

# JPL - Moon Augmented Reality

## Senior Design Document

Group 2

December 2025

# Contents

|  |          |
|--|----------|
| <b>Version History</b>                             | <b>2</b> |
| <b>1 Introduction</b>                              | <b>2</b> |
| 1.1 Purpose . . . . .                              | 2        |
| 1.2 Intended Audience . . . . .                    | 2        |
| 1.3 Overview of the System . . . . .               | 2        |
| <b>2 System Architecture</b>                       | <b>3</b> |
| 2.1 Workflow of the system . . . . .               | 3        |
| 2.2 Breakdown (Client-side, Server-side) . . . . . | 3        |
| 2.3 Interface Module (UIM) . . . . .               | 4        |
| 2.3.1 Responsibilities . . . . .                   | 4        |
| 2.3.2 Composition . . . . .                        | 5        |
| 2.4 Registration Module (RM) . . . . .             | 5        |
| 2.4.1 Responsibilities . . . . .                   | 5        |
| 2.4.2 Composition . . . . .                        | 5        |
| 2.5 3D-Model Module (3DM) . . . . .                | 5        |
| 2.5.1 Responsibilities . . . . .                   | 5        |
| 2.5.2 Uses/Interactions . . . . .                  | 5        |
| <b>3 User Interface</b>                            | <b>5</b> |
| 3.1 How the system works . . . . .                 | 5        |
| 3.2 Database Design . . . . .                      | 6        |
| <b>Glossary</b>                                    | <b>6</b> |
| <b>References</b>                                  | <b>7</b> |

# Version History

| Date       | Description            | Version |
|------------|------------------------|---------|
| 12/04/2025 | Updated for snapshot 1 | 1.0     |
| 12/06/2025 | Updated for snapshot 2 | 2.0     |
| 12/08/2025 | Updated for snapshot 3 | 3.0     |
| 12/10/2025 | Updated for snapshot 4 | 4.0     |

## 1 Introduction

### 1.1 Purpose

The purpose of this document is to give informative documentation of the web application MoonTrek: Telescope Augmented Reality, hosted by Jet Propulsion Laboratory. This document will go over how to use the web application MoonTrek as well as the constraints, abilities, and features the application will operate on.

### 1.2 Intended Audience

This software requirement specification document are written for the general audience however this document can also be directed to audiences that 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.

### 1.3 Overview of the System

The System Overview provides a high-level user understanding of the non-technical functions of MoonTrek. Later in this document, there is more in-depth on the technical aspects of the application MoonTrek.

- It is built with Express and Vue.js. On the MoonTrek 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.

## 2 System Architecture

### 2.1 Workflow of the system

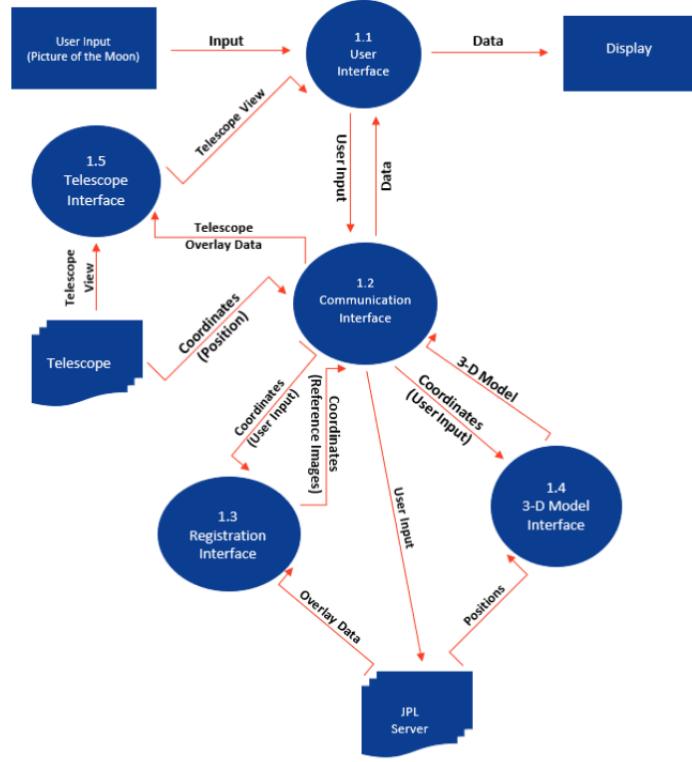
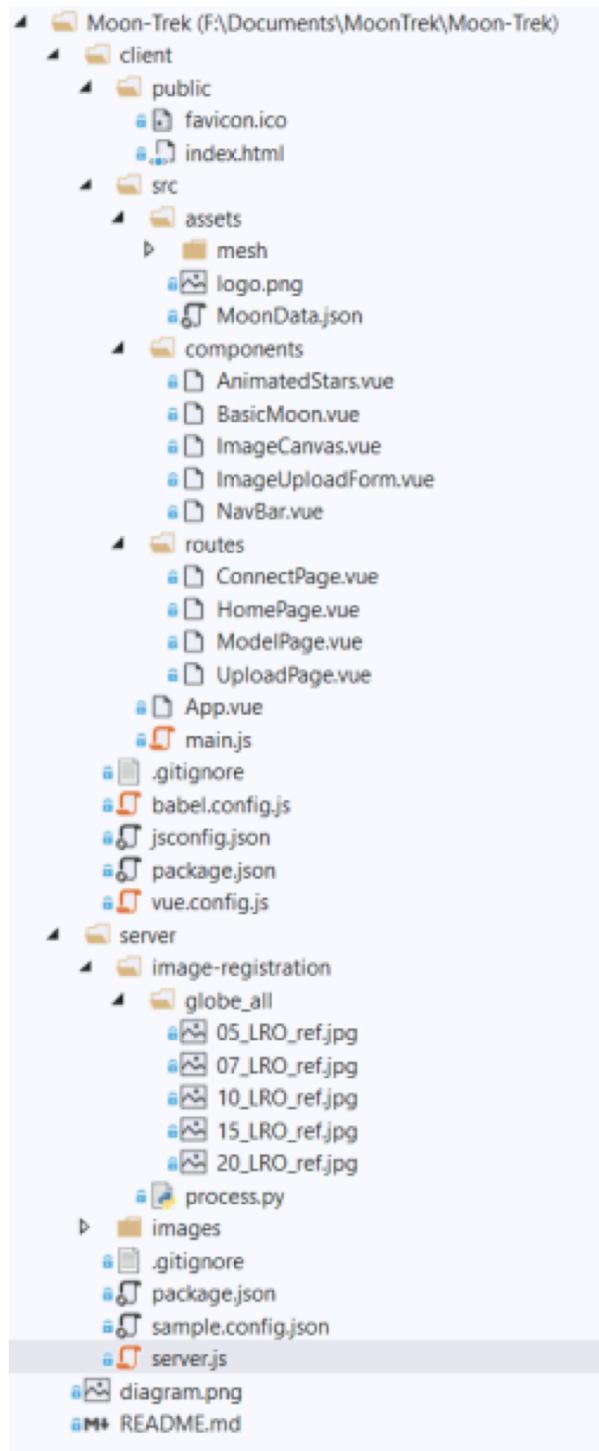


Figure 2: MoonTrek Data Flow Diagram 1 (DFD 1)

### 2.2 Breakdown (Client-side, Server-side)

The directory which contains all the source files for our web app is titled MoonTrek, and it contains two subdirectories: “client” and “server”. The client directory contains all the source files which make the web app functional for the user. This includes basic files for UI, such as the favicon, animated stars for the background, and the logo. The client directory also includes route files that manage the form upload process and the general structure of the backend. The server directory contains all the information necessary for image registration to take place. As of right now, this includes the reference images of the moon, which will be obsolete once our new method of image registration is implemented. Most importantly, it includes the process.py file which executes the SIFT algorithm which conducts image registration. The following image shows all the subdirectories and files in the MoonTrek directory:



## 2.3 Interface Module (UIM)

### 2.3.1 Responsibilities

The UIM displays VueJs routes to allow the user to interact with the MoonTrek project. The UIM is configured to be viewed on a smartphone, tablet, or computer. The UIM accepts the user's input images and shows a form if the image metadata does not include the required time and date information.

### 2.3.2 Composition

The UIM involves multiple .vue files, including ConnectPage.vue, HomePage.vue, ModelPage.vue, and UploadPage.vue, which create the HTTP responses for each page that the user will access. These pages then reference component files such as AnimatedStars.vue, ImageCanvas.vue, ImageUploadForm.vue, and NavBar.vue.

## 2.4 Registration Module (RM)

### 2.4.1 Responsibilities

The RM uses the SIFT machine learning algorithm to perform image registration between the user's image and a reference image created by the 3DM. The output is a transformation matrix that can be applied to the MoonTrek points of interest which will place them in the correct locations on the moon in the input image.

### 2.4.2 Composition

The bulk of the RM processing takes place in process.py on the server side. Reference images used for registration are stored in the adjacent images folder.

## 2.5 3D-Model Module (3DM)

### 2.5.1 Responsibilities

The 3DM is responsible for creating the context-aware reference image that the RM will use for image registration. The 3DM creates a dynamic photorealistic model of the earth, moon, and sun, and captures an image of the moon such that it appears very similar to the location and orientation of the input image. This will allow the image registration algorithm to be more accurate and dynamic since the images will be closer together to start with.

### 2.5.2 Uses/Interactions

The current version of 3DM requires a longitude, latitude, time, and date as input in order to orient the planetary bodies and the camera in exactly the way that they were when the picture was taken. This data is stripped from the input image's metadata, and if the metadata is missing or invalid, the user is prompted with a form through which they can input the missing information directly.

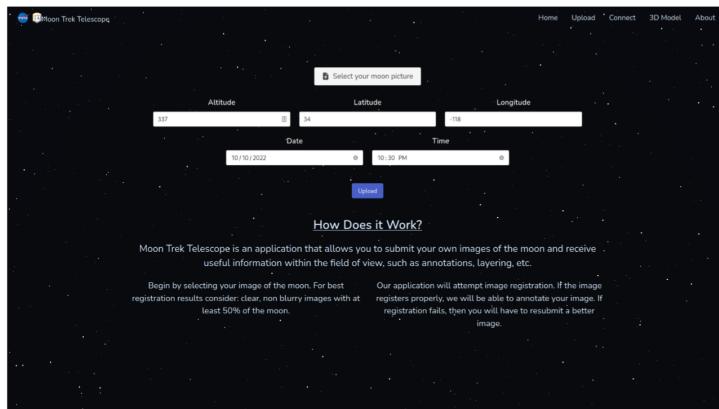
## 3 User Interface

### 3.1 How the system works

The user will open the application and see the homepage. From the homepage, the user will be able to upload their own image of the moon. The user is able to see their moon next to a picture of the Moon with all the data. We will also implement a 3d model that shows the Moon, Earth, and Sun. The user will also be able to click the 'About' pages, which show information about the project and group members.

## 3.2 Database Design

To use YourMoon, our team set up their own MySQL server using MySQL Workbench and XAMPP control panel for the server. Once the server is set up, team members can create a database and tables to store data. The database is then connected to the website using Express.js. The connection is established through a configuration file that contains the database credentials. The database can then be accessed by the client-side Vue.js framework to upload and retrieve data. YourMoon is a supplement website that allows users to upload their moon images to a MySQL database using Vue.js Framework for the client side and Express.js for the server side, while using JavaScript to handle upload features to the database. In addition, users can crop their images before uploading using a tool implemented with JavaScript, CSS, and HTML. This feature enhances the accuracy of our circle detection algorithm, allowing for better identification the moon in the images. To set up the MySQL server, users can use MySQL Workbench and XAMPP control panel for the server. With this implementation, YourMoon provides a user-friendly interface for image upload and processing, improving the accessibility of moon image data for research and educational purposes.



## Glossary

| Acronym | Long Version                  |
|---------|-------------------------------|
| SRD     | Software Requirement Document |
| UI      | User Interface                |
| SDD     | Software Design Document      |
| MLA     | Modern Language Association   |
| JPL     | Jet Propulsion Laboratory     |
| MT      | Moon Trek                     |
| UIM     | User Interface Module         |
| RM      | Registration Module           |

## References

| Reference Name                | Source  |
|-------------------------------|---|
| Software Design Document      | CSULA   |
| Software Requirement Document |   |
| MoonTrek API                  | <a href="https://trek.nasa.gov/tiles/apidoc/trekAPI.html?bod">https://trek.nasa.gov/tiles/apidoc/trekAPI.html?bod</a> |
| Django Web Framework          | <a href="https://docs.djangoproject.com/en/3.1/">https://docs.djangoproject.com/en/3.1/</a>                           |
| ASCOM Standards               | <a href="https://ascom-standards.org/">https://ascom-standards.org/</a>   |