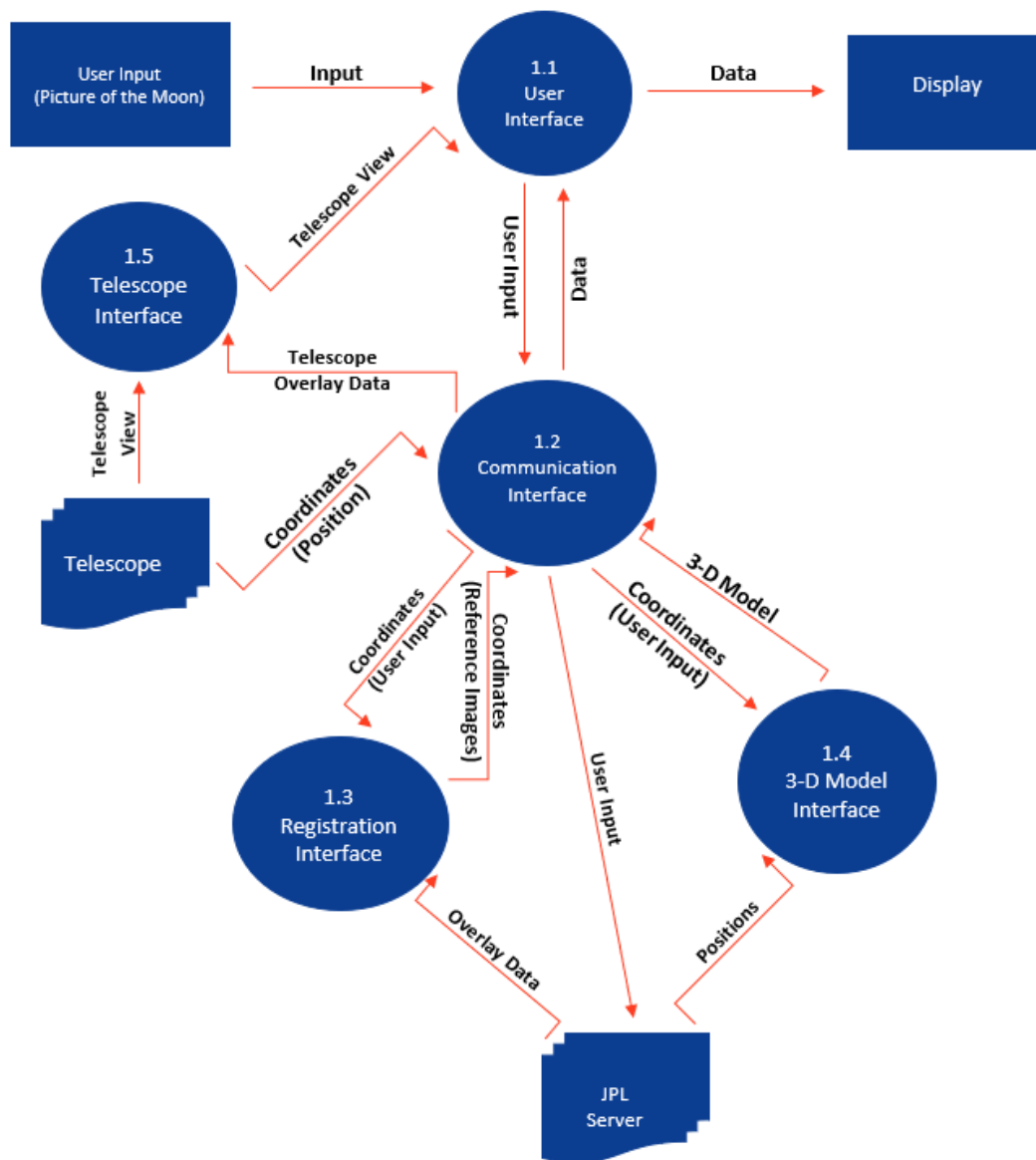| Acronym	Long Version
SRS	Software Requirement Specifications
UI	User Interface
SDS	Software Design Specification
JPL	Jet Propulsion Laboratory
MTTAR	MoonTrek: Telescope Augmented Reality
AR	Augmented Reality
SIFT	Scale-Invariant Feature Transform

Appendix B: Analysis Models

Figure 2: MoonTrek Data Flow Diagram 1 (DFD 1)