

System and Software Support Plan (SSSP)

Field Progress Web App

Team 04

Uche Uba: Project Manager and Frontend Lead

Akanksha Diwedy: Operational Concept Engineer

Mayank Kulkarni: Requirements Engineer and Backend Lead

Madhavi Shantharam: Life Cycle Planner

Sahithi Vlema: Software Architect

Aisharya Joisa: Feasibility Analyst

Kevin Grimes: IIV & V and Quality Focal Point Engineer

08/12/2019

Version History

Date	Author	Version	Changes made	Rationale
08/12/19	SK	1.0	<ul style="list-style-type: none">• Turf cutting algorithm created, and visualization component included	<ul style="list-style-type: none">• Fulfilling requirements listed by the clients
			<ul style="list-style-type: none">•	<ul style="list-style-type: none">•
			<ul style="list-style-type: none">•	<ul style="list-style-type: none">•

Table of Contents

SYSTEM AND SOFTWARE SUPPORT PLAN (SSSP)	I
VERSION HISTORY	II
TABLE OF CONTENTS.....	III
TABLE OF TABLES	IV
1. Support Objectives and Assumptions	1
1.1 Support Objectives	1
1.2 Support	1
2. Support Strategy.....	2
2.1 Support Lifetime	2
2.2 Release Strategy	2
2.3 Release Requirement Determination.....	3
2.4 Release Process	3
3. Support Environment	4
3.1 Hardware	4
3.2 Software	4
3.3 Facilities	7
4. Support Responsibilities	8

Table of Tables

<i>Table 1: COTS products, and subsequent support documentation.....</i>	<i>2</i>
<i>Table 2.1: Description of python software and required support plan</i>	<i>8</i>
<i>Table 2.2: Description of React software and required support plan</i>	<i>5</i>
<i>Table 2.3: Description of MapBox software and required support plan.....</i>	<i>5</i>
<i>Table 2.4: Description of Deck GL software and required support plan</i>	<i>5</i>
<i>Table 2.5: Description of Google OR-tools software and required support plan</i>	<i>6</i>
<i>Table 2.6: Description of cucumber software and required support plan</i>	<i>6</i>
<i>Table 2.7: Description of selenium software and required support plan</i>	<i>6</i>
<i>Table 3: Stakeholders and their supporting responsibilities</i>	<i>8</i>

1. Support Objectives and Assumptions

1.1 Support Objectives

To provide appropriate support for the Field Progress Web App product, this document takes into account a couple of key driving objectives, namely;

- Ensuring that commercial off the shelf (COTS) products utilized during development are kept up to date, and provide resources on how to make appropriate patches to the source code, should some product version reach its end of life (EOL) and become inefficient for use
- Documentation of the requirements for running the application
- Ensuring the user understands the current limitation to the free tier provided by Googles OR-tool Vehicle Routing Problem approximation algorithm

1.2 Support

The following assumptions have been made during the creation of this document, and if any of them are invalid, they could make this support plan unworkable:

- There will be at least one maintainer for the product, with knowledge of python for the backend, and if required, knowledge of JavaScript and the React framework for the frontend
- For the turf cutting algorithm to work, a google access token linked to an account that can be charged if the number of geographical points given as an input exceeds 25. i.e. the number of voters that should be clustered within a precinct Hover, if the points are does not exceed 25, then the algorithm would work just fine as it is within the free tier limit

2. Support Strategy

2.1 Support Lifetime

The envisioned support lifetime of this product is approximately 3 years. The Field Progress Application is highly dependent on the following COTS products and will need to be dynamically updated based on various version updates that might arise from any of the products. Their support related information is provided alongside them

COTS Product/ Open Source Library	Support-related information
Python 3	Documentation: https://docs.python.org/3/ Support: https://www.python.org/about/help/
React JS	Documentation: https://reactjs.org/docs/getting-started.html Support: https://reactjs.org/community/support.html
Deck GL	Documentation: https://deck.gl/#/documentation/overview/introduction Support (github page): https://github.com/uber/deck.gl
Mapbox	Documentation: https://docs.mapbox.com/ Support: https://www.mapbox.com/contact/
Google OR-Tools	Documentation: https://developers.google.com/optimization Support: https://support.google.com/

Table 1: COTS products, and subsequent support documentation

2.2 Release Strategy

At the discretion of key stakeholders (clients) the project was not deployed to a production server, and as a result, there is no strategy for its release to users, as the aim of the product is to provide more insight into the domain of turf cutting algorithms. However, in order to ensure the product doesn't experience issues when the COTS products are upgraded, regular updates that coincide with major upgrades to COTS products should be performed. In essence, once the COTS products are upgraded and causes there to be errors in the compiling of the source code, appropriate patches should be made to the source code, and relevant tests specified in the test documentation performed thereafter.

2.3 Release Requirement Determination

The final product will serve as an internal tool to key stakeholders (Clients). The motivation for the product is to have an algorithm downstream, where they can send filtered voter information to, along with precinct information. They can then take the resulting output and plug it into any visualization software of their choice, be it the frontend application provided, or one currently being built in house. Once they are able to discern that the algorithm provides results to their satisfaction, and they are able to pilot it in certain campaigns, then they would get to decide how and when they would release their own product that would incorporate our algorithm as a tool. So, the release of a product utilizing our product as more or less a sub product would most likely occur after a budget is approved in order to test the result of the google OR-tools on data points exceeding 25 (free tier), testing is done to verify the result of the optimal routes created, and the product found ready for market.

2.4 Release Process

If the clients do decide to release a product, the release process would be at their discretion primarily. However, each new process should ideally fix the bugs reported from the previous versions, update any COTS product if required, and have the tests listed in the test suite performed before being deployed. While in production, any bugs reported should have tickets created, and then thereafter taken care of.

3. Support Environment

In order to run the product, there are certain environment requirements the user should take into account, ranging from browser support, to hardware and software. This section details what such requirements are, and lists documentation for the relevant software. For the browser support, since react is a JavaScript framework, it supports the same browser that ECMAScript 5, which are:

- Internet Explorer version 9 and above
- Chrome version 51 and above
- Firefox version 54 and above
- Safari version 10 and above
- Edge version 14 and above

3.1 Hardware

The product backend is a Django server, which is a python framework. The server has the should be run on a computer that meets the following minimum hardware requirements:

- Intel Core i3 processor
- 64 Mb Graphics Card
- 1 Gb of RAM
- Operating System: Windows 7 or later, macOS, and Linux

3.2 Software

The following software are required for the maintenance of the deliverable software:

- Python
- Google OR-tools
- React
- Mapbox
- Deck GL
- Cucumber
- Selenium
- Django

Software Requirement:	Python: v3.7
Rationale:	Python was selected for the development of the development of the algorithm and the backend, because it is a great tool for rapid prototyping, and has a lot of libraries that make data manipulation and visualization easy. There is also a huge community behind the open source product

User/Operator Manual:	Documentation: https://docs.python.org/3/
Availability Information:	The python organization, and open source community currently supports this product, and it will still be supported at the time of delivery
Note:	Python 3 was selected over Python 2, because google OR-tools as of July 2019 does not support Python 2.7. Also, a lot of libraries for the earlier version of python have become deprecated

Table 2.1: Description of python software required for the support plan

Software Requirement:	React: v16.11
Rationale:	React was selected for the development of the frontend of the application, because it integrates well with the other modules used for data visualization. Namely, mapbox and deck GL. As a matter of fact, deck GL which is an open source geographic visualization tool used in the product was built with react js in mind
User/Operator Manual:	Documentation: https://reactjs.org/docs/getting-started.html
Availability Information:	The Facebook, and the react open source community currently supports this product, and it will still be supported at the time of delivery
Note:	

Table 2.2: Description of React software required for the support plan

Software Requirement:	MapBox: v1.4.1
Rationale:	MapBox was selected for the development of the product because deck GL has a custom wrapper designed specifically for mapbox that enables it to utilize the power of the interactable map generated by the mapbox api, and also to be able to overlay dynamic layers on top of the map
User/Operator Manual:	Documentation: https://docs.mapbox.com/
Availability Information:	Mapbox currently supports the above version and provides free usage for monthly active users less than 20,000. The product however does require an access token for use
Note:	In order to use this product, a mapbox access token is required. Currently, the token being used belongs to a developer, and as a result may need to be changed at some point in the future

Table 2.3: Description of MapBox software required for the support plan

Software Requirement:	Deck GL: v7.3.3
Rationale:	Deck GL was selected for the development of the visualization component of the product, because it provides a rich visualization interface. It creates a layered approach to data visualization, and as a

	result allows complex visualizations to be constructed by composing existing layers
User/Operator Manual:	Documentation: https://deck.gl/#/documentation/overview/introduction
Availability Information:	Uber and the Deck GL open source community currently provide support for the version listed above. The product will be supported by the time of delivery of the software
Note:	

Table 2.4: Description of Deck GL software required for the support plan

Software Requirement:	Google OR-tools: v7.4
Rationale:	Google OR-tools was selected for the development of the algorithm, because it provides a suite of software for optimization problems. As a result, the algorithm is able to obtain an approximate solution for the Vehicle Routing Problem, a variation of the travelling salesman problem with more than one salesman.
User/Operator Manual:	Documentation: https://developers.google.com/optimization
Availability Information:	Google supports the version of its OR-tools suite listed above, and will continue to do so past the point of delivery of the software, as the version was just updated in July of 2019
Note:	Without a google access token that links to a google account that can be charged, if the input of VRP i.e. the number of voters to cluster within a precinct; if it exceeds 25 points, it throws an error. This is because it exceeds the input limit which a solution can be provided for free

Table 2.5: Description of Google OR-tools software required for the support plan

Software Requirement:	Cucumber: v6.0.5
Rationale:	Cucumber was selected for the testing of the product, because it is a tool that supports behavior-driven development. It's also provides the added benefit of allowing expected software behaviors to be specified in a logical language
User/Operator Manual:	Documentation: https://cucumber.io/docs
Availability Information:	Smart Bear and it's open source community currently supports the version listed above, and will do so by the time of delivery of the software
Note:	

Table 2.6: Description of cucumber software required for the support plan

Software Requirement:	Selenium: v4.0
-----------------------	----------------

Rationale:	Selenium was selected for the testing of this product, because it provides multi-browser support, has an ease of implementation, and is relatively easy to learn
User/Operator Manual:	Documentation: https://selenium.dev/documentation/en/
Availability Information:	Selenium has support from most of the browser vendors, and they are currently taking steps to make it a native part of the browser. So, the product will still be supported by the time of delivery
Note:	

Table 2.7: Description of selenium software required for the support plan

3.3 Facilities

This product does not require any special facility to maintain the deliverable software. This is mainly because the product was not deployed to any cloud service at the request of key stakeholders (clients). However, if down the line a decision is made to deploy the product and make it available to various users, it should be deployed on a cloud service that supports Apache servers, and can provide the ability to scale when required

4. Support Responsibilities

Table 1: Stakeholders and their supporting responsibilities

Stakeholder	Supporting roles	#	Supporting Skills
Nikolaj Baer	Software Maintenance	1	Proficiency with python and JavaScript
Nikolaj Baer	System Administrator	1	Proficiency with git and using the shell/terminal, and an understanding of using google API's
Evan Shulman	Operational and user support	1	Have an understanding of the product, and the user manual