Software Requirements Specification

for

Operationalize Networked Collaboration Features for Moon Trek

Version 1 approved

Prepared by: Sean Chung, Aldo Gil I, Tommy Lay, Allen Marquez, Tam Nguyen, Alex Sahakian, Andy Tsan, Srivats Venkataraman, Jian Wu, Anna Yesayan

Sponsored by NASA JPL

April 29, 2022

Revision History

Name	Date	Reason for Changes	Version
Srivats Venkataraman	October 8, 2021	- Added new cover page - Added new TOC - Added section 1.3, 1.4, 1.5	0.1
Anna Yesayan	October 10, 2021	Added 2.7, 2.8	0.1
Alex Sahakian	October 13, 2021	Started Section 2.2, 2.3, 2.4, and 2.6	0.1
Anna Yesayan	October 14, 2021	Added 2.5, 2.1(part 1)	0.1
Tam Nguyen	October 26, 2021	Added section 5	0.2
Srivats Venkataraman	October 26, 2021	Updated sections 1	0.2
Anna Yesayan	October 26, 2021	Updated Section 2.1,2.4,2.5,2.9	0.2
Alex Sahakian	October 27, 2021	Added onto section 2	0.2
Sean Chung	October 27, 2021	Updated section 3.3, specifically the tech stack.	0.2
Whole Team	April 29, 2021	Updates sections to match code base and application changes	1

1. Introduction

The software requirement specifications document will cover the features, requirements, and implementation of the Networked Collaboration Features for Moon Trek (NCFMT). This document will provide the reader with a summary of how the project was developed, the requirements, and the functionality of the project. NCFMT will give users collaborative abilities such as displaying annotations between users, communication via chat box, navigation through the terrain, setting waypoints, and viewing satellite imagery of the moon together.

1.1 Purpose

The purpose of NCFMT is to implement and enable collaborations between a group of users within a collaboration room. The collaborative features that are developed will be merged into JPL's existing Moon Trek website.

1.2 Intended Audience and Reading Suggestions

This document is intended for project managers, developers, university faculty, students, and other members that are part of the development process of NCFMT. This includes the JPL team who is coordinating the project, the Senior Design team who is developing the project, and the advisor to the Senior Design team. This document's recommended reading order is to read the Overall Description section to get a general idea of the document before navigating to another section. The reader may also check the table of contents to find the specific section they are looking for.

1.3 Product Scope

This software is identified as the Networked Collaboration Features for Moon Trek. NCFMT shall provide collaborative markup of 3D solar system terrain such as the creation of waypoints, rapid navigation toward waypoints, text annotations, and freely drawn "ink" annotations on the new Moon Trek Lite application. NCFMT shall also "rooms" that let the user markup and share different states of 3D solar system terrain simultaneously, communicate via a text communications system, and create waypoints for rapid navigation. Upon release, NCFMT shall be completely open to the public and be used for scientific research, mission planning, educational purposes, and general exploration.

1.4 Definitions, Acronyms, and Abbreviations

- NASA - National Aeronautics and Space Administration
- JPL - Jet Propulsion Laboratory

- NCFMT - Networked Collaboration Features for Moon Trek
- RTC - Real-Time Communication
- MLT - Moon Lite Trek
- SST- Solar System Treks
- Collaborative Session - An online environment in which users can communicate, exchange data, and work with one another.
- CVSST - Collaborative Visualization for Solar System Trek

1.5 References

- **CSULA CS Department & Senior Design Faculty**, SRS-TemplateSeniorDesign_v2_2020-10-23.1
    - SRD document provided by the CS department for this project
- **Collaborative Visualization for Solar System Treks**, Collaborative Visualization for Solar System Treks
    - SRD documentation writer by the Fall 2021 Senior Design Team

2. Overall Description

2.1 System Analysis

1. Project Goals

1.1 The current Moon Trek web application does not have a collaboration network. The goal of the project is to implement collaborative features so that users are able to collaborate within the Moon Trek web application. Users will be able to create a room of their own and share the session ID for other users to join that room. Once the user joins a session, collaborative features that are planned to be implemented include the ability to chat with other users within the same room, creating waypoints, use of drawing tools to create annotations for other users to see, and view the list of participants. Users will also be able to create and save states within the session that allow the user to switch between one state of an annotated map to another. The frontend and backend technologies will also have to be converted from Dojo Toolkit to Angular for the frontend and a NodeJS based backend to a Java based backend.

2. Major Technical Hurdles

2.1 Networking

2.1.1 Since networking is one of the main goals of the project; there will be multiple sessions connected at the same time. There may be some issues with making sure the sessions synchronize.

3. Possible Solution

3.1 Networking

3.1.1

2.2 Product Perspective

Moon Trek is a system currently built to explore the moon. This program shares similar features to Google Earth. It allows you to explore the moon with similar abilities as Google Earth as it may also log where you are viewing the planet. Where it is different is what is presented; There are multiple locations on the moon that can be found that have a pop-up tab that shows historical events on the moon such as the moon landing with Apollo 11 during 1969.

2.3 Product Functions

Moon Trek will feature:

1. Ability to scroll and explore the moon in a 3D environment

2. Points of interest that will show a location accompanied with written history on that particular location
3. Chat logs
4. Create/Join Room
5. Room synchronization
   6.1 Generate room ID link for host to share
6. Ability to share location via messages to a particular location

2.4 User Classes and Characteristics

We anticipate that Moon Trek will be used by many users. Moon Trek will operate as an educational tool as it can be used by instructors, students, scientists for self-education as it is aimed to educate users about the moon and its history.

2.4.1 Scientists

- Scientists have an anticipated use for scientific purposes.
- Scientists have a high level of frequency of use.
- Scientists have a high level of education.
- Scientists have a high level of experience with the SST.
- It is necessary for Scientists to access and use the SST with the highest level of functionality that can be achieved to assist them with their research.

2.4.2 Instructors

- Instructors have an anticipated use for scholastic purposes.
- Instructors have a mid-level of frequency of use.
- Instructors have a high level of education.
- Instructors have a mid-level of experience with the SST.
- It is necessary for Teachers to have the basic functionality of the SST, as all the tools present in the SST may not be needed by a teacher.

2.4.3 Students

- Students have an anticipated use for scholastic purposes under the guidance of a Teacher.
- Students have a low level of frequency of use.
- Students have a low level of education.
- Students have a low level of experience with the SST.
- It is necessary for students to at least have a low level of functionality with the SST, as Students shall be receiving directions from Teachers and therefore, shall not be needing full functionality.

2.5 Operating Environment

The software is a web application, hence the operating environment requires a stable connection and some sort of a web browser. The popular ones that are frequently used are Google Chrome and Internet

Explorer. The application can be served locally or on the cloud, which will be accessed via a web browser. NASA will host this program on a Java backend with a Typescript and Angular front end for the users.

2.6 Design and Implementation Constraints

The program may need users to operate a stronger computer as Java will require a lot of resources and it may require a lot of VRAM as it is dealing with 3D graphics. It also applies to web browsers as they may either:

- Limit resource usage due to security purposes causing limited available features to the end user
- Completely blocking the program from the user's web browser
- The use of more resources to give the full experience for the front end to the end user.

2.7 User Documentation

- Software Requirements Specification (.pdf, .docx)
- Software Design Document (.pdf, .docx)
- Fall 2021 Presentation Slides (.pdf, .pptx)
- Project Plan (.xlsx)
- Final Project Report (.pdf, .docx)

2.8 Assumptions and Dependencies

2.8.1. Users are expected to have consistent and stable Internet connections to use this web application.
2.8.2. Users are expected to be familiar with an Internet browser and be familiar with handling the keyboard and mouse.
2.8.3 Developers are expected to know how to configure, maintain the servers, and maintain the program.

2.9 Apportioning of Requirements

Requirements that might be delayed until future versions:

- 3D Graphics

3. External Interface Requirements

3.1 User Interfaces

- This system shall provide the user with intuitive input systems for the user
    - These input systems include buttons, sliders.
- This system shall provide a feature where users can use text communication
- This system shall provide a feature where users can share SST together
- This system shall provide a feature where users can invite a person to a room session
- This system shall provide a feature where users can share changes made on the planet with another user
- This system shall follow the color scheme, design, and theme of the underlying SST software
- This system shall provide a feature where users can save the state of a session for a given planet
- This system shall provide a feature where users can annotate on the planet
- This system shall provide a feature where users can search an area of the planet and navigate to that area

3.2 Hardware Interfaces

To run this software, users will need a well-performing laptop/computer. At least, 4GB ram, enough to run a modern web browser.

3.3 Software Interfaces

The TREK Lite software application utilizes a variety of APIs to function. These APIs include: Search API and functions, Fly To function, layers and globe, WMTS API, current user interface, geospatial functions to calculate distance, elevation profile, sun angle and distance, and finally, 3D print functions. TREK Lite is built using commercialized software; interface interactions flow between Java and other software such as JSON parsers, database queries, or API responses.

- The list below is software used by the TREK Lite app:
    - **Tomcat 7.0.59, https://archive.apache.org/dist/tomcat/tomcat-7/v7.0.59/bin/**
        - Used for testing the software environment through localhost.
    - **Esri ArcGIS API for Javascript Version 4. x, https://developers.arcgis.con/javascript/3/jssamples/**
        - Generates 2D visuals for the application.
    - **Java WebSockets, https://socket.io/**
        - Primarily utilized for funneling data between clients to the server in real-time with low latency.
    - **Eclipse Jersey, Eclipse Jersey (eclipse-ee4j.github.io)**

    - Used for connecting server clients together in real-time.
- **Existing APIs in use in the existing Trek application**
    - Primarily utilized for funneling data between clients to the server in real-time with low latency.

3.4 Communications Interfaces

The TREK application requires users to use a web browser of type WSS since it ensures an entirely secure connection using SSL. In addition, the online application queries data from JPL's APIs via JSON and text/file format. When the WSS connection is turned on, the software will synchronize the user to the web server via serialization and processing. The software uses Websockets since it is capable of creating a full-duplex and low-latency communication system. Still, Websockets are prone to data transfer issues when used in a large-scale system. Moving on, the software's backend system will require a fast database to avoid any potential network bottlenecks. Users will be able to use a chatroom service that allows them to connect with other individuals in real-time.

4. Requirements Specification

4.1 Functional Requirements

4.1.1. The system shall allow users to host a "collaborative session" (See Section 1.4).

i. The system shall reflect all visuals (See Section 1.4) to other users in a session as they are placed.

ii. The system shall synchronize all visuals between users in a session unless configured otherwise.

iii. The system shall share the same 2D/3D solar system terrain between users in a session.

iv. The system shall display the same 2D/3D solar system terrain between users in a session.

v. The system shall provide a text communications system for users in a session.

vi. The system shall allow users to invite other users to their session.

4.1.2. The system shall provide tools for markup onto 2D/3D Solar System terrain in varying sizes and colors.

i. The system shall provide an "ink-based" annotation tool for free-hand drawing.

ii. The system shall allow users to draw simple 2D/3D shapes (squares, circles, triangles, etc.) on the 2D/3D terrain.

iii. The system shall allow users to project 2D/3D text onto the 2D/3D terrain.

iv. The system shall allow users to create polyline (See Section 1.4) annotations.

4.1.3. The system shall allow users to create interactive navigation waypoints on 2D/3D Solar System terrain.

i. The system shall allow users to move the navigation waypoints.

ii. The system shall allow users to modify the navigation waypoints.

4.1.4. The system shall allow users to create multiple "SST states" (See Section 1.4) within a session.

i. The system shall allow users to access different states within a session.

ii. The system shall reflect all changes made within each SST state.

4.1.5. The system shall allow users to save their current "SST state".

i. The system shall allow the user to extract all markup to be saved for later use.

ii. The system shall allow the user to extract layer data to be saved for later use.

iii. The system shall allow the user to import saved markup to be visualized onto SST.

iv. The system shall allow the user to import saved layer data to be visualized onto SST.

4.2 External Interface Requirements

The software system is supported by the following hardware mediums:

- Mobile device support
- PC browser support
    - Mouse movement
    - Web Scrolling
    - Keyboard press events
    - Mouse click events

4.3 Logical Database Requirements

- Types of information used by various functions.
    - HTTP/ JSON Requests
    - Model Layer Data
    - Mesh data
    - Coordinate data
    - Text data
    - Session Data

- Types of information used by various functions
    - Frequency of use
        - Data pulled based on every host request of the session
    - Accessing capabilities
        - Database access based on the current session host and members
    - Data entities and their relationships
        - Solr
        - ArcGIS
            - Accumulo
    - Integrity constraints
        - Database must support NASA's Trek Data
        - Current Session Memory size

    - Security
- Data retention requirements
    - Data shall be saved temporarily during each session
    - The system shall allow users to revisit previously used data from the current session

4.4 Design Constraints

With the use of certain frameworks in the current Solar System Trek software, issues may arise when tackling networked sessions or free-hand “ink” annotations. It is also unknown how much network traffic shall be utilized concurrently between the server and client and how much server resources may be used in processing requests and other data. This brings up the possibility of the software not being very scalable. There are also hardware limitations, while a small number of entities and annotations on the 3D solar system terrain might not be a problem, larger amounts of entities may cause issues for devices with low graphics processing power.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

Some performance requirements that the application shall have:

- The system shall transfer data from client to server in real-time.
- The system shall transfer data with low latency.
- The system shall run smoothly in real-time such that there are no noticeable performance or rendering issues.

5.2 Safety Requirements

There are no safety requirements that need to be taken into consideration when using the CVSST. Even though the software and controlling system do not pose any harm toward users, security for user data will be implemented.

5.3 Security Requirements

Every user's identity and private data shall always be protected. Hosts shall be authenticated upon request, in order to grant larger freedoms within the application. HTTPS secure protocols will also safeguard encrypted HTTP data transferred over a secure connection.

5.4 Software Quality Attributes

- Adaptability: Currently has user collaboration, no virtuality reality.
- Availability: It is accessible through a website.
- Correctness: Shows the planet from NASA Trek API.
- Flexibility: It has features for Researchers, Teachers, and Students.
- Maintainability: Grabs data from the NASA Trek API.
- Portability: Can support Desktop, Laptop, and Mobile devices.
- Usability: Simple User Interface for all users

5.5 Business Rules

The product will be accessible to any user.

6. Legal and Ethical Considerations

Discuss any legal and/or ethical issues involved in the project. Justify the decisions made based on legal and/or ethical principles.

Appendix A: Glossary

Define all the terms necessary to properly interpret the SRS, including acronyms and abbreviations. You may wish to build a separate glossary that spans multiple projects or the entire organization, and just include terms specific to a single project in each SRS.
If this section is very short you may include it in section 1.4. If your list is very long you may include it here and put a reference to this Appendix in section 1.4.

Appendix B: Analysis Models

Optionally, include any pertinent analysis models, such as data flow diagrams, class diagrams, state-transition diagrams, or entity-relationship diagrams.

Appendix C: To Be Determined List

Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.