Covid Vaccine Appointment Scheduler
Programmer’s Documentation

Joshua Fawcett, Max Hopkins, Meghan Riehl, Yuyao Zhuge

April 26, 2021

This document should help facilitate programmers in making additional modifications and improvements to the scheduling and searching capabilities in the Covid Vaccine Appointment Scheduler. We will give an overview of:

1. System Files
2. Procedures
3. User Interaction and Interface
4. Possible Improvements

The appointment scheduler allows users to search for available appointments with the following parameters:

- Vaccine type (Moderna, Pfizer, Janssen (Johnson & Johnson), or any)
- Distance (the distance the clinic is from the user’s input address)
- Day/time (weekday/weekend and am/pm, or any)

After a successful search (an available appointment is found) the system will display a list of available appointments and their details for the user to select and reserve. The scheduler will then display a confirmation number and mark the appointment as reserved in the database. Users will also be able to cancel their appointment by providing the confirmation number, they will then need to search for a new appointment if desired.

System Files:
LIST MAIN FILES HERE - What they’re for/do

flask_main.py: This is the main file of the system. It is in charge of the system website and the main interactions between the different components of the system.

Database.py: This file is in charge of managing the database.txt file. It provides an interface for the other modules to be able to add appointments, delete appointments, and update appointment information. This file implements a class called database which stores the database.txt filename and a list of each line in the file. It has methods called “self.addApp”, “self.deleteApp”, and “self.updateApp”.

search.py: This file accesses the database to search for appointments that meet the user’s search criteria. The search.py file imports from database.py and api.py in order to implement a function that takes a vaccine type, a maximum distance from user to clinic, a date, and the user location and outputs a list of all of the appointments that met the specified criteria.

app_scheduling.py: This file is in charge of generating a unique confirmation number and writing that confirmation number into the database. The file imports from database.py and search.py in order to implement a function called “schedule” that takes a string (in the format of an appointment) and a database object as parameter and returns the confirmation number generated.

Api.py: This file uses the google maps api in order to find the distance from the user to a specific clinic. Api.py imports from “requests” and “smtplib” in order to implement a function called “getDistance” which takes the user’s address and the clinic’s address as parameters and outputs the distance between the two.

Procedures:

1. Flask.py imports database.py, search.py, and app_scheduling,py to create an interaction between the “frontend” and “backend”. This means that flask.py does all of the following:
    a. Takes the user input
    b. Redirects the user input to the searching module
    c. Receives output (a list of appointments) from searching module and displays it to the user
    d. Receives a scheduling request from the user and redirects this request to the scheduling module
    e. Receives the confirmation number from the scheduling module and displays it to the user
2. Search.py imports from database.py to get access to each line in the database.txt file in order to find appointments that match the search criteria. Search.py also imports from api.py in order to use the getDistance function to find the distance between the user and a specific clinic.
3. App_scheduling.py imports from database.py in order to use the database.updateApp method to write the confirmation number into the database.txt file.

User Interaction and Interface:

1. The user interface is a simple website page that allows the user to input text into several text boxes to enter their personal information, and lets them select search criteria from drop down lists.
2. After the user hits the search button, they are redirected to a search result page which displays a list of all the appointments that met their search criteria.
3. From the search result page they can schedule any of the appointments by clicking “schedule” which will take them to the confirmation page which displays the confirmation number to the user.
4. At the very bottom of the main website page there is a button that says “cancel appointment” which when clicked, takes the user to the cancel appointment page.
5. On the cancel appointment page the user can input their confirmation number to cancel their appointment

Known Possible Improvements:

- Format times to look better
    - Currently the times are just a number but we could format the times to look more like a reasonable person would expect (1:00 pm as opposed to 1)
- Allow searching for specific days
    - Currently the user is only allowed to search via weekday/weekend. Though the available appointments displayed include a specific day it is possible to add a search parameter to only include certain days or a range of days
- Email confirmation sent to user
    - Currently the only confirmation the user gets is a displayed number that they must write down to remember or use to cancel their reserved appointment but emailing capabilities could be added to allow the system to send an email to the one provided to the user with the confirmation number and appointment details
- Clinic contact information
    - There is not other information beyond the clinic name and address attached to the appointment details. The clinic contact information could be added to the database to be displayed along with the appointment information.