Mission: The UMBC Covid-19 Tracker - Provide Covid-19 related case and vaccine information for public safety and knowledge

DID: Software Requirement Specification (SRS)

PREPARED by: Quinque Equos

(

[Alex Wilson](mailto:alexanw1@umbc.edu)

Safia Shah

[Israel Morocho](mailto:cx59298@umbc.edu)

[Duncan Taylor](mailto:ak58697@umbc.edu)

[Tom Tennant](mailto:tent1@umbc.edu)

[David Kravets](mailto:kravets1@umbc.edu)

)

DATE: MAR-2021

**Table Of Contents**

[**Introduction**](#_8h3d3lt852y2) **4**

[1.1 Purpose](#_bsz6prdlcf48) 4

[1.2 Scope](#_p751ziw6c6xr) 4

[1.3 Definitions, acronyms and abbreviations](#_jsscezvbzr24) 4

[1.4 References](#_vgsn0qag5caj) 4

[1.5 Overview](#_j7otus7i5te) 4

[**Overall Description**](#_1jt0mgdl5g59) **5**

[2.1 Product Perspective](#_w7r3uozdauwx) 5

[2.2 Product Functions](#_v0gxcer6vxhn) 5

[2.3 User Characteristics](#_7iibe564p8cf) 6

[2.4 Constraints](#_33y5u3h6jjdz) 6

[2.5 Assumptions and Dependencies](#_qiyo5x4reypt) 6

[**Specific Requirements**](#_d91bshcj6cy6) **6**

[3.1 External interface requirements](#_cbonce537kyg) 6

[3.1.1 User interfaces](#_x4n44hvk0fwp) 6

[3.1.2 Hardware interfaces](#_jb526azcje2b) 7

[3.1.3 Software interfaces](#_gfv74yc7w2jx) 7

[3.1.4 Communications interfaces](#_hnbk4o3bxvvg) 7

[3.2 Functional requirements](#_x8e2bmisxxeu) 8

[3.2.1 CT\_F\_1:](#_2jge2o9d3214) 8

[3.2.2 CT\_F\_2:](#_8iqa955pw2fs) 8

[3.2.3 CT\_F\_3:](#_1b7gd2d2s2zs) 8

[3.2.4 CT\_F\_4:](#_wtdez2kbdrdc) 8

[3.2.5 CT\_F\_5:](#_946dm7c8m3is) 8

[3.2.6 CT\_F\_6:](#_61w7cp7912j) 8

[3.2.7 CT\_F\_7:](#_1u6dihbbarka) 8

[3.2.8 CT\_F\_8:](#_eac76yq0mkh2) 8

[3.2.9 CT\_F\_9:](#_f2q09t8lnzo8) 8

[3.2.10 CT\_F\_10:](#_a0zhhmq82nrf) 9

[3.2.11 CT\_F\_11:](#_p8ot1spfi4vf) 9

[3.2.12 CT\_F\_12:](#_g3uuu2tyygrv) 9

[3.2.13 CT\_F\_13:](#_3k4cctgi37o) 9

[3.2.14 CT\_F\_14:](#_9vmpxfx5wufg) 9

[3.2.15 CT\_F\_15:](#_1luho313nogm) 9

[3.3 Configuration Requirements](#_jy8gl6drg16q) 9

[3.4 Error Requirements](#_9jqz94dhbocm) 9

[3.7 Qualification provisions](#_bc8pppg34dod) 10

[3.7 Software system attributes](#_8v5p0z88cd8g) 11

[3.7.1 Reliability](#_4gazf3mjw282) 11

[3.7.2 Availability](#_b379708k3y55) 11

**List of Tables**

Revision Log 2

Qualification Provisions 9

**List of Figures**

Cloud Level Use Case Diagram 5

Quinque Equos REVISION LOG

| Revision | By | Description | Date |
| --- | --- | --- | --- |
| A | Team | Initial Version | 03/03/2021 |
| B | Team | Updated requirements to reflect customer and TA meeting | 4/15/2021 |

# Introduction

## 1.1 Purpose

This Software Requirements Specification (SRS) pertains to the development of the UMBC Covid-19 Data Tracker (CDT). The document contains the functional and non-functional requirements of the application, as well as any assumptions made and required constraints to the development. The intended audience includes the engineers tasked with developing the application and the test conductors responsible for the verification and acceptance of the application.

## 1.2 Scope

The software to be produced is a web application called UMBC Covid-19 Data Tracker (CDT). This web application takes in verified Covid-19 case and vaccine data based on state and location to create a heat map to visualize the data. The tracker will not report on individual cases of Covid-19 where you would be able to tell if an individual has Covid-19. The web application will be using data from The Broadstreet Covid-19 Data Project to create the heat map, as their data is organized by state and county. Vaccine data and information will be received by a similar database. Settings will be implemented to allow a user to see cases, vaccinations, or both. Thus, creating an efficient software to aid in public safety and knowledge about the ongoing pandemic.

## 1.3 Definitions, acronyms and abbreviations

UMBC - University of Maryland, Baltimore County

CDT - UMBC Covid-19 Data Tracker

SRS - Software Requirements Specification

## 1.4 References

1. The Broadstreet Covid-19 Data project

<https://covid19dataproject.org/>

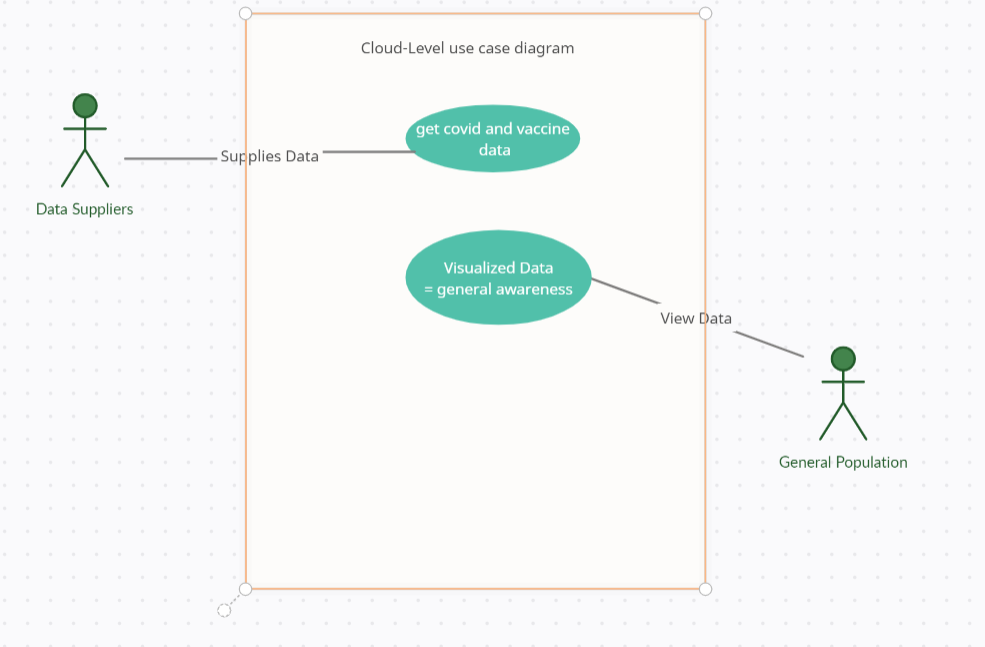
## 1.5 Overview

The rest of the SRS is organized as follows: section 2 provides an overall description, section 3 provides functional and nonfunctional requirements of the software. Within section 2, subsections 2.1 - 2.3 describe the product perspective, function, and user characteristics respectively. Of note, Sections 2.4 and 2.5 describe the constraints and the assumptions of the system. In the section 3 scope, section 3.2 lists the functional requirements, Sections 3.2-3.6 describe the non-functional requirements, and system attributes are presented in Section 3.7

# Overall Description

## 2.1 Product Perspective

The overall purpose of the UMBC Covid-19 Tracker is to provide information to the user base with regards to the virus and the vaccination. The application’s front-end will be broken up into those two listed parts. One page for the statistics of the virus, sorted maybe by county/city or by a different metric. Then another page with statistics of the multiple vaccinations that have been made available for Covid-19. We intend to intake the raw data and culminate it into a heat map in order to visualize the data into a more digestible and useful form. While there are many of these already in existence, we hope to make ours better and more importantly faster.



## 2.2 Product Functions

We will be getting raw data about Covid-19 and raw data regarding the multiple available vaccinations for the virus from multiple sources. We will then parse through the data and input it into a queryable database. The data will then be visualized either separately or together in the form of a heatmap. This is to both inform the general public and raise awareness of the danger of both the virus and certain areas.

## 2.3 User Characteristics

The users of the UMBC Covid-19 Tracker will be the general public with internet access. We intend to make this website accessible and user friendly to anyone who views it. To begin with, the users will be people who have some sort of interest in data for the general Maryland area. Potentially later in the development process, we may expand to other areas within the United States but we currently do not have plans to expand any wider than that region. The idea is that anyone who can use Google Maps, should be able to use and understand our product. To facilitate this, we may include a user help or teaching area in our web application on how to operate our product.

## 2.4 Constraints

One of our main constraints would be time. Given that our team is a collection of students, our time must be divided into this project as well as other subjects. The second constraint would be data storage, we will hold data of all Covid cases and Vaccine information initially for just Maryland but potentially for the entirety of the United States. As such we must store and retrieve our data as efficiently as possible so that it’s data size won’t slow down our product. Readability, as our product needs to be able to deliver its information in a way the average person can easily understand.

## 2.5 Assumptions and Dependencies

Our application’s main function is to display Covid cases and Vaccine information, as such it is heavily dependent on this data and it’s accuracy. We gather this data through a multitude of different sources, comparing their numbers in order to ensure an accurate data set. Our application is an online tool, therefore it depends on a strong internet connection to be high quality.

# Specific Requirements

## 3.1 External interface requirements

In the sections below we have defined the requirements for interactive elements between the user and the UMBC Covid-19 Tracker web application. These definitions cover interactive elements for user, hardware and software interfaces.

### 3.1.1 User interfaces

The web application will consist of various GUI components to build two distinct interactive views for the user.

* One view will allow users to select specific regions on a map and visualize associated covid case data based on their selection. This map will utilize a heatmap to display cases as well as supplementary graphs and statistical data the user can interact with to show trends of cases over time
* A secondary view that will show related vaccination data based on the same region selection process done by the user before

These two views are the primary definitions for interactive components on the user interface for the web application. This list is by no means exhaustive, as such an exhaustive list will be made available later in the Software Design Description (SDD).

### 3.1.2 Hardware interfaces

The web application will have hardware interaction abstracted through the modern web browser (such as Google Chrome). Resource requirements will need to be reviewed at a later time, this section will be updated to reflect the necessary requirements.

### 3.1.3 Software interfaces

The web application will interact with two externally defined software interfaces.

* The Covid-Tracker API
  + The web app sends requests for data to the API for a given geographical region notated by latitude and longitude. The API responds with the requested data.
  + Responses will include:
    - Reported cases by day, deaths by region, trends over time, vaccinations by area, etc.
    - This is not an exhaustive list and will be gone into more depth at a later stage
* DjangoDB
  + Aggregated covid case and vaccination will be stored in this database and directly interfaced through the Covid-Tracker API

### 3.1.4 Communications interfaces

A communication interface would be theUMBC covid tracker web page, and the accessible Covid-tracker API.

## 3.2 Functional requirements

#### 3.2.1 CT\_F\_1:

The application shall provide up to date Covid-19 case data, published from the Broadstreet data publication. (includes cases, probable cases, mortality, and probably mortality)

#### 3.2.2 CT\_F\_2:

The application shall present semi up to date, state-specific, closure information relevant to COVID19. (includes curfew, gathering sizes, etc.)

#### 3.2.3 CT\_F\_3:

Our application shall collect data concerning Covid-19 vaccine information. As a baseline will include vaccine information (number of people vaccinated, type, company, etc).

#### 3.2.4 CT\_F\_4:

Our product shall have an API able to translate the interaction between our database and user’s inputs.

#### 3.2.5 CT\_F\_5:

Our product shall have a database table to store our Covid-19 cases information

#### 3.2.6 CT\_F\_6:

Our product shall have a database table to store our Covid-19 vaccine information

#### 3.2.7 CT\_F\_7:

Our product shall have a database table to store our Covid-19 closure information

#### 3.2.8 CT\_F\_8:

Our product shall contain a visual component that shows a heat map over a google map made up of the requested data-set

#### 3.2.9 CT\_F\_9:

The UMBC tracker shall have a button to switch the map into a case view at the top of the map.

#### 3.2.10 CT\_F\_10:

The UMBC tracker shall have a button to switch the map into a vaccine view at the top of the map. (contingent on vaccine data being present)

#### 3.2.11 CT\_F\_11:

The UMBC tracker shall have a button to switch the map into a state-closure view at the top of the map.

#### 3.2.12 CT\_F\_12:

Our product shall contain a drop down menu for the user to select a State

#### 3.2.13 CT\_F\_13:

Our product shall contain a drop down menu where once they have selected a state, they can select a county within that state

#### 3.2.14 CT\_F\_14:

Our product shall consist of a statistic GUI component that displays statistics on Covid case/vaccination/closure data depending on map-state. Will be displayed to the right of the map once an area is chosen.

#### 3.2.15 CT\_F\_15:

Our product shall contain a vaccine information button that displays the type of vaccines, links to vaccination areas, and safety info.

## 3.3 Configuration Requirements

**3.3.1 CT\_C\_1:**

The product shall test every 24 hours to see if a new csv file has been published by Broadstreet, and if it has, the data shall be automatically imported to the database.

## 3.4 Error Requirements

**3.4.1 CT\_E\_1:**

The UMBC Covid-19 Tracker shall check to make sure the user has entered a valid location in which the map represents. If not, the product will inform the user that their input is invalid and re-prompt

## 3.7 Qualification provisions

Requirements will be checked in the table below to notate what stage of development they are in:

1. Waiting for development
2. Internal team peer review check pass
3. Full Team Check pass
4. Deployment Testing pass

| **Requirement** | **Stage 1** | **Stage 2** | **Stage 3** | **Stage 4** |
| --- | --- | --- | --- | --- |
| *CT\_F\_1* | X |  |  |  |
| *CT\_F\_2* | X |  |  |  |
| *CT\_F\_3* | X |  |  |  |
| *CT\_F\_4* | X |  |  |  |
| *CT\_F\_5* | X |  |  |  |
| *CT\_F\_6* | X |  |  |  |
| *CT\_F\_7* | X |  |  |  |
| *CT\_F\_8* | X |  |  |  |
| *CT\_F\_9* | X |  |  |  |
| *CT\_F\_10* | X |  |  |  |
| *CT\_F\_11* | X |  |  |  |
| *CT\_F\_12* | X |  |  |  |
| *CT\_F\_13* | X |  |  |  |
| *CT\_F\_14* | X |  |  |  |
| *CT\_F\_15* | X |  |  |  |
| *CT\_C\_1* | X |  |  |  |
| *CT\_E\_1* | X |  |  |  |

## 3.7 Software system attributes

### 3.7.1 Reliability

Reliability of the application is defined as the probability that the software will fulfill the functions under a test environment imposed on the application by the software test designers. The test environment should consist of unit and component tests. Unit tests should test the functionality of queries and their display on the front end. Component tests should isolate specific functions on the front and back end, and test with a more rigorous and specific data set.

The application shall produce a reliability rating of at least 0.95, where the rating is defined as the number of test runs that completed successfully over the total number of test runs.

A failure is defined or recorded as the application failing to perform one of its functions and/or failing to meet the functional requirement.

### 3.7.2 Availability

Availability of the application is defined as the amount of time that the software is awake and able to perform its functions properly. Up time for this software should be 24/7, as this tracker will be used for real time data analysis. The functions of this software should be accessible to any number of users at any time.

**Notes**