**Moon Trek: Telescope  
Augmented Reality**

Sponsored by:

NASA Jet Propulsion Laboratory

Software Design Specification

(SDS)

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**](#_heading=h.47qtsiqfyo5n) **……..……………………………………………………………….........2**

1. [**Introduction**](#_heading=h.v0kcmv2jy4j3) **………….…………………………………………………………………………....3**
   1. [Purpose](#_heading=h.qm3lx35349f7) …………………………………………………………………………………….3
   2. [Document Conventions](#_heading=h.hzgvkvm2ftaa) ……………………………………………………………………3
   3. [Intended Audience and Reading Suggestions](#_heading=h.gwat2aczjr7l) ……………………………………………..4
   4. [System Overview](#_heading=h.p9xfg711hvlw) ………………………………………………………………………….5
2. [**Design Considerations**](#_heading=h.kx03se7p37u9) **…………………………...…………….…………………………….…...6**
   1. [Assumptions and Dependencies](#_heading=h.rr6n5mezkiy) ……….………………………………………………….6
   2. [General Constraints](#_heading=h.34sbqvptgda0) …………………..……………………………………………………6
   3. [Goals and Guidelines](#_heading=h.yj9tapmkhiaj) ………………….…………………………………………………..8
   4. [Development Methods](#_heading=h.m9p3r3ey3hhw) …………………………………………………………………….8
3. [**Architectural Strategies**](#_heading=h.a85ao7s9yz84) **…………………..………………………………………………………9**
   1. [Telescope Integration](#_heading=h.a8o4v2vdu4b) ………………………………………………..………………….…9
   2. [Image Registration](#_heading=h.4onsk2kwtym4) ………………………………………..……………………………….9
   3. [3D Modeling](#_heading=h.diicorgghcnw) ………………………………………………………………………………9
   4. [User Interface](#_heading=h.wumca690wafg) ……………………………………………...………………………………9
   5. [Team Leads](#_heading=h.pabofr3yiwqy) ...……………………………………………...………………………………9
4. [**System Architecture**](#_heading=h.b7dc7ryh3sd6) **……………………………………………………………………………..10**
5. [**Policies and Tactics**](#_heading=h.87uwyebi45nw) **………………………………………………………...……………………13**
   1. [Choice of Specific Products Used](#_heading=h.zaq2qa3n9klp) ……………………………….…………………….13
   2. [Ensuring Requirements Traceability](#_heading=h.dlp2rnc1vq51) ……………………………….………………….…14
   3. [Testing the Software](#_heading=h.3ujffutksk9v) ……………………………………………………………………..14
   4. [How to run the Application](#_heading=h.h384rja9ylm8) ……………………………………..……………………..14
6. [**Detailed System Design**](#_heading=h.v1xdma2cjvre) **…………………………………………………….……………………15**
   1. [User Interface Module (UIM)](#_heading=h.qgbvsop0p3ut) ………………………………………..………………..…15
   2. [Communication Module (CM)](#_heading=h.r00fvbwo9veq) ………………………………………..…………………16
   3. [Registration Module (RM)](#_heading=h.6o74vllkp0oq) ……………………………………………………………..17
   4. [Telescope Module (TM)](#_heading=h.dm2a62edv3zx) …………………………………………………………………17
7. [**Detailed Lower Level Component Design**](#_heading=h.bvbf93t1swb1) **……………………………………………………..19**
   1. [Overview of User Interface](#_heading=h.d27dfjxfhk2u) ………………………………………….………………..…19
8. [**Database Design**](#_heading=h.4m8n1ojlst62) **…………………………………………………………………..……………..21**
9. [**Requirements Validation and Verification**](#_heading=h.lm1jhvyuv503) **…………………………………………………..22**
10. [**Glossary**](#_heading=h.w4n9d7yljj5i) **…………………………………………………………………...……………………..23**
11. [**References**](#_heading=h.o63y9xx8xlmw) **………………………………………………………………..………………………24**

# Document Change Log

| Name | Date | Reason For Changes | Version |
| --- | --- | --- | --- |
| Draft 1 | 12/09/20 | Start of documentation | 1.0 |
| Final Draft | 05/14/21 | ? | 1.1 |
| Version 2 | 12/10/21 | Grammatical errors, change in formatting, and updates made to fit new objectives | 2.0 |
| Spring Semester End | 05/13/22 | Update outdated information | 2.1 |

# Introduction

## Purpose

The primary purpose of this document is to present a detailed description of the Moon Trek: Telescope Augmented Reality, hosted by Jet Propulsion Laboratory. It will cover all aspects of the software, including the purpose and features, the application’s interface, what the system will do, and the constraints it will operate on.

## Document Conventions

Describe the standards or typographical conventions followed when writing this documentation, such as fonts, highlighting that have special significance, or listing conventions. For example, it states whether priorities for higher-level requirements are assumed to be inherited by detailed requirements or whether every requirement statement is to have its preference.

* + - **Main Paragraphs**
      * As seen above bullet points and directly below numbered subheadings, indicate a description of topics.
    - **Specific Components**
      * As seen next to black bullet points, list specific parts of bolded topics.
    - **Specific Details**
      * As seen next to hollow bullet points, describe specific components and their function/ role within the bolded topic.
    - **Outline Form**
      * Some main paragraphs may be in the shape of an outline.
      * The form of this outline will usually be a numerical number, followed by lowercase letters and then lowercase roman numerals.
    - **Acronyms**
      * If used, it will be written out entirely once and then followed by the first instance of the word's acronym.
    - **Typefaces**
      * The paper will primarily use one font.
      * *Italics,* denote titles and names of works or objects to allow titles or names to stand out and not be used overtly.
      * **Bold,** font is used to denote a heavier weight of the significance;   
        Used in titles and to show importance.
    - **Margins**
      * The document shall follow Modern Language Association (MLA) format and incorporate 1-inch margins on each section.

## 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 Moon Trek: Telescope Augmented Reality development. This includes software developers, project advisors, liaisons, team managers, and whoever works on this project next. This document does not need to be read sequentially; users are encouraged to jump to any section they find relevant.

* + - **Introduction**
      * This section summarizes the project, including purpose, document conventions, intended audience and reading suggestions, and system overview.
    - **Design Constraints**
      * This section describes many of the issues that need to be addressed or resolved before devising a complete design solution.
    - **Architectural Strategies**
      * This section describes any design decisions and/or strategies that affect the overall organization of the system. It will clarify the fundamental abstractions and mechanisms used in the system architecture.
    - **System Architectures**
      * This section offers a high-level overview of how the functionality and responsibilities of the system were partitioned and then assigned to subsystems or components.
    - **Policies and Tactic**
      * This section describes design policies and/or tactics that do not have sweeping architectural implications. These will affect the details of the interface and/or implementation of various aspects of the system.
    - **Detailed System Design**
      * Each subsection of this section will refer to or contain a detailed description of a system software component.
    - **Detailed Lower level Component Design**
      * This section will describe other lower-level classes, components, subcomponents, and assorted support files.
    - **Database Design**
      * This section will include details about any databases used by the software.
    - **Requirements Validation and Verification**
      * This section offers a table that lists each of the requirements specified in the Software Requirements Document (SRS) for this software.
    - **Glossary**
      * This section provides definitions for relevant terms, acronyms, and abbreviations necessary to comprehend this document.
    - **References**
      * This section lists any other documents or web addresses to which this Software Design Specification (SDS) refers.

## System Overview

The System Overview gives a brief description of how our application functions and will

later discuss how the program is to be planned. This is meant to be read from a non-technical/high-level user’s point of view.

* + - Build upon the previous Moon Trek Django-web application. The user shall capture an image of the Moon from their telescope or upload a corresponding image into the web application.
    - The user shall give the image to the web application; then, in turn, the web application provides points of interest on the Moon such as craters, maria, and landing sites.
    - The augmented reality portion of the project improves the data overlays, creating a 3D model of the Moon created by Jet Propulsion Lab's high-quality images and user-uploaded images of the Moon.
    - The team's objective is also to complete the telescope for computer communication, improve the accuracy of the image registration, and create a 3D model of the Sun, Moon, and Earth based on the images uploaded.
    - The same 3D model of Earth will also annotate where the picture was taken and its time.

# Design Considerations

This section describes many of the issues that need to be addressed or resolved before devising a complete design solution.

## Assumptions and Dependencies

Describe any assumptions or dependencies regarding the software and its use.   
These may concern such issues as:

* + - **Celestron Telescope (Unknown NexStar Evolution Model)**
      * Unknown models make compatibility difficult.
      * Telescope access through the spring 2022 semester.
    - **User Telescope (Unknown entity)**
      * We will know much more about the Celestron telescope than the user’s telescope, whose compatibility will be more difficult to gauge.
    - **Cross-Browser Functionality**
      * No cross-browser compatibility tests have been done so far, so applications running on different browsers are assumed.
    - **NASA’s Moon Trek API**
      * Has existing calls that meet functional requirements.
      * Individual components can call JPL’s APIs.

## General Constraints

Describe any global limitations or constraints that significantly impact the design of the system's software (and describe the associated impact). Such restrictions may be imposed by any of the following (the list is not exhaustive):

* + - **Hardware or Software Environment**
      * Lack of Django experience: May limit the application’s functionality to be adequately executed.
      * Lack of 3D modeling experience: May skew image registration accuracy on the sphere object.
      * Lack of image registration experience: Image registration may not function properly on all images.
      * Lack of telescope knowledge: May have trouble getting applications to function alongside certain types of telescope.
      * Lack of software intersectionality awareness: We don’t know how well the components will come together in Django.
      * Lack of server experience: It is unknown how well the application will behave once implemented in a host’s server.
      * Lack of documentation: Some incomplete modules were left unfinished and may be undermined.
    - **End-User Environment**
      * Cannot determine how the styling of the page looks on different computer screens. Depending on a user’s display specifications, some components may not look as great.
    - **Availability or Volatility of Resources**
      * Cannot determine how the end user’s computer will handle the costs of running the application on their computer, may have to make the application more cost-effective once delivery date comes closer.
    - **Interoperability Requirements**
      * Ensure that image data is accepted in the application to add data layers.
      * Ensure that connection to the telescope can be verified to submit images to the application.
    - **Interface/Protocol Requirements**
      * Must make the interface as user-friendly as possible since there is no guarantee that users have already used the NASA Moon Trek application.
      * Must create alerts if specific inputs from a user are invalid or inoperable so that the application doesn’t crash.
    - **Data Repository and Distribution Requirements**
      * The software aims to store locally as a means of privacy and prevents requiring various permissions that would otherwise be necessary.
    - **Security Requirements (or other Regulations)**
      * As noted in the Association for Computing Machinery (ACM) code of ethics, staff or people maintaining software must uphold integrity and discretion.
    - **Memory and other Capacity Limitations**
      * May need to shrink the media file to increase the application speed, as multimedia files often use a lot of memory.
    - **Performance Requirements**
      * Must be able to execute API calls.
      * Must be able to run a python script.
    - **Network Communications**
      * Access to the application is as dependable as the internet connection. May need to make certain functions available offline.
    - **Verification and Validation Requirements**
      * Must ensure that inputs other than telescope images will not be accepted and will not crash the application.
      * Must ensure that the user knows why specific inputs are not accepted.

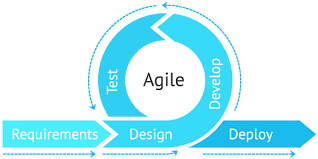
## Goals and Guidelines

* + - Designing a user-friendly, efficient, and intuitive UI encourage amateur astronomers to cultivate their curiosity about the Moon.
    - Developing a UI that bridges the gap and serves as the connection between the backend development, telescope, and user.
    - The application is an extension and user-friendly version of the existing NASA Moon Trek portal.
    - The application can perform image registration to gather appropriate images for data layering and API annotations.
    - Use a 3D modeling JavaScript library to replicate the positioning of the Moon, Earth, and Sun systems.
    - The application has a mandatory delivery date that must be met. (by May 2022)

## Development Methods

The development methods that we plan to use for the team are to follow Agile development similar to the previous group’s. The feedback from both the advisor and liaisons will supplement the work in the two weeks. It will enable a client-customer collaboration on the strengths and weaknesses of the development by following the cycle of identifying requirements, designing, developing, testing, deploying, and receiving feedback.  
  
The following constraints are listed:

* + - 2-week sprints of fulfilling the objectives set out.
      * 6-8 objectives outlined
    - Weekly meetings/stand-ups
    - Bi-weekly meetings with liaisons



# Architectural Strategies

The following are the respective teams’ strategies to carry out their objectives.

## Telescope Integration

* Telescope integration will consist of a telescope, the Moon Trek application, and possibly a mobile device that displays and controls the telescope view.
* The telescope view will be sent to the user interface, where a photo can be captured.
* The captured photo will be able to be uploaded and registered on Moon Trek.

## Image Registration

* + - Image Registration is used to identify points of interest on the Moon.
    - Points of interest will be found using circle detection and SIFT, an algorithm used to detect and match locations on the Moon.
    - After the image registration process is complete, the points of interest, such as craters, landing sights, and maria, will be plotted.

## **3D Modeling**

* + - Create 3D models representing the Moon, Earth, and Sun while replicating their positions based on a timestamp from a screenshot of the Moon.
    - This will be done using Three.js, a JavaScript library, which will allow us to perform texture mapping with a Lunar Reconnaissance Orbiter Wide Angle Camera map provided by JPL.
    - In addition, this library will implement directional lighting onto a 3D model to cast shadows.

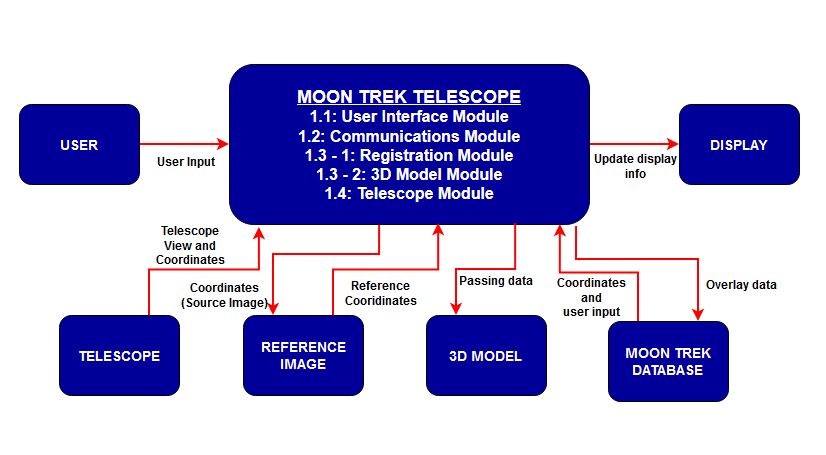
## User Interface

* + - Utilize buttons as guides to the user.
    - Allowing users to decide what they want to do on our website.
    - Mobile Integration for mobile users.
    - Make fluid and seamless.

## Team Leads

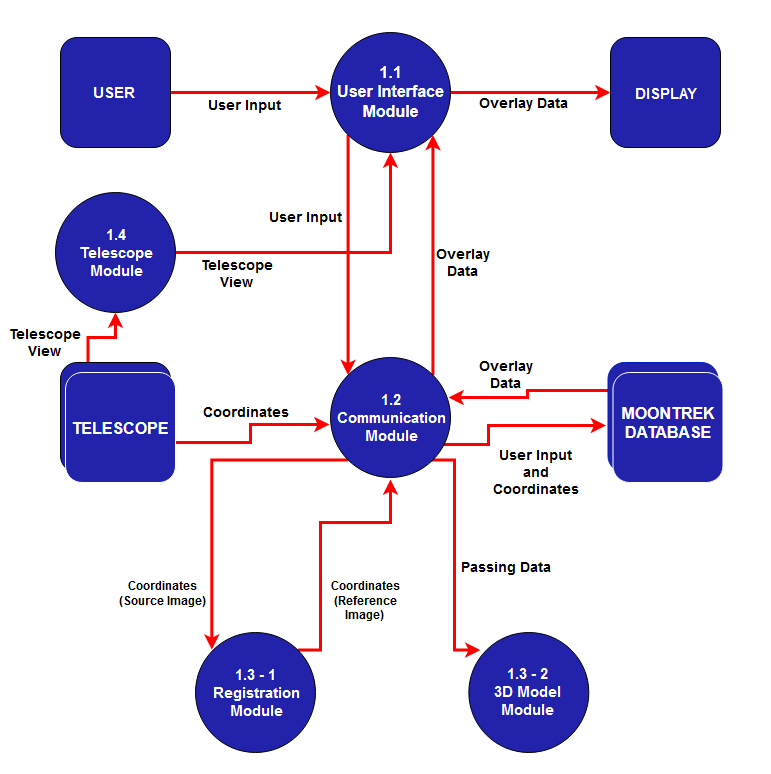
* + - Collaborate with all four teams and keep the road map as the objective for each team.
    - Help navigate teams with the Django web framework, and upload to Git.
    - Fill team roles, provide additional aid to each respective team.
    - Serve as leadership and guidance for the project team.

# System Architecture



DFD Level 0

The modules of the Moon Trek Telescope are the User Interface Module, the Communications Module, Registration Module, 3D Model Module, and Telescope Module. The user interface module is in charge of displaying the information gathered from JPL’s database. This information can be annotations or overlays relative to the graphical coordinates of the moon present in the user's view. The registration module will be in charge of adequately locating each coordinate in the source view of the moon to the reference image of the moon. The 3D Model Module provides the user with a 3D model representation of the Moon, Earth, and Sun system with the option to toggle data overlays. The communications module would allow us to reach the JPL’s datasets of the moon and fetch them to use in our user interface module. The Telescope Module will connect the telescope to MoonTrek and display an uploaded photo from the live feed to the UI.



DFD Level 1

The system comprises four modules; user interface, communications, registration, and telescope module. The main modules doing the work are the communications and registration modules. The telescope image will be provided to these modules and execute crucial tasks.

Images from the telescope will be passed to the registration module, where all image registration procedures occur. The correct coordinates will be generated and given back to the communication module. The communication module will fetch overlay data with the correct produced coordinates implemented from the registration module. Then the data will be passed to the user interface and 3D model module to display it to the user’s view of the moon. The telescope module will connect the application and send the uploaded photo from the live feed to the user interface.

# Policies and Tactics

## Choice of Specific Products Used

For development, the following products were used:

| **Products Used** | **Purpose/Function/Details** |
| --- | --- |
| Django | Python-based, accessible, and open-source web application framework. |
| WebGL | Javascript API for rendering 2D & 3D graphics in a web browser. |
| Three.js | Javascript library and application programming interface to utilize 3D graphics in a web browser using WebGL. |
| OpenCV | Library of multiple languages used for image recognition and real-time computer vision. |
| SQLite | Database management System in C. |
| CelestronSkyPortalApp | The application pairs with a compatible Celestron telescope to control a computerized telescope and view overlays. |
| ASCOM  (Under consideration) | Cross-platform protocol for communicating with astronomy equipment. |
| Alpaca  (Under consideration) | Network-based protocol built on ASCOM’s libraries. |
| Notion | To coordinate with the team and plot out a whiteboard and tasklist. |

## Ensuring Requirements Traceability

* Requirements listed in the Software Requirements Specification (SRS), Software Design Specification (SDS), and spring’s Powerpoint presentation will serve as roadmaps to follow through.
* The secretary/team leads to plan the meetings, set the agenda, create a meeting minutes template to follow for the adjacent meeting.
* The team follows a pseudo-form of *Robert’s Rule of Order*, which mandates guidelines for running a meeting.
* The two team leads assigned respective teams and helped guide them to their objectives.
* The team communicated through *Discord, Slack*, *Zoom*, meeting minutes, or email.
* We were guided by an advisor and liaisons who explained our requirements depending on our progress.

## Testing the Software

Software testing occurred during the software development, as the team needed to constantly ensure that APIs used for calculations within the code were behaving as expected. Also, when compiling/running the code in a testing environment, ensure desired images and results are loaded on the screen.

The software will later be deployed on JPL servers and tested to ensure all functionality stays the same as the testing environment.

## How to run the Application

1. Download the web application here: <https://github.com/Tonize/MoonTrek-Telescope-AR>
2. The project is installed in IP address: 54.157.167.17 and port: 8000
3. To run the application from a local browser, the applied address would have to be http://54.157.167.17:8000/, and the application should appear after JPL has activated the server.

# Detailed System Design

# The Detailed System Design showcases the responsibilities of what needs to be changed within our system. In time, this will change throughout the semester.

## User Interface Module (UIM)

### Responsibilities

The UIM works to display an image of the moon, and the statistical data is based on geographical portions of the moon from the user’s given position on Earth. The UIM is configured to connect through a smartphone and a telescope or smartphone and user input. The UIM overlays informative annotations over the portion of the Moon being displayed based on the user’s geographical location based on longitude and latitude.

### Constraints

The UIM is expected to work as long as the user has a good and steady internet connection and would otherwise be non-functional without an internet connection. Without a connectable telescope, this application will be given a set of hard-coded data that may not accurately represent the moon from the user’s location on Earth.

### Composition

The UIM acts specifically to relay information to the user and connects with the Registration Module (RM) and the Communication Module (CM) to display the data to the user. The UIM receives the overlay information that the CM obtains from the RM. The UIM then displays the overlay data to the user via smartphone or monitor display.

### Uses/Interactions

The user interactions with the UIM include a drop-list that can display various statistical data from JPL’s database. Assuming the user is using a monitor with a mouse and keyboard, the user will use the mouse to select the type of information of the moon that they choose from the drop-list provided.

### Resources

The UIM uses JPL’s database to obtain data about the moon that relays through the CM and RM to display recorded data of the moon to the user.

### Interface/Exports

The UIM displays information to the user with a drop list on the left side of the display window. The rest of the window shows the moon and the related data based on user input with annotations related to the user’s selection from the drop-list.

## Communication Module (CM)

### Responsibilities

The CM works as a link between the UIM and the RM, which allows the software to fetch necessary data from the RM and the JPL databases. The CM will then relay that information to the UIM. The primary use of the CM is the connection for data to be displayed to the user using the user’s telescope geographical coordinates and the Moon Trek Database.

### Constraints

The CM could take longer to relay information without any connectivity disruptions based on the internet connectivity and JPL’s databases’ speed. The hardware and software of connectable telescopes can partially delay receiving information.

### Composition

The CM receives the user’s coordinates on Earth and then connects with the RM to obtain a reference image. The CM then takes the user’s input, coordinates, and the reference image from the RM and runs these components through the Moon Trek database to send information from the database to the UIM to display.

### Uses/Interactions

The user does not directly interact with the CM. The CM interacts with the user’s telescope and uses the coordinates to interpret necessary data and relay the information. The CM receives a hard-coded user’s location when accessing the Moon Trek telescope without a connectable telescope.

### Resources

The CM connects with the user’s telescope to obtain location coordinates and links to the Moon Trek Database to interpret necessary data to send to the UIM for display. The CM interacts with the RM to collect a reference image based on the user’s coordinates on Earth.

### Interface/Exports

The CM exports data from the Moon Trek database to the UIM for display. The CM exports geographical coordinates from the user’s telescope or hardcoded values to the RM to interpret a reference image.

## Registration Module (RM)

### Responsibilities

The RM interprets a reference image based on coordinates received from the user’s telescope that the CM exports. The RM correctly locates each coordinate of a source image and creates a reference image for display.

### Constraints

The RM is reliant on the user’s coordinates to relay information. The user’s geographical coordinates are necessary for the RM to function.

### Composition

The RM relies on the source image and is in charge of adequately locating each coordinate in the source view of the Moon to the reference image of the Moon.

### Uses/Interactions

The user does not directly interact with the RM. The RM interacts with the user’s input of coordinates to produce a reference image for display.

### Resources

The RM needs a source image and the geographical coordinates of the user to display the necessary data correctly.

### Interface/Exports

The RM exports a reference image determined by a source image based on the user’s coordinates. The reference image is exported to the CM to display by the UIM.

## 3D-Model Module (3DM)

* + 1. **Responsibilities**

The 3D-Model Module (3DM) provides a way for users to learn about the Moon through a 3D model representation of the Moon, Earth, and Sun system. The primary feature of the 3DM is to show users the closest point between the Moon and Earth, along with an annotation of where the Moon photo was taken. Additionally, the 3DM utilizes an API in order to position the Moon, Sun and Earth, while also sizing them based on a scale ratio. The 3DM can toggle between different data layer textures for the display model of the Moon.

* + 1. **Constraints**

The core of 3DM is the models, and these are created by calling an image file for each individual model. Most of the web servers that were used had set a Cross-Origin Resource Sharing policy which created a constraint for the 3DM to obtain and make use of these images.

* + 1. **Users/Interactions**

The current version of 3DM requires the user to manually provide a timestamp and geolocation (latitude, longitude) to position each 3D model on the correct vector coordinate, show the user the closest point between Earth and Moon, and display an annotation of where the photo was taken. The user can also click on a specific data layer to display the Moon under different data layers.

* + 1. **Resources**

The 3DM requires an image file for each individual model as mentioned in section 6.4.2. 3DM is also required to fetch each model’s coordinate from JPL’s planet-vector API, as well as Earth and Moon’s latitude and longitude from JPL’s closest-point API for the camera to position properly on both Earth and Moon.

* + 1. **Interface/Exports**

The 3DM only provides read only feature for educational purposes, it does not provide any import/export functionality.

## Telescope Module (TM)

### Responsibilities

The TM connects the user’s telescope to Moon Trek. The telescope shows a live feed of its point of view through other software/hardware. A photo is captured and sent through the TM to the user interface. The user interface displays the photo, the user can capture and upload a photo using the buttons on the user interface.

### Constraints

The TM relies on the compatibility of the telescope being used and its connection to another device connected to Moon Trek. The TM also relies on external hardware and software to view the telescope’s point of view.

### Composition

The TM connects with the telescope, sends a photo captured from the live feed of the connected telescope to the Moon Trek UI using the device connected to the telescope.

### Uses/Interactions

The user will connect the telescope to Moon Trek. The telescope will connect to another user’s device connected to Moon Trek. The live feed obtained from the device and telescope will be used to capture a photo and route it to the user interface.

### Resources

The TM will need a compatible telescope and a device connected to the telescope that displays the telescope view. The telescope and device's connection to Moon Trek allows the live feed to be viewed.

### Interface/Exports

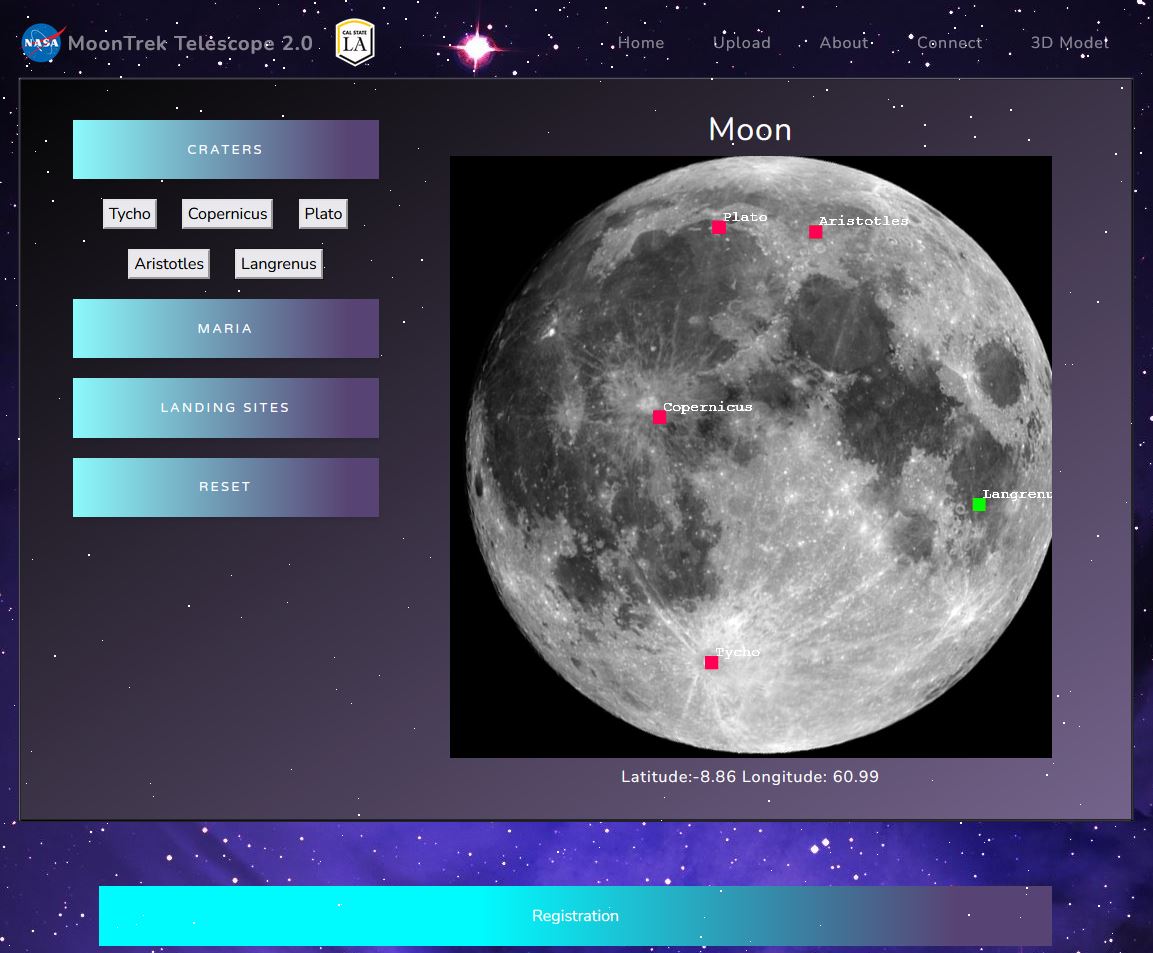
The TM exports a live feed of the telescope view to the UIM. The live feed is used to capture a photo on the UIM.

# Detailed Lower Level Component Design This describes the functionality of the system from the user’s perspective. It explains how the user will use your system to complete all the expected features in their respective team/category.

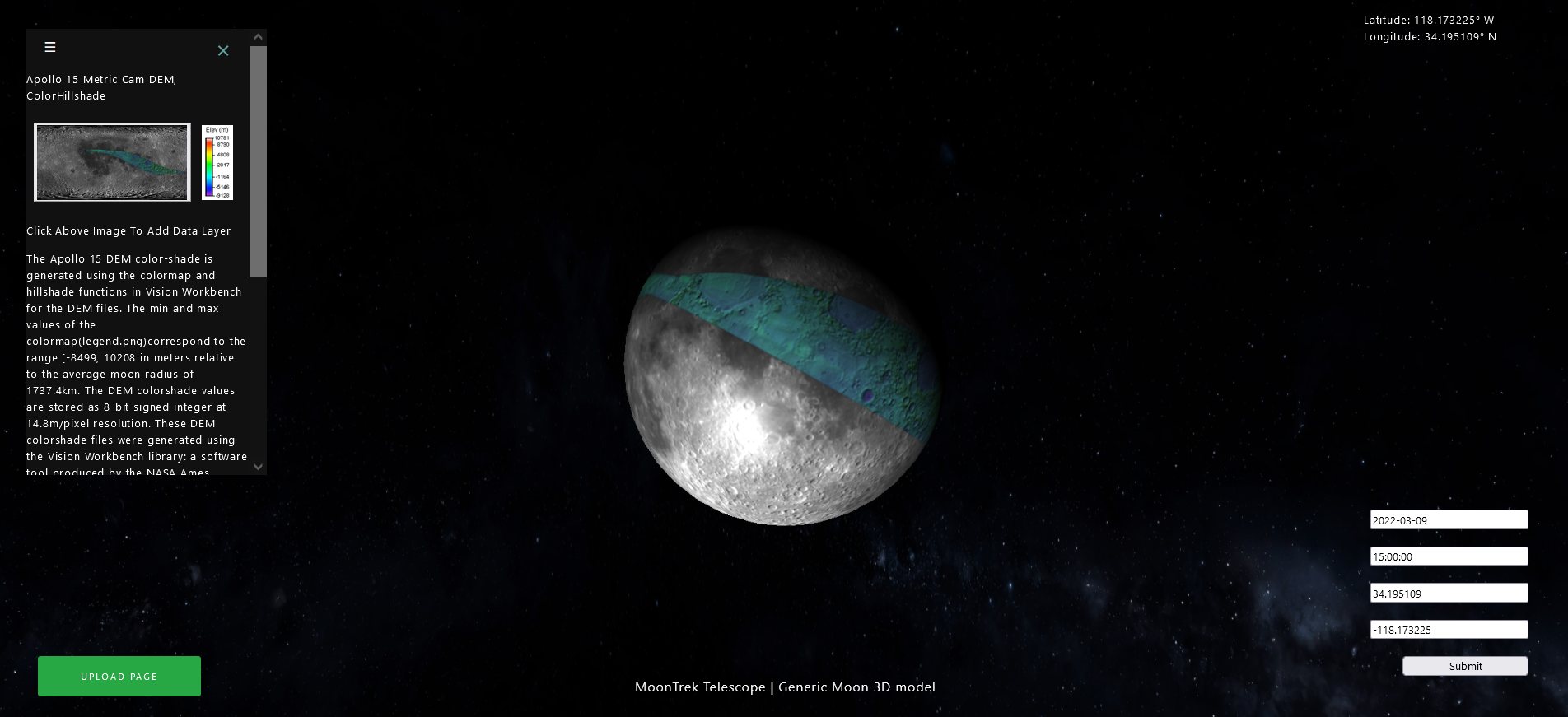
### Overview of User Interface

* **Functionality of the System from the User’s Perspective**
  + Users will interact with two buttons on the landing page: “Choose File” and “Upload.”
  + “Choose File” will allow a user to select an image to upload.
  + “Upload” will allow the user to upload their selected image.
* **Feedback Information that will be displayed for the Use**
  + Rendered and tagged model corresponding to their photo.
    - Any points of interest that our imaging algorithm finds will be displayed to the user.
* **The Usage**
  + The UI will display information and models back to the user.
  + It is interactable through buttons the Moon accurately represents.
  + It follows a linear approach that guides the user through the upload process.
* **Screen Frameworks/Images**
  + Figure 1: Landing page for the Moontrek website.



* + Figure 2: The final page contains annotations like craters, layers, mares, etc.

* + Figure 3: A generic 3D model representation of the Moon for users to see without the use of proper images.



# Database Design

The Moon Trek telescope application shall use the Nasa Moon Trek portal database and API to retrieve moon data.

To process the user moon images, we use a basic SQLite database to store references to images.

# Requirements Valid and Verification

| Requirements | Modules/UI/Components and Testing Methods |
| --- | --- |
| The application shall connect to the user’s telescope | * Telescope Module and User Interface * Running application |
| The application shall display the telescope view from the telescope or device controlling the telescope | * Telescope Module and User Interface * Running application |
| The application shall take in a picture upload from the user | * Communication module, Telescope Module, and User Interface * Running application |
| The application shall query and display different data layers from the MoonTrek API and return specific details. | * User Interface * Running the application trying out different layers of data |
| The application shall limit user input to available data layers. | * User Interface * Run the application to see what layers the user can see |
| The application shall be able to overlay layers and change opacity for visualization convenience. | * User Interface * running application trying out different layers and opacity |
| The application should allow users to search for a specific data set | * User Interface * Search for some data set |
| The application should be able to allow users to change the scale of the returning data layer | * User Interface * Try to change the scale of the data layer |

# Glossary

| **Acronym** | **Long Version** |
| --- | --- |
| SRS | Software Requirement Specifications |
| SDD | Software Design Document |
| JPL | Jet Propulsion Laboratory |
| UI | User Interface |
| WAC | Wide Angle Camera |
| LROC | Lunar Reconnaissance Orbiter Camera |
| MT | Moon Trek |
| NMT | Nasa’s Moon Trek |
| ACM | Association of Computing Machinery |

# 

# References

| Reference Name | Source |
| --- | --- |
| Software Design Document  Software Requirement Document | CSULA |
| ASCOM Standards | <https://ascom-standards.org/> |
| Moon Trek API | h[ttps://trek.nasa.gov/tiles/apidoc/trekAPI.html?body=moon](https://trek.nasa.gov/tiles/apidoc/trekAPI.html?body=moon) |
| Django Web Framework | h[ttps://docs.djangoproject.com/en/3.1/](https://docs.djangoproject.com/en/3.1/) |
| OpenCV | h[ttps://docs.opencv.org/master/d6/d00/tutorial\_py\_root.html](https://docs.opencv.org/master/d6/d00/tutorial_py_root.html) |
| ThreeJS | h[ttps://threejs.org/docs/](https://threejs.org/docs/) |