

---

# **Software Requirements Specification**

## **for Pandemic Tracker**

**Version 1.0 approved**

**Prepared by Máté Hekfusz, Daniel de Beer, Mar Jaramillo, and Sangjin Lee**

**New York University Abu Dhabi**

**10 April 2020**

## Table of Contents:

<b>Introduction</b>	1
Purpose	1
Intended Audience and Reading Suggestions	1
Product Scope	1
<b>Overall Description</b>	1
Product Perspective	1
Product Functions	2
User Classes and Characteristics	2
Operating Environment	2
Design and Implementation Constraints	3
User Documentation	3
Assumptions and Dependencies	3
<b>External Interface Requirements</b>	3
User Interfaces	3
Hardware Interfaces	3
Software Interfaces	4
Communications Interfaces	4
<b>System Features</b>	4
Data Retrieval through Web Scraping	4
Data Storage	5
Country Specific Data Portrayal and Data Comparison	5
<b>Other Nonfunctional Requirements</b>	6
Performance Requirements	6
Safety Requirements	6
Security Requirements	6
Software Quality Attributes	6
Business Rules	6

## Revision History:

Name	Date	Reason For Changes	Version
------	------	--------------------	---------

<i>Version 1.0</i>	<i>10/07/2020</i>	<i>SRS document created</i>	<i>1.0</i>

# **1. Introduction**

## **1.1 Purpose**

This SRS covers the software requirements of the entirety of the Pandemic Tracker 1.0 program. It provides a reference for developers while building the software and a roadmap for the customer organizations while navigating it.

## **1.2 Intended Audience and Reading Suggestions**

This document is intended as a reference for all active members involved in the construction and testing phases of the software, including developers and project managers. The SRS contains the scope of the project, its overall description, external requirements, system features, and functional and nonfunctional requirements.

Those wishing to become familiar with Pandemic Tracker must read the document in order from the start, while developers can rely more on System Features in Section 4.

## **1.3 Product Scope**

The Pandemic Tracker 1.0 will aggregate information from trusted sources (WHO, CDC, Johns Hopkins etc.) about the current Covid-19 epidemic. The data will not be limited to just the number of cases but will include other information, such as travel bans and country preparedness. The system will present all this information in an accessible manner to end users, allowing them to browse global as well as national data. Users will be able to rate the information that is presented to them in terms of usefulness.

The platform will help citizens avoid infection, which will reduce the economic strain of the epidemic on countries. If users refrain from travelling (or become better at avoiding infection) because of information from the service, that will save governments the cost of healthcare and containment and continuous lockdown.

# **2. Overall Description**

## **2.1 Product Perspective**

Currently, there is much spread of misinformation around the world concerning current events, especially with the onset of the novel Coronavirus (COVID-19). This leads to fear, panic and major financial and social damage.

The idea for Pandemic Tracker came from the desire to have a centralized platform to render up-to-date and factual information about world health. Our goal is to have a one-stop service that people can rely on and which will help resolve uncertainty and tackle biased media coverage.

This new, self-contained product relies on APIs from numerous external platforms like the CDC, WHO, and John Hopkins in order to provide a unified service in which citizens can stay up to date on travel restrictions, government regulations, pandemic data, and healthcare options.

## **2.2 Product Functions**

- Data Access (front end)
  - Worldwide
    - Data (Number of new infections, deaths, recovery, curves)
    - Information about pandemic
    - WHO and CDC guidelines
  - Per Country
    - Travel restrictions
    - Data (Number of new infections, deaths, recovery, curves)
    - Government restrictions
    - Healthcare options (respirator availability, masks, hand sanitizers, hospitals)
- Data retrieval (backend)
  - APIs
  - Web scraping
- Data storage (backend)
  - Custom Amazon Web Services database

## **2.3 User Classes and Characteristics**

All users of the software will have access to the safe interface and functionalities, but there are some users that will have priority interests in specific functionalities of the software. This is outlined as follows:

- **Users:** All concerned citizens
  - Business Travelers:
    - Economic effects of the pandemic
    - Travel restrictions
  - Students:
    - Travel restrictions
  - Healthcare workers:
    - Healthcare options
  - Infected or potentially infected individuals
    - Healthcare options:
  - Government employees:
    - Travel restrictions, Foreign Government Data

## **2.4 Operating Environment**

The tracker will be a website hosted on Amazon Web Services, meaning it will be accessible to all operating systems and web browsers. The website will be accessible for mobile users as well. This is to allow maximum ease of access to those who are without reliable web service.

## 2.5 Design and Implementation Constraints

The database of information will need to be constantly updated, meaning the service will have to be connected to the Internet. Real-time updating will be constrained by the issues we may encounter while website-scraping: some services are difficult to scrape, while some might limit the availability of information to the public. In addition, though we will be very selective about the platform's sources, we will inevitably have to rely largely on governments being truthful in their published data. Furthermore, healthcare availability promises to be the most difficult statistic to accurately track.

## 2.6 User Documentation

Though we plan our program to be user-friendly and easy-to-use, we will include an about page with all main functionality described therein, and some diagrams portraying how to access the various functionalities of the PandemicTracker.

## 2.7 Assumptions and Dependencies

We are completely reliant on third-party data, from government websites and pandemic information compilation sources. We are assuming that we will have access to up-to-date information regarding travel restrictions, government regulations, healthcare services, and infection data. However, not all websites may not have user-friendly APIs, or may be difficult to scrape due to privacy concerns.

# 3. External Interface Requirements

## 3.1 User Interfaces

1. We will have a home screen that provides the most urgent information: the global number of infections, recoveries and deaths.
2. After this home screen we will have two main user pathways:
  - a. Comparison between different countries and their data concerning the pandemic (will be limited to comparing case data and travel restrictions).
  - b. Specific data about a single country. The categories will be as follows:
    - i. travel restrictions
    - ii. case data (infected, dead, recovered)
    - iii. healthcare availability
    - iv. government restrictions (lockdowns, fines, etc)
3. The user will also be able to access the up-to-date WHO and CDC guidelines for physical distancing and how to handle the disease.
  - a. A general information tab by category
  - b. Frequently asked questions

## 3.2 Hardware Interfaces

The main customer hardware interface will be smartphones, tablets and personal computers, used to access the website.

### 3.3 Software Interfaces

**Databases:**

We will use the external APIs and scraped information to update our own database, which we use to render the information in our website. We ensure that duplicated data is handled internally before rendering to the user. The databases are country specific.

**Tools:**

- For infection data, we will rely on the following John Hopkins repository:  
<https://github.com/CSSEGISandData/COVID-19>.
- GHO Odata API - WHO's open data and statistics:  
<https://www.who.int/data/gho/info/gho-odata-api>
- Multiple APIs that provide COVID-19 status:  
<https://rapidapi.com/collection/coronavirus-covid-19>
- For the latest travel restrictions, the following website:  
<https://www.iatatravelcentre.com/international-travel-document-news/1580226297.htm>  
will be scraped. It includes all up-to-date travel regulations for each country.
- Hospital bed capacity per country:  
[https://en.wikipedia.org/wiki/List\\_of\\_countries\\_by\\_hospital\\_beds](https://en.wikipedia.org/wiki/List_of_countries_by_hospital_beds)

**Libraries:**

We will use the Python Pandas library to manipulate our scraped data. Beautiful Soup will be used to scrape the data.

### 3.4 Communications Interfaces

In order to continuously scrape the data, we will use HTTP requests and direct API access. As we will not directly be using user information on our platform, any communication security or encryption issues will not be a pivotal concern here.

## 4. System Features

### 4.1 Data Retrieval through Web Scraping

#### 4.1.1 Description and Priority:

Data retrieval will be done through web scraping of the most accurate and reliable websites available. Data retrieval is a **high priority** feature, as all other features depend on the proper and accurate functioning of the data retrieval.

#### 4.1.2 Stimulus/Response Sequences:

The server will have a timer that will continuously request data from the API and update the database. The user is not exposed to the web scraping.

#### 4.1.3 Functional Requirements:

DATR-REQ-1: Web Scraping from websites detailed in **3.3**

DATR-REQ-2: Transferring data to storage

## 4.2 Data Storage

#### 4.2.1 Description and Priority:

This feature is of **high priority** and it relies on a database that will be used for rendering.

#### 4.1.2 Stimulus/Response Sequences:

Stimulus: User clicks on the specific country / data they want to access and the

Response: Website renders the information in the database.

#### 4.1.3 Functional Requirements:

DATS-REQ-1: A functional running server

DATS-REQ-2: A running database

## 4.3 Country Specific Data Portrayal and User Data Comparison

#### 4.3.1 Description and Priority:

This feature is **medium priority**, it relies on the database to render and is a front-end feature that does not carry the whole system. This is one of the key features, but other features will still be useful if it fails.

#### 4.3.2 Stimulus/Response Sequences:

The initial step the user will take is accessing the website of the Pandemic Tracker. The user will not be required to login to the website, as this is an open platform for anyone that is concerned about health issues. The system will lay out all the available country lists and the data corresponding to it. The data will be of the country's travel restrictions, data (Number of new infections, deaths, recovery, curves), government restrictions, healthcare options (respirator availability, masks, hand sanitizers, hospitals).

The user will then choose which information they want to learn and by clicking the corresponding tab, the system will display the specific information. The user can also click on a button to compare between two different countries so then the system will give a visual comparative analysis on the two countries by displaying two statistics on a bar graph.

Another user stimulus would be when the user clicks on the tab for WHO/CDC guidelines and as the response the system will display the most up-to-date healthcare recommendations.

#### 4.1.3 Functional Requirements:

DATC-REQ1: Running database.

DATC-REQ2: Running server to continuously host the website. Needs to be able to handle high user traffic



## 5. Other Nonfunctional Requirements

### 5.1 Performance Requirements

When users interface with the platform, they must be able to access information quickly (within five seconds), at all times, and that information must be accurate (on a daily basis). This means that the frequency of data retrieval from third-party services will be balanced between speed and accuracy as well. The exact timings will be determined through testing; the default update rate will be daily.

### 5.2 Safety Requirements

Though we will attempt to only pull data from trusted sources, the accuracy of our service will depend entirely on the accuracy of the data provided by the third-party services, like governments and international organizations. Thus, we cannot take responsibility for inaccurate information and any harmful behavior it causes.

### 5.3 Security Requirements

Users will only be able to access the front end of the platform, the data display. There will be no ways to upload data or send anything to the database, meaning exploit risk is minimized. Since the program does not handle sensitive data (everything that is scraped or pulled from APIs is also publicly accessible), there are no privacy risks.

### 5.4 Software Quality Attributes

**Availability:** since our goal is to create a unified platform people can turn to whenever necessary, the platform must be available at all times.

**Reliability:** the platform must be able to pull data from several different sources at frequent intervals, merge them into our custom database, and present it to the users in the correct format.

**Usability:** the platform must be easy to navigate; information must be clearly presented.

### 5.5 Business Rules

- Only the developers must be able to access the data retrieval backend and the database created from it.
- Users must only see and interact with the data display frontend.
- Users must not be allowed to access the internal database.