**System Design Document**

**For**

**COVID-19 Tracking App**

Team Members:

● Sultan Abdalla Salim Saif Alteneiji

● Justin Andrews

● Bryce Cole

● Jonah Corbin

|  |  |
| --- | --- |
| Version | Date |
| V1 | 11/10/2020 |
|  |  |
|  |  |
|  |  |

[**1 Introduction**](#_oefunvmdjzst) **3**

[1.1 Purpose](#_mujlbix2au55) 3

[1.2 Objectives](#_nlrhhj4a784o) 3

[**2 Functional Scope**](#_ie4zhy7jnqpl) **3**

[**3 Overall Strategy and Approach**](#_5cocbmt6yrs6) **3**

[3.1 Testing Strategy](#_hkxsjccxk1g6) 3

[3.2 System Testing Entrance Criteria](#_97pd05mmj1xg) 3

[3.3 Testing Types](#_pdpfzb605mwf) 4

[3.3.1 Usability Testing](#_ndbjrwgitcsh) 4

[3.3.2 Functional Testing](#_cad4cs7mx1tb) 4

[3.4 Suspension Criteria and Resumption Requirements](#_wmiit9fvkykc) 4

[3.4.1 Suspension Criteria](#_d5byzbu7yydy) 4

[3.4.2 Resumption Requirements](#_d5byzbu7yydy) 4

[**4 Execution Plan**](#_d4ybtpohz153) **4**

[4.1 Execution Plan](#_u0629pef2onn) 4

[**5 Traceability Matrix & Defect Tracking**](#_45i4x59y2gyy) **5**

[5.1 Traceability Matrix](#_pmzx93gvle9q) 5

[5.2 Defect Severity Definitions](#_pmzx93gvle9q) 5

[**6 Environment**](#_vizmcpruj3xc) **5**

[6.1 Environment](#_es130irsnkzr) 5

[**7 Assumptions**](#_g8dsogg3ibjs) **5**

[**8 Risks and Contingencies**](#_g8dsogg3ibjs) **5**

[**9 Appendices**](#_g8dsogg3ibjs) **5**

# 

# 

# 1 Introduction

## 1.1 Purpose

The purpose of this document is to be a test plan for the COVID-19 Tracking project. This document will clearly describe the approach and methods the team will use to test the application and ensure that it meets the requirements set by the team and product owner.

## 1.2 Objectives

The objectives of this document are:

* To guarantee that the application created meets the requirements that were set by the team and product owner.

# 2 Functional Scope

This document will outline testing off all functionality and all files that are in the covid\_app/lib folder of the application. At a high level this will include the ability for the user to upload their data to the app, have that data be parsed inside the app, alter the weights of the ego network algorithm if the user chooses so, have the ego network algorithm analyze that data, and output an interactive result to the user. We will also be testing the intuitiveness and usability of the app UI. Examples of this include how easy it is to understand what each tab of the app does, can a new user of the app easily understand how to upload their data and then analyze it, can the user see the results from their most previous ego network calculation.

# 3 Overall Strategy and Approach

## 3.1 Testing Strategy

The testing of the COVID-19 Tracking project will include testing what is listed in the Functional Scope section above. The testing of this project will also include testing of functionalities that are added or modified, data validation, work flows.

## 3.2 System Testing Entrance Criteria

To begin testing there must be criteria to determine if testing is ready to begin, these criteria are:

* Functionality Completion: The functionality/functionalities being tested must be either completed or operational.
* Functionality Integration: The functionality/functionalities being tested must be fully integrated into the app.

## 3.3 Testing Types

### 3.3.1 Usability Testing

This testing includes the user interface intuitiveness, usability, presentation, and completion. Usability testing ensures that the user interface is easy for a user to operate and allows them access to all functionality and content within the application.

### 3.3.2 Functional Testing

This testing will ensure that all of the major and minor functionalities set by the team and product owner in other documents relating to this project are operational and working correctly.

## 3.4 Suspension Criteria and Resumption Requirements

This section contains the criteria to suspend and resume testing of the application.

### 3.4.1 Suspension Criteria

Testing of the application must be suspended if issues in the application prevent testing of other portions of the application. Once testing is halted the issues in the application must be fixed so that testing may resume. Based on what the issue was portions of the application may need to be re-tested.

### 3.4.2 Resumption Requirements

Testing may be resumed once the issue regarding the functionality that was being tested has been fixed and has been re-tested to ensure the issue and application were fixed.

# 4 Execution Plan

## 4.1 Execution Plan

The test cases to be executed for this project are:

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case Number | Test Scenario | Test Steps | Expected Result |
| T-1 | Upload Data | 1. Go to Upload Data tab 2. Click Upload Data button 3. Click on the data files to be analyzed 4. Click Submit | The user will be taken back to the Ego Network tab and will be shown an icon to indicate that their data was successfully linked |
| T-2 | Create Ego Network | 1. Data has been uploaded successfully 2. Go to Ego Network tab 3. Click calculate Ego Network | The user will be shown a categorized, interactive list of their friends based on the data that was analyzed |
| T-3 | Adjust Algorithm Weights | 1. Go to Ego Network tab 2. Click Adjust Weights button 3. Click the fields displayed to input chosen value as the weight | The user can successfully input a valid value into the field and then calculate an ego network using those altered weights |
| T-4 | App Intuitiveness | 1. App is given to a user that has no connection to the creation of this project 2. The user uses the app with no instruction of how to operate it | The user is able to select the data they wish to upload, calculate an ego network, and explore their results |
| T-5 | Reset Algorithm Weights | 1. Go to Ego Network tab 2. Click Adjust Weights button 3. Click Reset Weights button | The weights are reset back to their default values and an ego network can successfully be calculated using these reset values |

# 5 Traceability Matrix & Defect Tracking

## 5.1 Traceability Matrix

|  |  |
| --- | --- |
| Test Case | Corresponding Requirements |
| T-1 | F-2, F-3, F-4, F-5, F-8, F-12, D-1, D-2, D-3, S-1, S-3 |
| T-2 | F-1, F-2, F-3, F-4, F-5, F-7, F-9, F-12, S-2, S-3, QA-1 |
| T-3 | F-13 |
| T-4 | UH-1 |
| T-5 | F-13 |

## 5.2 Defect Severity Definitions

|  |  |
| --- | --- |
| Critical | The defect causes a loss of major, core functionality of the application. The functionality cannot be used or does not operate correctly. A defect of this magnitude will take significant effort to fix.  Examples of critical defects are:   * Data cannot be linked * Ego network results are not displayed * Ego network results are incorrect |
| Medium | This defect causes disruption to the user but the functionality of the application is still intact. A defect of this magnitude will take medium effort to fix.  Examples of medium defects are:   * Data cannot be linked on the first attempt * Algorithm weights cannot be adjusted |
| Low | This defect is typically cosmetic and causes little disruption to the user. A defect of this magnitude takes minimal effort to fix.  Examples of low defects are:   * Text fonts are incorrect * Fields do not align correctly |

# 6 Environment

## 6.1 Environment

The testing environment will be the virtual mobile device emulator used to develop the application.

# 7 Assumptions

There are no assumptions to list at this time.

# 8 Risks and Contingencies

There are no risks and contingencies to list at this time.

# 9 Appendices

Not applicable at this time.