**Moon Trek: Telescope  
Augmented Reality**

Sponsored by:

NASA Jet Propulsion Laboratory

Software Requirements Specification

(SRS)

Prepared by:

| Cindel Lopez-Sianez, | Fu-Cheng Chuang, |
| --- | --- |
| Eduardo Cruz, | Jingchao Feng, |
| Byron Garibay, | Daniel Gonzalez, |
| Tony Hong Jr., | Matthew Johnson, |
| Niloy Azad, | Jonathan Navarrete |

13 May 2022



# Table of Contents

**Section Page**

**Document Change Log ……………………………………………………………………………………3**

1. [**Introduction**](#_heading=h.l51ucx6pv0o9) **……………………………………………………………………………………………..…4**
   1. [Purpose](#_heading=h.417wx15mtmvl) …………………………………………………………………………..…………………4
   2. [Intended Audience and Reading Suggestions](#_heading=h.1ke9ymwnh2ou) …………………………………...…………………4
   3. [Product Scope](#_heading=h.wa5azcpqnve8) ………………………………………………………………….…………………..6
   4. [Definitions, Acronyms, and Abbreviations](#_heading=h.v9q734fb8xt0) …………………………………….………………….6
   5. [References](#_heading=h.5t0cja20daal) …………………………………………………………………………………………7
2. [**Overall Description**](#_heading=h.ta15pglyf9lj) **……………………………………………………………………..…………………8**
   1. [System Analysis](#_heading=h.m0nqwzabf7ts) ……………………………………………………………………………………9
   2. [Product Perspective](#_heading=h.bljazby40rne) ……………………………………………………………...…………………9
   3. [Product Functions](#_heading=h.amzbcq5wgtxj) ………………………………………………………………………………….9
   4. [User Classes and Characteristics](#_heading=h.6iwwqciu6wsk) ……………………………………………….………………10
   5. [Operating Environment](#_heading=h.vcaqlhn692sd) ………………………………………………………...…………………10
   6. [Design and Implementation Constraints](#_heading=h.d74q4lqvwdj) ……………………………………...…………..………11
   7. [User Documentation](#_heading=h.bgspaa2vv10x) ………………………………………………………………………………11
   8. [Assumptions and Dependencies](#_heading=h.iwzlu3ckzmye) ……………………………………………….………….………11
   9. [Apportioning of Requirements](#_heading=h.na1f9dhso77s)  ………………………………………………..………….………11
3. [**External Interface Requirements**](#_heading=h.yhd8qz526d0g)  **………………………………………………………………………12**
   1. [User Interfaces](#_heading=h.s5jah2ml9mjd) ………………………………………………………………...………….………12
   2. [Hardware Interfaces](#_heading=h.luqvec361gfp) ………………………………………………………………….…………13
   3. [Software Interfaces](#_heading=h.luqvec361gfp) …………………………………………………………...…………..………13
   4. [Communications Interfaces](#_heading=h.hhym91u54dk6) …………………………………………………..…………..………14
4. [**Requirements Specification**](#_heading=h.dcasyfbffjsw) **…………………………………………………………..…………….……15**
   1. [Functional Requirements](#_heading=h.qnxc82dx0fs3) ……………………………………………………....…………………15
   2. [External Interface Requirements](#_heading=h.flr56s90y2wy) ……………………………………………...…………..………15
   3. [Logical Database Requirements](#_heading=h.5yz8p6s9ab7o) ………………………………………………………….………15
   4. [Design Constraints](#_heading=h.68d1kegjtnr9) ………………………………………………………………………..………15
5. [**Other Nonfunctional Requirements**](#_heading=h.apivgmo5iqv5) **………………………………………………...…………..………16**
   1. [Performance Requirements](#_heading=h.52m9zrlptssm) ……………………………………………………………….………16
   2. [Safety Requirements](#_heading=h.uy7auvjksdzy) …………………………………………………………..………….………16
   3. [Security Requirements](#_heading=h.x35lkck6pjoi) ………………………………………………………....…………………16
   4. [Software Quality Attributes](#_heading=h.vgp2dpladmnu) …………………………………………………...…….……………17
   5. [Business Rules](#_heading=h.bunx8eabw1p2) ………………………………………………………………....…………………17
6. [**Legal and Ethical Considerations**](#_heading=h.bpiftnnl5e3s) **………………………………………………….……………………18**
7. [**Appendix A: Analysis Models**](#_heading=h.42geab9fwkq0) **…………………………………..……………………………….………18**

# Document Change Log

| Editor 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 | 1.2 |
| Version 2 | 11/18/21 | New Group | 2.0 |
| Tony | 12/7/2021 | Table of contents, fixing formats, tagged errors, page setup | 2.01 |
| Fall Semester End | 12/10/2021 | Fix grammatical errors, more changes to format, added additional information | 2.02 |
| Spring Semester End | 05/13/2022 | Update outdated information | 2.03 |

# Introduction

## Purpose

The main purpose of this document is to present a detailed description of the Moon Trek telescope program. It will cover all aspects of the software including the purpose and features, the interface of the application, what the system will do, future goals, and the constraints under which it will operate.

## 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 Moon Trek 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.

* + - **Introduction**
      * This section offers a summary of the Moon Trek 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 Moon Trek 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.

## Product Scope

The scope of this document includes the following:

* + - The Moon Trek 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.

## Definitions, Acronyms, and Abbreviations

| **SRS** | - Software Requirements Specification |
| --- | --- |
| **SDD** | - 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 |

## 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.

“**Django Documentation | Django Documentation | Django**.” Docs.djangoproject.com, docs.djangoproject.com/en/3.1/.

“**OpenCV: OpenCV-Python Tutorials**.” Docs.opencv.org, docs.opencv.org/master/d6/d00/tutorial\_py\_root.html.

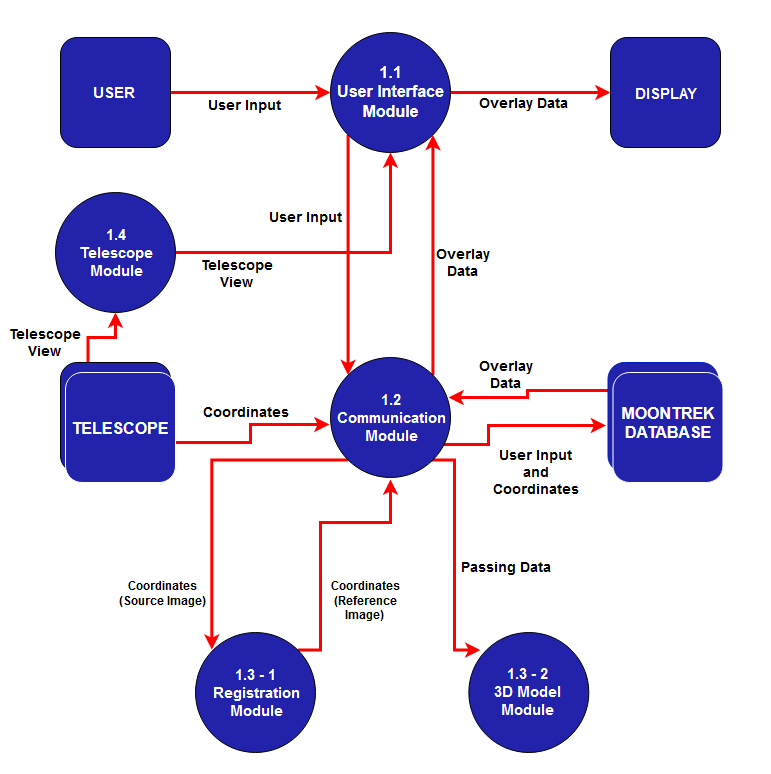
“**Three.js – JavaScript 3D Library**.” Threejs.org, 2019, threejs.org/.

**“Closest-Point API | Planet-Vector API”** JPL,

http://54.157.167.17:5000

# Overall Description

DFD Level 1



## System Analysis

* + - The Moon Trek web 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 were 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.

## 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 overlay 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 Moon Trek 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 application's interface. Then the application will use the timestamp associated with the image to replicate the position of the Moon, Earth, and Sun system with respect to the given time through the use of the JPL’s API calls that will handle the computation of various planetary geometries. An LRO WAC Mosaic rectangular map of the Moon from JPL’s portal will be applied to the surface of our 3D Moon model, while the textures of the Earth and Sun models will be open source textures.

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

## 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 3D model representation of the Moon, Earth, and Sun.

## 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.

## Operating Environment

Operating environment for the Moon Trek telescope Program will be a web-based application accessible from any laptop or smartphone.

* + - **Client/Server System:** Django
    - **Database:** SQLite

## Design and Implementation Constraints

Limited time - 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, our ability to obtain the necessary data for our project to display the Moon in a wider range of areas have been affected.

## 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.

## 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 given a set of hardcoded coordinates for a user to research information.

## Apportioning of Requirements

Connectivity of a telescope to a mobile application implementation.

# External Interface Requirements

DFD Level 0

## User Interfaces

### Initialization

* 1. **Request:** The user will request the Moon Trek website and the initial landing page will be displayed.
  2. **Response:** Moon Trek website will be displayed to the user.

### Web Home Page

* 1. **Logo:** Clicking on it will take the user back to the home page.
  2. **Navigation bar:** Displays the different links for different types of pages.
  3. **Choose file / Upload:** Gives the user the functionality to upload an image.

### Moon Trek Services:

* 1. **Moon image:** The website will display images from coordinates provided in the image/telescope and by connecting to our model.
  2. **Moon information services:** The website will provide statistical data like location, vector, and nearest point information from image/telescope coordinates.
  3. **Moon overlay services:** The website will prove utilities to overlay informative images onto previous Moon images.
  4. **Hardware Interfaces**

Telescope (TBD: future plans)

## Software Interfaces

* + - **Python, version 3.7.9. Source:** [https://www.python.or](http://www.python.org/)g/
      * Language used to implement front-end and back-end interfaces and API.
    - **Django, version 3.1.7. Source:** [www.djangoproject.com](http://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 3D environment representing the Moon, Earth, and Sun.
    - **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](https://pillow.readthedocs.io/en/stable/)/
      * 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.

## 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.

# Requirements Specification

## Functional Requirements

1. The application shall take in a Moon visual captured by a telescope, camera, etc.
2. The application shall query and display different layers of data from the Moon Trek API and return specific details such as annotations of the maria, craters, and landing sites.
3. The application shall apply real data to scale 3D models for the Moon, Earth, and Sun, along with their distance ratios.
4. The application shall replicate the position for the 3D models representing the Moon, Earth, and Sun, while also adding directional lighting to the Sun model so it can cast shadows.
5. The application shall be able to overlay layers and change opacity for visualization convenience to a 3D model of the Moon.
6. The application should allow users to obtain accurate longitude and latitude for any given pixel on the Moon's surface.

## External Interface Requirements

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

## Logical Database Requirements

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

## 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.

# Other Nonfunctional Requirements

## 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.

## 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 judgment for their own safety.

## 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.

## 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 to accurately display information about 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.

## Business Rules

None at the moment.

# 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 A: Analysis Models

Old Architecture Concept