

Software Requirements Specification

for

MoonTrek Telescope

Version 1 approved

Prepared by
Kevin Aguilera
Pavit Chawla
Jacob Fausto
Alex Lamb
Nicolas Ojeda
Albert Ramirez
Gerard Rosario
Elvira Sakalenka
Dakota Townsend

NASA Jet Propulsion Laboratory (JPL)

05/14/2021

Table of Contents

Table of Contents.....	pg #2
Revision History.....	pg #3
1. Introduction.....	pg #4
1.1. Purpose.....	pg #4
1.2. Intended Audience and Reading Suggestions.....	pg #4
1.3. Product Scope.....	pg #5
1.4. Definitions, Acronyms, and Abbreviations	pg #5
1.5. References.....	pg #5
2. Overall Description.....	pg #6
2.1. System Analysis	pg #6
2.2. Product Perspective.....	pg #7
2.3. Product Functions.....	pg #7
2.4. User Classes and Characteristics.....	pg #7
2.5. Operating Environment.....	pg #8
2.6. Design and Implementation Constraints.....	pg #8
2.7. User Documentation.....	pg #8
2.8. Assumptions and Dependencies.....	pg #8
2.9. Apportioning of Requirements.....	pg #9
3. External Interface Requirements.....	pg #10
3.1. User Interfaces.....	pg #10
3.2. Hardware Interfaces.....	pg #10
3.3. Software Interfaces.....	pg #11
3.4. Communications Interfaces.....	pg #11
4. Requirements Specification.....	pg #12
4.1. Functional Requirements.....	pg #12
4.2. External Interface Requirements.....	pg #12
4.3. Logical Database Requirements.....	pg #12
4.4. Design Constraints.....	pg #12
5. Other Nonfunctional Requirements.....	pg #13
5.1. Performance Requirements.....	pg #13
5.2. Safety Requirements.....	pg #13
5.3. Security Requirements.....	pg #13
5.4. Software Quality Attributes.....	pg #14
5.5. Business Rules.....	pg #14
6. Legal and Ethical Considerations.....	pg #15
Appendix A: Glossary.....	pg #16
Appendix B: Analysis Models.....	pg #17
Appendix C: To Be Determined List.....	pg #18

Revision History

Name	Date	Reason For Changes	Version
First Draft	10/20/20	Adding/Removing Information	1.0
Final Draft	05/14/21	Updating and Adding Information	2.0

1. Introduction

1.1 Purpose

The main purpose of this document is to present a detailed description of the MoonTrek Telescope program version 2.0. It will cover all aspects of the software including the purpose and features, the interface of the application, what the system will do, and the constraints under which it will operate.

1.2 Intended Audience and Reading Suggestions

While this software requirement specification document is written for a more general audience, this document may be directed towards individuals more involved in the development of MoonTrek Telescope. This includes software developers, project advisors, liaisons, and team managers. This document need not be read sequentially; users are encouraged to jump to any section they find relevant.

- Part 1 – Introduction
 - This section offers a summary of the MoonTrek project, including purpose, scope definitions/abbreviations, and references.
- Part 2 – 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.
- Part 3 – External Interface Requirements
 - This section offers details on how the MoonTrek software interacts with any external interfaces.
- Part 4 – 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.
- Part 5 – 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.
- Part 6 – Other Requirements
 - This section discusses any legal and/or ethical issues involved in the project.

1.3 Product Scope

The scope of this document includes the following:

- The MoonTrek Telescope software will provide an interface that bridges the use of the telescope used to look at the moon to return an annotated image routed to a smartphone or laptop and will encourage amateur astronomers with easily accessible interesting knowledge.
- The software will be accessible through a user friendly and efficiently designed application. This software should be accessible through a smartphone or laptop and will act intuitively to encourage the interest in astronomy of its users.

1.4 Definitions, Acronyms, and Abbreviations

SRS - Software Requirements Specification

SDD - Software Design Document

JPL - Jet Propulsion Laboratory

UI - User Interface

WAC – Wide Angle Camera: often used with lunar photography

LROC - Lunar Reconnaissance Orbiter Camera: often used with lunar photography

1.5 References

Macias, Jose Recommended Template for Software Requirements Specifications

ASCOM Standards <https://ascom-standards.org/>

MoonTrek API: <https://trek.nasa.gov/tiles/apidoc/trekAPI.html?body=moon>

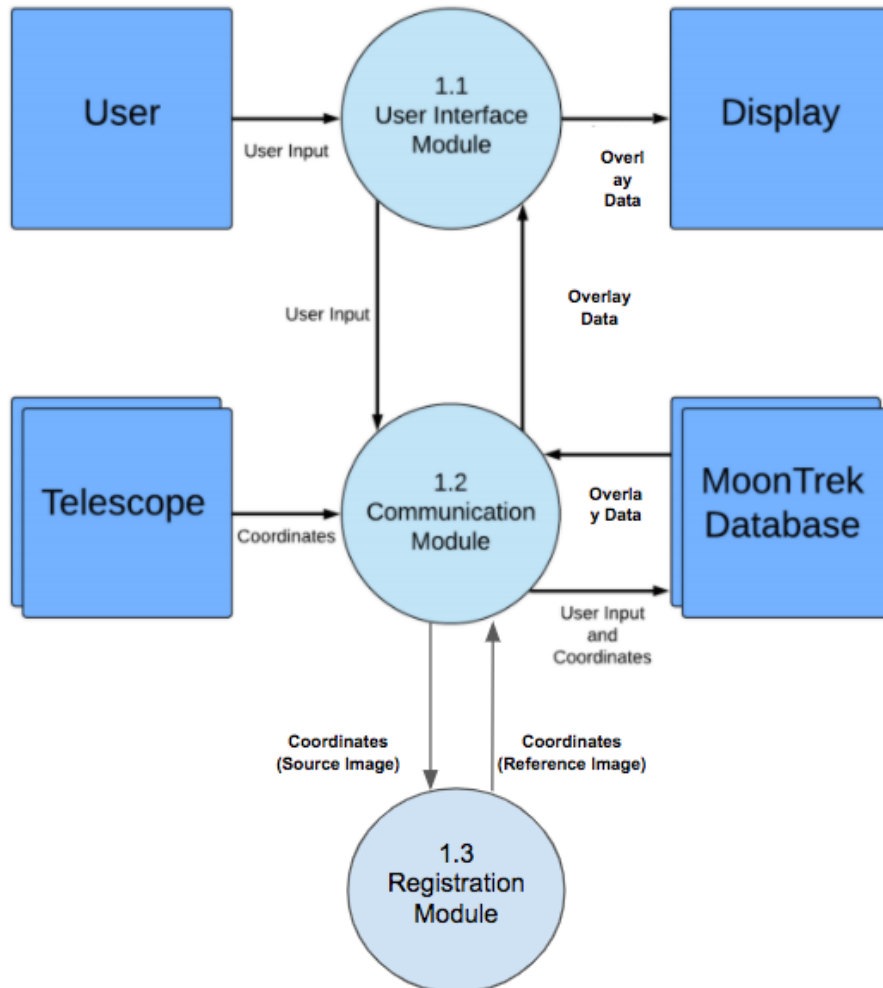
Django Web Framework: <https://docs.djangoproject.com/en/3.1/>

OpenCV : https://docs.opencv.org/master/d6/d00/tutorial_py_root.html

ThreeJS : <https://threejs.org/docs/>

2. Overall Description

DFD Level 1



2.1 System Analysis

- The MoonTrek application should bridge the gap between the provided telescope image and anyone's smartphone or computer. It will allow anyone interested to explore the moon with easily accessible knowledge at their fingertips.
- The major technical hurdle that presented itself during the beginning stages of development was the ability to accurately register the user's image into a reference image to properly obtain the coordinates of the moon at any given pixel of the moon. The system should be able to pinpoint the correct latitude and longitude at any given pixel in the user's moon. The difficulties are successfully implemented through the implementation of circle detection , and image registration to correctly identify pictures of the moon were solved by testing out a variety of parameters until better results were found.

2.2 Product Perspective

The Moon Trek Telescope is intended to work with a telescope and JPL's Moon Trek Portal . The application is very similar to the Moon Trek portal in the sense that it will provide similar data and overlaying features . The main difference is that the image of the moon will be routed from a telescope capture instead of just the Lunar Reconnaissance Orbiter (LRO) Wide Angle Camera (WAC) Mosaic . The goal is to utilize the data already present in the Moon Trek portal , such as layers and be able to use it in the user's image of the moon , which could be routed from a telescope .

This software is dependent on two outside products, a digital telescope or any device where an image of the moon can be captured and NASA's MoonTrek Portal which contains already large data sets of the moon.

Similar to cameras of phones , the image coming from the telescope will be routed to the applications interface . Then the application's dependency to provide an accurate layer to the user will require proper image registration to a 3D model constructed with the LRO WAC Mosaic rectangular map of the moon from JPL's portal.

The application will be a first of its kind in the sense that it will be the first interface for amateur astronomers combining both a telescope moon captures and JPLs databases.

2.3 Product Functions

- Display an image of the moon from the telescope's perspective.
- Register the source image (image from telescope) to the reference image (JPL's LRO WAC Mosaic)
- Provide statistical data based on the geographical portions of the moon.
- Overlay informative annotations over the current portion of the moon they are observing.
- Provide a generic 3D model to which the user can interact and add layers.

2.4 User Classes and Characteristics

The user classes that belong to MoonTrek 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 mare.
- Users will have the ability to explore their moon capture with easily accessible data from the comfort of their own device.

2.5 Operating Environment

Operating environment for the MoonTrek Telescope Program will be a web based application accessible from any laptop or smartphone.

- Client/Server System: Django
- Database: SQLite

2.6 Design and Implementation Constraints

Limited Time as briefly alluded to developing an app for experienced professionals takes a lot of time, and even they have trouble meeting deadlines. Our application will be due by the end of the year, and we still have less time and less experience than professionals have.

Familiarity with Specific Data Types - When brainstorming about certain aspects of our application, we thought about Data Structures that we knew about conceptually, and knew would get the job done best, but had never used in actual practice.

Limited access to equipment for data collection due to COVID-19 has affected our ability to obtain the necessary data for our project to display the moon in a wider range of areas.

2.7 User Documentation

Within the Moon Trek Telescope, there will be a “Manuals” button that when pressed will display detailing how to use the application. The tutorials will provide step-by-step examples of how to use the application.

2.8 Assumptions and Dependencies

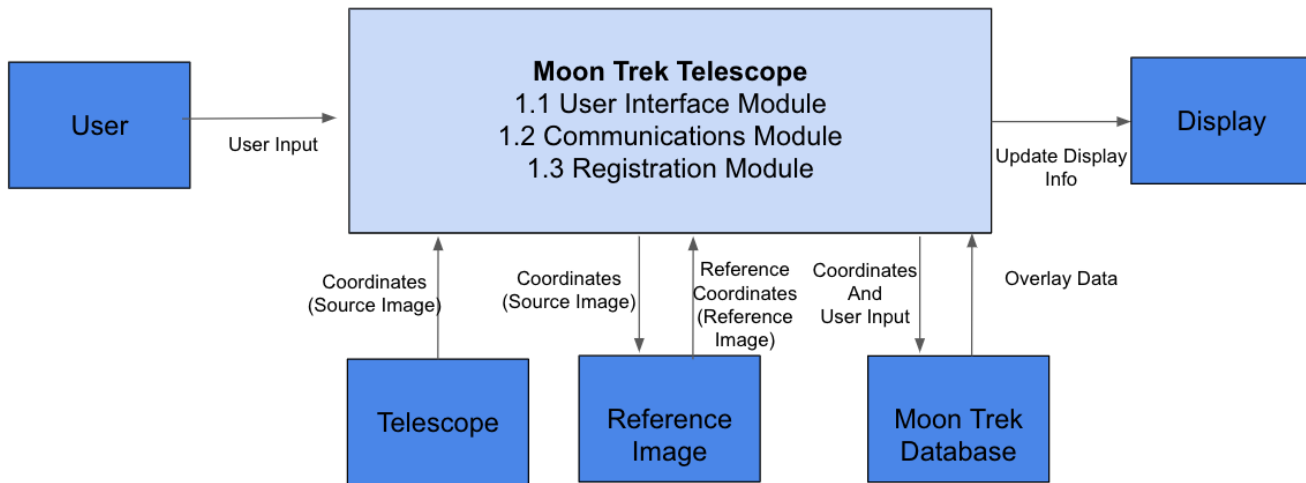
MTTA will fulfill all requirements with the assumption that the user has a valid internet connection. This software will not be functional without an internet connection. It is also assumed that the user will have a compatible telescope to integrate with MTTA. Without the presence of a telescope the application will be provided a set of hardcoded coordinates for a user to research information.

2.9 Apportioning of Requirements

Connectivity of a telescope to a mobile application implementation.

3. External Interface Requirements

DFD Level 0



3.1 User Interfaces

a. Initialization

- i. **Request:** User will request MoonTrek website and the initial landing page will be displayed
- ii. **Response:** MoonTrek website will display to the user

b. Web Home Page

- i. **Logo:** Clicking on it will take user back to the home page
- ii. **Navigation Bar:** Displays the different links for different types of pages
- iii. **Choose file / Upload:** Gives the user the functionality to upload an image

c. MoonTrek Services:

- i. **Moon Image:** Website will display image from coordinates provided in image/telescope and by connected to our model
- ii. **Moon Information Services:** Website will provide statistical data like location, vector, and nearest point information from image/telescope coordinates
- iii. **Moon Overlay Services:** Website will provide utilities to overlay informative images onto previous moon image

3.2 Hardware Interfaces

Telescope (TBD: future plans)

3.3 Software Interfaces

- Python, version 3.7.9. Source: <https://www.python.org/>
 - Language used to implement front-end and back-end interfaces and api.
- Django, version 3.1.7. Source: www.djangoproject.com
 - Web framework used to execute and run our web application.
- Opencv-contrib-python, version 3.4.2.17 or 4.5.0. Source: <https://opencv.org/>
 - Allows the implementation, execution, and testing of the circle detection program.
- Threejs, version r123. Source: <https://threejs.org/>.
 - Implement the back end image registration between user image and spherical moon model.
- Numpy , version 1.20.1 . Source : <https://numpy.org/>
 - Numerical computer tools for our calculations .
- Pillow, version 8.1.2 . Source : <https://pillow.readthedocs.io/en/stable/>
 - Library to allow us work with images .
- MoonTrek 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

- MoonTrek application portal
 - The application shall require internet connection to access web portal
- MoonTrek project image registration
 - The application shall receive user images via image submission form in 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 privacy of submitted data.

- The application shall use HTTPS to maintain 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

4.1-1 The application shall take in a moon visual captured by a telescope , camera , etc.

4.1-2 The application shall query and display different layers of data from the MoonTrek API and return specific details such as annotations of mare , craters , and landing sites.

4.1-3 The application shall limit user input to available data layers.

4.1-4 The application shall be able to overlay layers and change opacity for visualization convenience to a generic 3D model of the moon.

4.1-5 The application should allow users to obtain accurate longitude and latitude for any given pixel on the moon 's surface.

4.2 External Interface Requirements

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

4.3 Logical Database Requirements

The MoonTrek telescope application database contains mainly references to images uploaded by users . The many different moon layers, such as:”Landing sites”,”Craters”, and “Mare” and information are hard coded within the program . By specifying the correct coordinates of each item , the application shall place a mark on the correct location.

4.4 Design Constraints

- Standard limitation

Limited experience with technology such as : Django Web Framework, OpenCV, ThreeJS

- Hardware limitation

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

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Our software will handle only 1 image per upload. All calculations are done on the back-end; the user's wait times will be reflective of their internet connection speed when they upload their photos to our server. The average internet speed in the United States is 40 Megabits per second, which would be 5 megabytes per second of upload speed. An average user's image size will be between 3 and 15 megabytes, so the average wait time due to their connection speed will be 0.6 seconds and 3 seconds.

5.2 Safety Requirements

Users of this software must understand that their photos, along with any relevant data, will be stored in a database. As such, there is always an inherent risk that said photos and said data may be compromised. It is to always be assumed that a security system can be compromised, therefore, users must understand the inherent risks associated with sharing content on the internet.

The core of the software's utility is rooted in lunar photography. As such, the only way to obtain photographs of the moon is to do so from the outdoors. As of the time of the software's development, there are still COVID safety regulations in place. It is up to the user to take the proper precautions on their own behalf in order to adhere to local COVID safety regulations concerning masks and social distancing. One portion of our target audience, the amateur astronomer, may have become accustomed to partaking in "watch parties". For reference, a "watch party" is an event where a group of people will gather to view some specific occurrence, such as a meteor shower, full moon, or lunar eclipse. It is then up to the viewer, should they choose to attend a watch party, to take into consideration the risks associated with large group gatherings. No official safety certifications must be satisfied, but the user should follow their best judgement for their own safety.

5.3 Security Requirements

There are no security or privacy issues that our software requires to function appropriately. No login or user information is required or stored to access the software.

5.4 Software Quality Attributes

The following quality characteristics are what the software would require to be properly used by both users and developers.

Availability: The software shall be available on the website 24/7.

Flexibility: The software shall perform correctly for any uploaded user image of the Moon as well as any user.

Reliability: The software shall make various geometric calculations throughout the process in order to accurately display information of the Moon.

Responsiveness: The software shall load and display necessary features to the user in a matter of seconds.

Simplicity: The software features labels throughout and includes a tutorial that can be read at any time to simplify features.

Usability: The software shall be able to be used effectively, efficiently, and safely.

5.5 Business Rules

None at the moment.

6. Legal and Ethical Considerations

The user's privacy is of utmost concern, and we do our best to minimize the amount of personally identifiable information retained from the user's images. It is due to this ethical consideration that we are currently focused on retaining as little information about the user as possible; further, any data that is obtained by our software will not be sold, shared, or given to third parties. All user data will stay on our servers and will not be shared with any outside party or used for applications outside of the scope of this project.

The user must agree that the content they upload is their own and therefore not stolen or used without permission. We will comply with removing any material that is found to be copyrighted or stolen.

Illegal content will not be hosted on our servers, as such, any users that are found to be hosting illegal or malicious content will be banned based on their IP, and depending on the severity of the content, turned over to the proper authorities. Due to the crowdsourcing nature of the project, we can not allow malicious material to propagate on our servers.

Appendix B: Analysis Models

Architecture Concept

