

Proposal

CORONAPYRUS

Alexander Claman
Noah Jaccard
James McSweeney

Advisor: Dr. Vanessa Aguiar

Submitted in partial fulfillment
Of the requirements of CSC-431
Software Engineering course project

2/10/2021

Preface

This is a proposal for the Coronapyrus project for partial fulfillment of the requirements of a Software Engineering course (CSC431) project in the department of Computer Science at the University of Miami.

This proposal provides the scope and context of the project to be undertaken. It details the intended user group and the value that the system will have to them.

The intended audience of this document is the course professor and teaching assistants so that they can determine whether the project should be approved as proposed, approved with modifications, or not approved.

Table of Contents

1.0 Overview

pg 1-2

1.0 Overview

1.1. *Purpose, Scope and Objectives*

We will complete the proposed project in two parts. Firstly, we will construct a data-driven Python package called Coronapyrus. This package will have an open API designed to handle queries about COVID-19 data; it will also use web scraping and governmental statistics to provide region-specific information upon request. We will then use this package to deliver add-ons for social media applications, specifically targeting Discord and Slack. These extensions will ensure automated responses to user requests for COVID-19 information. A computing device with an internet connection and the relevant application installed will be necessary to use our proposed add-ons or package.

We will use Python for the entirety of the development process. We will make use of common data science packages like Pandas, Numpy, Scipy, Seaborn, and Matplotlib for data analysis and visualization. We will also incorporate web scraping tools like Selenium and the built-in requests module to collect recent data to be returned by the external API. To construct the add-ons, the preexisting discord.py and slack_bolt packages will be used for a Discord bot and a Slack app, respectively.

The creation of this package and these add-ons will meet the ongoing need for the dissemination of easily understandable, current COVID-19 data. Not only is the spread of misinformation common, especially in social spaces with little fact checking, but regional differences in COVID-19 severity, news, and data can often lead to discrepancies in opinion. Being able to quickly request and show local COVID-related news and public data within the context of a social application will combat the above issues. Further, add-ons beyond just those for Discord and Slack are easily made possible by the creation of an API, allowing our provided package to be extensible for further developer use.

1.2. *Project description*

Coronapyrus (“the Package”) will have the following features:

- web-scraping features to extract data from Internet sources (Selenium, requests)
 - will provide either official COVID data or news/media
- tools to create visualizations from acquired data (Matplotlib, Seaborn)
- variable scopes for information requests
 - geographically scoped (nation-wide or local)
 - temporally scoped (more or less recent)
 - scoped by source (CDC or other official data, news or media sources)
- an external API that makes all calls to internal functions necessary and returns data organized by the above scopes per a user’s request

Each add-on (“the Applications”) will have the following features, created by utilizing pre-existing platform-specific packages or by writing new code:

- a user-callable “help” command
 - will provide in-depth information about the capabilities of the Application and how to use it (discord.py)
- the ability to make API calls to the Package for COVID-19 information as needed
 - calls fall within the same limits as the scopes provided by the Package
- various user-callable functions to provide COVID-19 information on request
 - visual information – data visualized in graphs (Matplotlib)
 - raw information – numerical data presented in an easy-to-read, tabular format (Pandas, discord.py)
 - news or media – linked, with a blurb for at-a-glance information (discord.py)
- a notification system
 - the Application will send pings with new data at user-defined intervals (discord.py)

Possible complications or further requirements for development include:

- the need to address serving many requests with the Package
 - may need to implement parallel processing techniques such as threading with built-in Python packages to handle increased load as needed
- the need to address serving many different Discord or Slack instances with a single server-side managing instance of code for each Application
 - may need to implement parallel processing/threading as above
 - may also need to create a SQL database to handle notification timing/activity (PyMySQL)