
Software Requirements Specification

for

TraViz - Web Travel Tool

Version 1.0 approved

Prepared by

Arshiya Khattak

Gopika Krishnan

Matthew Varona

New York University, Abu Dhabi

October 2021

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction	1
1.1 Purpose	1
1.2 Document Conventions	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Product Scope	1
1.5 References	1
2. Overall Description	2
2.1 Product Perspective	2
2.2 Product Functions	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment	2
2.5 Design and Implementation Constraints	2
2.6 User Documentation	2
2.7 Assumptions and Dependencies	3
3. External Interface Requirements	3
3.1 User Interfaces	3
3.2 Hardware Interfaces	3
3.3 Software Interfaces	3
3.4 Communications Interfaces	3
4. System Features	4
4.1 System Feature 1	4
4.2 System Feature 2 (and so on)	4
5. Other Nonfunctional Requirements	4
5.1 Performance Requirements	4
5.2 Safety Requirements	5
5.3 Security Requirements	5
5.4 Software Quality Attributes	5
5.5 Business Rules	5
6. Other Requirements	5
Appendix A: Glossary	5
Appendix B: Analysis Models	5
Appendix C: To Be Determined List	6

Revision History

Name	Date	Reason For Changes	Version

--	--	--	--

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of TraViz which is a Web Travel Tool. It will explain the purpose and features of the system, explain what the system will do, interfaces of the system, and constraints that will be considered when the system is in operation. This document is intended for the stakeholders of the system including the developers and end-users.

Finding an all-in-one system for travel and tourism-related information for different locations can be cumbersome as features like airfare prices, popular tourist attractions, food, etc. of a location usually need to be looked up separately. There is a need to have a space where such information is readily available on one click with the information presented in a concise manner, wherein the user can get a quick overlook of a city they are interested in traveling to. Moreover, the same system keeps track of the user's travel history. This would be appealing to a large audience as the website would provide custom-build visualizations to the users based on their history which would serve as a memoir of their travels and a basis for their decisions for future trips. Based on user history, the system also recommends potential places to travel to the user. This is an alluring feature as it provides suggestions that are custom picked towards users' tastes. Thus, this system would incite the desire to travel in the user, and in turn, this would help in boosting the tourism and travel industry.

This document will go through the above-mentioned features in more detail in terms of how it will be achieved, requirements and constraints.

1.2 Document Conventions

This document is formed using the IEEE template for System Requirement Specifications (SRS) documents. The text follows the Times New Roman font, size 12, while the headings are in bold and in title case. Italicized text is used in the document for the purpose of emphasis.

1.3 Intended Audience and Reading Suggestions

This document is intended for all the stakeholders of the project especially the end-users and developers. For end-users, the overview section will have the most information and for the more technically inclined users, the External Interface Requirements and the System features sections will be insightful. The developers will find the same sequence helpful. While those in charge of the project would find the information in other nonfunctional requirements after reading the overview section more relevant. Skimming through the External Interface Requirements and the System

features sections will give a quick overview of the technical aspects of the system. Regardless, we suggest reading the overview section to get a quick look into the functionality of the system.

1.4 Product Scope

The system is an all-in-one web travel tool that has three main functionalities. First, it assists the user in visualizing their travel experiences and provides data visualizations on other interesting features of these places. Second, it functions as a travel planning tool wherein users can enter destinations they intend to travel to, and the system in turn would provide highlights and interesting insights to their chosen destination. Given that the visualizations are custom-built for the users as per their requirements, it would be more favored over the information overload of travel-related information online. Third, it provides recommendations to users on places to travel to, based on their travel history. The latter two functionalities can be used when the user switches to the 'Planning mode' of the system.

1.5 References

"Flight Data API Documentation (Travelpayouts) | Rapidapi". Rapidapi.Com, 2021, <https://rapidapi.com/Travelpayouts/api/flight-data/>.

"Locationiq - Free & Fast Geocoding, Reverse Geocoding And Maps Service". Locationiq.Com, 2021, <https://locationiq.com/docs>.

"Sygic Travel API". Docs.Sygictravelapi.Com, 2021, <http://docs.sygictravelapi.com/1.1/>.

"Travelperk | Developers". Travelperk | Developers, 2021, <https://developers.travelperk.com/>.
(WAI), W3C. "Introduction To Web Accessibility". Web Accessibility Initiative (WAI), 2021, <https://www.w3.org/WAI/fundamentals/accessibility-intro/>.

2. Overall Description

2.1 Product Perspective

This product is developed for everyone who's interested in traveling and for an audience who is interested in learning more about the places they've been to and the places they would like to go to in the future. It is a new self-contained product that provides a personal touch to travel-related systems as it takes in input from the user and produces visualizations related to travel based on their input as well as recommends places to the user based on their history and interests in travel. Figure 1 represents major components of the system and how they interact with each other.

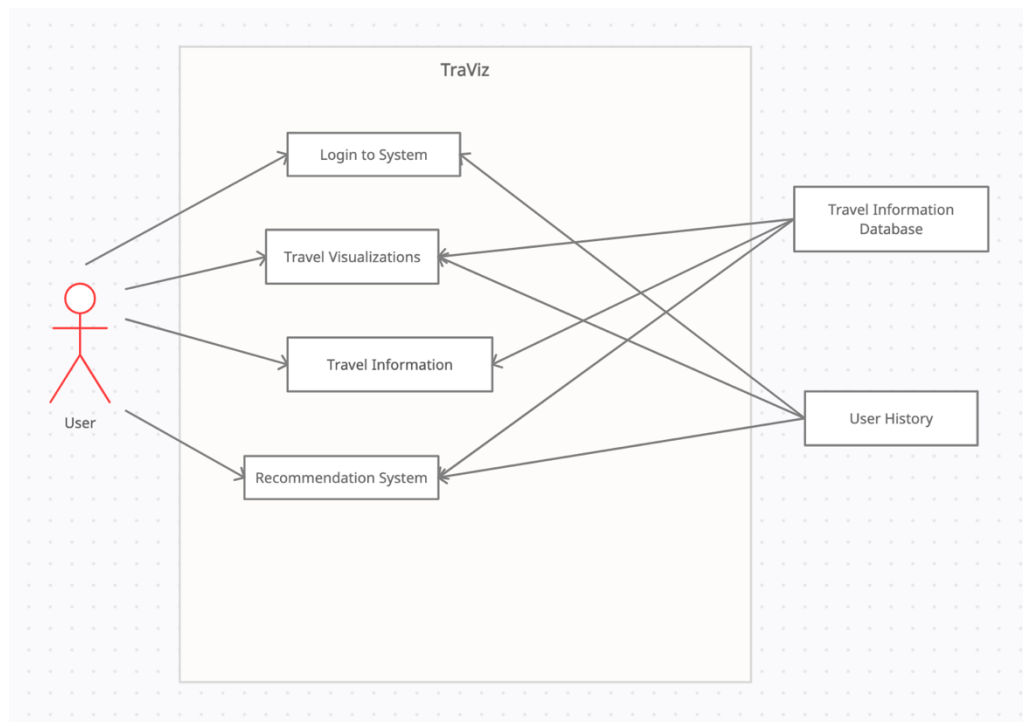


Figure 1 - Overview of the System

2.2 Product Functions

- Login – Allows users to create a profile and log in the places they’ve traveled to and search for places they are interested in
- Generate location visualizations – Based on the input of the places users have been to so far, visuals with a map showing the places they’ve been to are generated
- Generate more information on location – Based on the locations the users have been to, visualizations are produced about features of the place like the points of interest (POIs) of a location, average weather, food, etc.

- Learn more about location – On inputting a location the user is interested in traveling to, the system provides a pocket-size guide filled with visual information on highlights of the place like POIs, weather, food, etc.
- Suggest a location – Based on the locations the user inputs as location history and their rate of experience in each place, the system recommends the user with locations they might like
- Export file as SVG/PNG/PDF – Print out the travel-related visualizations produced by the system; recommendations provided by the system as well as information related to different locations

2.3 User Classes and Characteristics

The system has one actor, the cooperating system, and three core use cases. The primary actor who is the user of the system can be either a logged-in member of the system or a guest user. A guest user can create a profile and search for information related to locations they input. A logged-in user has more functionalities including access to data visualizations of places they've traveled to, additional information related to these places, and recommendations of locations. There are two main databases – one being the user profiles and their interests, two being information related to locations.

2.4 Operating Environment

The software system operates in common web browser environments including Google Chrome, Apple Safari, Internet Explorer, Mozilla Firefox using SSL/TLS encryption protocols.

2.5 Design and Implementation Constraints

The system should be developed and running in the span of 2-3 months, there is a firm deadline set in December. All maintenance work should be scheduled between 07:00 and 09:00 and should be designed to be available to end-users 24 hours, 7 days a week. Considerations related to security and privacy must be given, especially to the login system and with the recommendation system (if the recommendation system is designed to work on data of other users as well).

2.6 User Documentation

User documentation will be done in the form of guidelines, FAQs (Frequently Asked Questions), and a help section. They will be available on the system website in the form of PDFs (Portable Document Format) and on the website as HTML (HyperText Markup Language).

2.7 Assumptions and Dependencies

No assumptions are considered at the time. No dependencies are identified for the project so far.

3. External Interface Requirements

3.1 User Interfaces

3.1.1 Visualizing Travel History

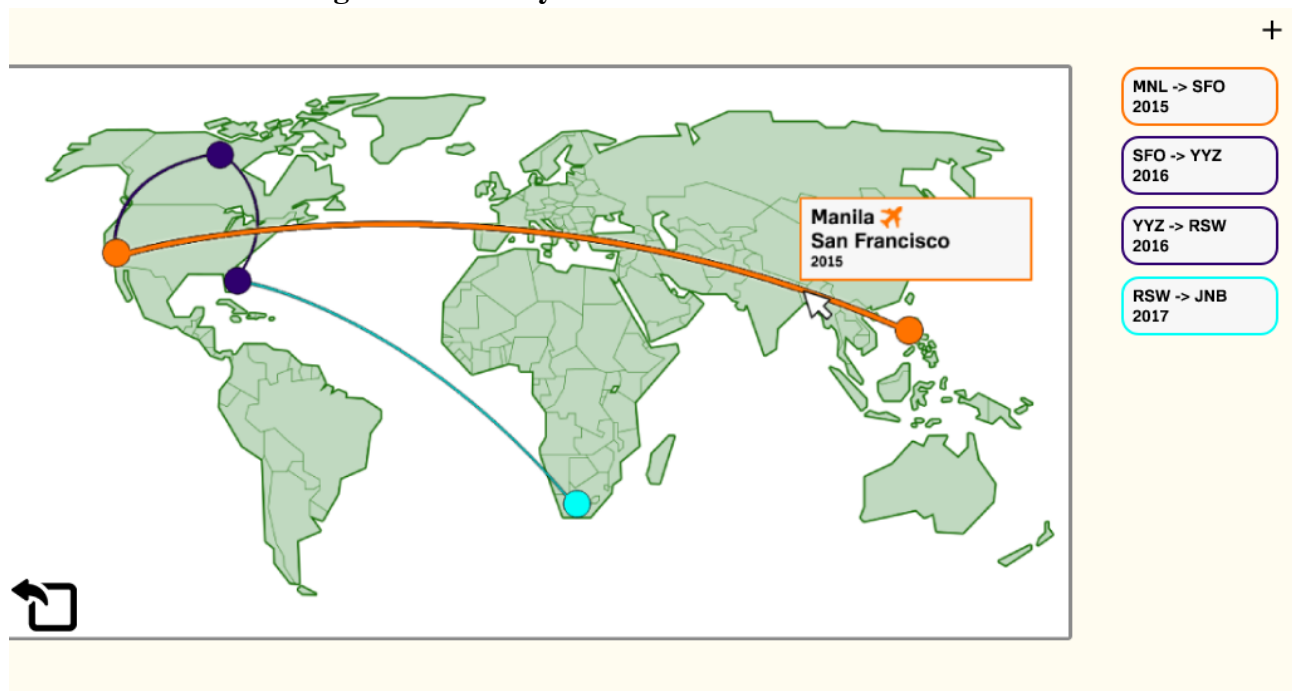


Figure 2 - Mockup of user interface

The travel history interface is composed of 3 main parts:

A. Adding Trips

By clicking on the plus sign in the top right, users can add a trip. Users will enter the start and end points of the trip, the mode of transportation used, the date, and will also be able to upload a picture of the trip.

B. Travel Timeline

Trips that have been added by the user are displayed in a timeline on the right. Hovering over a trip will display its full details in a card, as well as the uploaded picture if applicable.

C. Travel Map

Trips are also visualized on the world map, and are color-coded based on the mode of transportation (boat, bus, plane, etc.) Hovering over a trip will display its full details in a card, as well as the uploaded picture if applicable. Clicking the “Export” button on the bottom left exports an image file of the map.

3.1.2 Planning Trips

Users can interactively plan trips. When users add a trip with a date in the future, our app will automatically retrieve helpful information about the trip and display it on the card:

- Flight ticket costs (only for flight-based trips)
- Points of interest in the location
- COVID-19-related travel restrictions

3.2 Hardware Interfaces

Our system will not be particularly dependent on hardware, although it will be optimized for desktop web (as maps are hard to deal with on mobile devices).

3.3 Software Interfaces

Our system will use the following software components:

D3.js: Javascript-based visualization library.

Javascript: For interactivity throughout the web app.

Flask: Python-based backend library.

LocationIQ: API for geocoding. Will be used for determining locations on the map visualization. Location data (coordinates) will communicate with the D3.js part of the app to accurately display the locations on the map.

Sygie Travel: API that returns points of interest in cities. Will be used for the trip planning function. Retrieved data will be displayed using D3.js.

Travelpayouts: API that returns flight ticket costs. Will be used for the trip planning function. Retrieved data will be displayed using D3.js.

TravelPerk: API that returns travel restrictions in given locations. Will be used for the trip planning function. Retrieved data will be displayed using D3.js.

****NOTE:** The last 4 software interfaces may be subject to change or removal depending on financial constraints. Some of these APIs advertise free trials, but these free trials may not actually be suitable for our project (limited/missing functionality). In case we cannot find an appropriate free API for a particular function, that function may be removed.

3.4 Communications Interfaces

Our system will have to be able to communicate with the APIs listed above. Otherwise, there are no special communication requirements.

4. System Requirements

4.1 Visualizing Travel History

4.1.1 Description and Priority

The “travel diary” portion of the project. Users can add trips and see them visualized in unique ways.

Priority: **10/10**. This is the unique selling point of our project; while many travel planning apps exist, few apps exist for systematically and visually documenting travel history in the same way that TraViz does.

4.1.2 Stimulus/Response Sequences

User: Click “Add trip” button

System: Request trip information (start, end, date, mode of transport, optional photo)

User: Enter trip information and confirm addition

System: Add trip to database, visualize trips on timeline and map

User: Hover over trip

System: Display additional information about trip

4.1.3 Functional Requirements

REQ-1: Contain user interface objects for adding trips and entering information.

*The user can only click “confirm” once the required information is added (start, end, year, mode of transport). Otherwise, the button will be grayed out, preventing invalid input.

REQ-2: Communicate with the geocoding API, LocationIQ, when users enter the start and end locations. The system should retrieve the coordinates of entered locations from the API.

REQ-3: Visualize the travel data accordingly. The visualizations should include a world map using D3.js, as well as a timeline using vanilla Javascript.

REQ-4: Interactivity. Users should be able to hover over entries in the timeline or map, at which point the system will display a card with additional information about the trip.

4.2 Planning Trips

4.2.1 Description and Priority

The “travel planning” portion of the project. When users add trips with dates in the future, the application will display helpful information about the potential trip.

Priority: **6/10**. This feature is not as unique as the visualization aspect of TraViz, and as such, people may prefer to use one of many other travel planning applications that are already out there. However, if the team’s schedule permits, this would be a “nice-to-have” feature which complements the user’s existing travel history with future plans.

4.2.2 Stimulus/Response Sequences

User: Add trip with date in the future

System: Identify the trip as a future trip

System: Communicate with APIs to get travel planning information (ticket price, tourist spots, COVID restrictions)

System: Add trip to database, visualize trips on timeline and map

User: Hover over trip

System: Display additional information about trip, INCLUDING travel planning information (ticket price, tourist spots, COVID restrictions)

4.2.3 Functional Requirements

REQ-5: Communicate with the travel interest API, Sygic Travel, when users enter the end location. The system should retrieve the top 3-5 tourist destinations in that location, as determined by the API, and save them. If no tourist destinations are found, the card will display a message, “No tourist locations found”.

REQ-6: Communicate with the flight ticket API, Travelpayouts, when users enter the start and end locations. The system should retrieve the 3-5 cheapest tickets for those destinations, as determined by the API, and save them. If the trip is not flight-based, this does not occur and the flight prices are not displayed. If no flights are found, the card will display a message, “No flights found”.

REQ-7: Communicate with the travel restriction API, TravelPerk, when users enter the end location. The system should retrieve the “authorization_status” attribute for the given destination. If no status is found, the card will display a message, “No status found”.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system should perform on par with industry-recommended standards in relation to storage and processing time to ensure maximum user satisfaction and utilization of the resources available for the system.

5.2 Safety Requirements

The system should have frameworks put in place to ensure the safety of the users in terms of their data. Also, the information showcased on the website should follow safety requirements set by the industry.

5.3 Security Requirements

The system should ensure the protection of user data as the data is sensitive. There is an authentication process to facilitate user profiles in the system. If the developers implement a recommendation system using data from users who have similar interests, then the system must ensure the privacy of this information and the team must ensure a framework that will prevent data leakage. Moreover, the system should follow industry standards for security practices.

5.4 Software Quality Attributes

The system must have the qualities of correctness, availability, and adaptability. The system is a facilitator of information related to the travel industry, and thus correctness is an important quality for the software. Moreover, given the dynamic nature of the industry, adaptability is important. Also, since there are no set business hours for the system as the nature of the industry means the end-users look up information any time, availability is an essential quality to the system.

5.5 Business Rules

The system must grant special access to more features (data visualizations, recommendations) to users logged into the system. So, there must be a guest role as well as a logged-in user role for the user class in the system.

6. Other Requirements

6.1 Accessibility and Inclusion

As much as possible, we hope to develop our project in line with [Web Accessibility Initiative accessibility principles](#). These requirements may include, but are not limited to:

- Compatibility with screen readers
- Ability to navigate the website using only keyboard or only mouse
- Inclusion of “alt text” where relevant
- High text readability through whitespace (line length, text spacing)
- Accessibility-friendly color choices (in terms of contrast, etc.)

The team will keep these principles in mind throughout the development of the project. It is important to us that users spanning a wide range of abilities can use and enjoy our service.

6.2 Code Reusability

The codebase of the project should be constructed such that it would be easily reusable for future projects involving data visualization. This would mean that any future expansions of the project, or related projects, would be easier to develop.