

Group 3 Project Plan

Author: Meghan Riehl

Overview:

Our project is to create a vaccination search and scheduling system. This will be achieved by breaking down the entire system into 4 more manageable parts.

1. The Database will be built by Josh
2. The searching module will be built by Meghan
3. The scheduling module will be built by Yuyao
4. The user interface, in the form of a website, will be built by Max

Integration between modules will be handled by those whose modules need to communicate. Thus when the searching module needs to give the user interface the results that will be handled by Meghan and Max.

Our team lead is Josh though as a small team most decisions are made collectively and smaller implementation details will be left to the individuals directly involved with their portions of the project.

Communication:

Our team will meet at least twice a week:

- Tuesdays @ 12:00 PM
- Thursdays @ 3:00 PM
- Extra meetings to be scheduled as needed

The team will meet via zoom so that we can talk face to face and collaborate on implementations and problem solving. Day to day communication and general updates will be handled through a discord server where team members can talk via instant messaging and voice chat channels as needed.

The individual modules and codes will be kept on a GitHub repository where all team members can access all parts. Any and all edits will be recorded by GitHub at the time of the edit, including adding and removing items from the repository.

Tuesday and Thursday scheduled meetings will have a mandatory attendance while extra meetings will not though all members who can are expected to join. Team members will report on the completion of their assigned tasks either before or during a scheduled meeting so that other team members can confirm and double check their work.

Build Plan:

Basic breakdown:

1. Database
2. Website
3. Searching
4. Scheduling

We will start with the database as everything else is dependent on that information. The other three modules can be built roughly around the same time and can have pseudo code stand-ins for basic implementation lines (i.e. `#import database here`). Searching will need to know how the search parameters from the website will be specifically passed to it but the basic algorithm can be written in the meantime. As is the case with the user selected appointment and the scheduling module. Overall everything builds on what comes before it though is not wholly dependent on it.

*See the final page for a detailed breakdown of individual tasks and deadlines.

Risks:

- Expected
 - Lack of knowledge
 - Poor communication
 - Delayed assigned tasks
 - Time restraints
 - Unavailability
- Unexpected
 - Emergencies
 - Hardware malfunctions
 - Other unforeseen problems

The expected risks can be mitigated with proper, prompt, communication. Unexpected risks, while can be planned for in part, are unexpected for a reason. In the case of a team member who is unable to complete their assigned task(s) they must report to the team lead (Josh) as soon as possible with the problem, any other coinciding issues, and, if possible, suggestions for a solution.

Team members are expected to ask for help when needed and complete their assigned tasks to the best of their ability.

When possible follow up responses are expected to confirm receipt of any communication when not in a face to face meeting.

Vaccine Scheduling System Software Requirements Specification

Authors: Joshua Fawcett and Max Hopkins

Table of Contents

- 1. SRS Revision History**
- 2. The Concept of Operations (ConOps)**
 - a. Current System or Situation**
 - b. Justification for a New System**
 - c. Operational Features of the Proposed System**
 - d. User Classes**
 - e. Modes of Operation**
 - f. Operational Scenarios**
- 3. Specific Requirements**
 - a. External Interfaces**
 - b. Functions**
 - c. Usability Requirements**
 - d. Performance Requirements**
 - e. Software System Attributes**

SRS Revision History

Date	Author	Description
4-10-2021	MH/JF	Created initial document
4-26-2021	MH/JF	Revised

The Concept of Operations

The proposed vaccine distribution system will provide an interface which will give users the ability to search for and schedule Covid-19 vaccination appointments based on specific search criteria such as the physical distance from the user to the vaccination clinic, the desired type of Covid-19 vaccine (moderna, pfizer, etc.), and the time of the day that the vaccine is offered. The interface will be available through a website and will use data from vaccine providers such as health clinics to provide the users with accurate and real-time information on the available appointments.

Current System or Situation

The current coronavirus pandemic is ravishing the world, with economies slipping and lives being lost. However, vaccines have been developed that help combat the symptoms that are caused by the coronavirus. Distributing these vaccines to the general public is of utmost importance to combatting the pandemic as a whole. Current systems in place check whether an individual is eligible, and then present appointment times to the individual to schedule an appointment. They then receive a confirmation number in order to streamline their arrival at the specified location.

Justification for a New System

Our new system will attempt to build on the current vaccine scheduling system employed by adding three extra search requirements: vaccine type, times that the patient is available, and distance from the patient's address. If an individual wishes to schedule an appointment with a clinic that has a certain vaccine, they should certainly be able to do so. We will also provide functionality for an individual to cancel or reschedule an appointment if they need to. This will open up appointments for others to fill up slots as much as possible.

Operational Features of the Proposed System

This system is designed to solve the problem of scheduling a vaccine appointment to stop the spread of the novel coronavirus. The system must be simple and quick to use in order to ensure users are able to complete scheduling appointments in a timely manner, without being confused about how to schedule an appointment.

1. Scheduling begins with the user entering some personal information such as their name, address, email address, city, state, and zip code.
2. The user will then search for a clinic based on three search parameters: type of vaccine they wish to receive, the maximum distance from their current location, and their own personal availability.
3. Once the search is complete, the user will have a variety of clinics to choose from based on their search requirements, and will be able to schedule an appointment based on what works best for them.
4. Once the user has chosen an appointment, they will receive their appointment confirmation number, and their appointment slot that they chose will be updated to reserved.
5. The user will also be able to cancel or reschedule appointments up to 24 hours before their appointment time.

User Classes

There are two distinct user classes for the proposed vaccination distribution system.

1. The **individual class** is made up of any individuals that are looking to schedule an appointment for a Covid-19 vaccination. Generally, the individual class will contain adults who are seeking either their first or second dose of the vaccine.
2. The **administration class** consists of the development team that is responsible for creating and maintaining the system.

Modes of Operation

The proposed vaccination distribution system has two modes of operation, one for the users and one for the administrators.

1. The **User Mode** of operation will be made for the individual user class and will provide access to the search and scheduling systems. Individuals will be able to access this mode of operation via the system website in order to schedule for vaccination appointments.
2. The **Administration Mode** of operation is the mode that the administration user class will use in order to provide various services to the system, such as maintenance and adding/removing vaccine appointments from the system database. The administration will access this mode via the system code and database file in order to make any necessary changes to the system.

Operational Scenarios

Use Case: Scheduling an appointment

Brief description: This use case describes the process in which an individual would use this vaccination distribution system to schedule for an appointment to receive the Covid-19 vaccine.

Actors: An individual seeking Covid-19 vaccination

Preconditions:

1. The individual has access to the internet and can navigate to the system's website
2. The user is prepared to enter personal information such as their name, email address, and location.

Steps to Complete the Task:

1. The individual navigates to the website in their search browser
2. The individual uses the search feature to find appointments
3. The system displays a list of appointments that match the user's search criteria
4. The user selects an appointment
5. After reviewing the information, the user reserves the appointment by clicking the "schedule this appointment" button

Postconditions:

- The system displays an appointment confirmation number to the user and sends an email to the user's email address containing the confirmation number.

Use Case: Canceling an appointment

Brief description: This use case describes how an individual would cancel an appointment that they had previously scheduled using this system.

Actors: An individual user of the system

Preconditions:

1. The individual has access to the internet and can navigate to the system's website
2. The user has a confirmation code from previously scheduling with the system

Steps to Complete the Task:

1. The individual navigates to the website in their search browser
2. The user navigates to cancel appointment page
3. The user enters their confirmation number and selects cancel
4. The system database is updated to reflect this change

Postconditions:

- The user is no longer scheduled for the appointment

Use Case: Updating available appointments

Brief description: This use case describes the process in which the system administration would go about updating the available appointments in the system.

Actors: System Administration

Preconditions:

1. There are new appointments that need to be added to the database or there are old appointments that need to be removed

Steps to Complete the Task:

1. The administrator updates the database by manually adding/deleting entries from the database file

Postconditions:

- The database is up to date

Specific Requirements

The specific requirements for the system will be split up into four categories: must have, should have, could have, and won't have.

Must have:

1. Schedule based on vaccine type
2. Schedule based on location
3. Schedule based on patient availability
4. Search for a clinic based on the search requirements
5. Output confirmation number after scheduling appointment
6. Update databases after scheduling an appointment

Should have:

1. Simple website layout with a specific field for the user to enter each of the following:
Name, email address, physical address, city, state, zip code, and search parameters

2. Minimal steps from searching to scheduling to streamline process
3. Email appointment details to user
4. Cancellation of appointments previously scheduled by a user
5. Rescheduling of an appointment

Could have:

1. User can also schedule for dependents
2. Automatic appointment suggestion for second dose if required
3. Email reminders about first appointment/scheduling second appointment
4. Collect data on common symptoms
5. Symptom reporting/collecting
6. Directions to clinic/link website

External Interfaces

User Inputs:

1. Name: Search input
2. Purpose: System will take the search requirements and determine the best appointment choices based on the parameters.
3. Source of input: User looking to schedule an appointment will input the search requirements
4. Valid ranges: Inputs will be in the form of a dropdown table with these parameters:
 - a. Vaccine type: Moderna, Pfizer, J&J, AstroZenica
 - b. Location: less than 10 miles, less than 15 miles, less than 20 miles, less than 25 miles, less than 30 miles, greater than 30 miles.
 - c. Availability: Calendar where user selects dates/times that are available for them
5. Units of measure: Vaccine type, miles, days/times
6. Data formats: String format from dropdown table

1. Name: User personal information
2. Purpose:
 - a. To calculate the user's distance from each of the clinics
 - b. To send a confirmation email to the user
3. Source of input: User will fill out a user profile
4. Valid ranges:
 - a. Name: Any string of characters
 - b. Address: Any string of characters
 - c. Email: Any string of characters
5. Data formats: stored in strings/database

System outputs:

1. Name: Search results
2. Purpose: The system will output appointment times/locations based on user requirements from "search input"

3. Output destination: Output will display on screen after a user searches. The user will be able to choose an appointment.
4. Valid ranges: Should not output appointments outside the range of user requirements, or appointments that are already scheduled.
5. Data formats: Pulled from database to be displayed on user's screen

1. Name: Confirmation number
2. Purpose: To differentiate between appointments in the database
3. Destination of output: Will be displayed to the user after selecting an appointment time
4. Valid outputs: Will be pulled from the clinic database in order to ensure each confirmation number is unique.
5. Data formats: displayed as a string on user's screen

1. Name: Email output
2. Purpose: To ensure the user has a reminder/papertrail for their appointment
3. Destination of output: the user's email pulled from their user profile
4. Valid ranges: Should only email confirmation number, date, time, and location
5. Data formats: string sent to a user's email

Functions

1. Inputs for searching will be based on a selection from a drop-down list, so error checking will not be necessary here.
2. The default input in the list will be "Any" (any vaccine, any location, any time).
 - a. These search parameters will be used in the searching algorithm to find appointments that fit the search requirements.
 - b. The algorithm will consider time slots first, then location, and vaccine type last. (For example, a location closer to the patient with a better time slot will be presented before an appointment with the right vaccine type but is farther away and in an unavailable time slot)
 - c. We will also check to make sure that the confirmation number will be unique
3. There is no error handling for the case in which a user enters invalid information such as their email address, name, or physical address.
 - a. An invalid email address entered by the user will cause the system to not send a confirmation number via email. However, the system will still display the confirmation number to the user.
 - b. An invalid name entered by the user has no effect on the system as it is not used by any of the components other than the sending of the confirmation email.
 - c. The system does not check for an invalid address as there is no way to tell what the user intended to enter
4. Any inputs into the system will be entered either as the user's personal information, or as search criteria.

- a. The search criteria input will yield a list of appointments that meet the criteria as output. The conversion from search criteria to a list of appointments happens via the search module which uses an algorithm to scan the database for appointments that match the criteria.
- b. Personal information input is used to generate output in the form of a confirmation email which reminds the user of their confirmation number. The conversion from personal information to confirmation email happens in the scheduling system.

Usability Requirements

The usability requirements for the proposed vaccine distribution system are split into two categories: The user interface and the functionality of the system.

User Interface:

1. The most important requirement for the user interface is that it needs to be easy to navigate. The user interface needs to have an intuitive layout to give the users the ability to navigate the website with as little confusion as possible.
2. If users are unable to find the account creation page or the search page, then they may be discouraged from using this vaccination distribution system.
3. The user interface must also be visually appealing to the users.

Functionality:

1. The usability requirements for the system functions revolve around the speed and responsiveness of the system. This means that each of the different functions of the system, such as the search or schedule functions, need to complete as quickly as possible.

Performance Requirements

- 100% of all input from the user should be processed and stored in the database in less than 1 second.
- Searching for clinic appointments shall not take more than 10 seconds.
- Creating a unique confirmation code shall not take more than 1 second.

Software System Attributes

The most important attributes for the proposed system are accuracy and operability.

1. **Accuracy:** The system needs accurate results when it comes to features such as searching or scheduling.
 - a. The search system needs to provide appointments that meet the specified criteria in order to function acceptably. If the search system returns appointments that

are already reserved or do not meet the specified criteria, then the user will not be able to schedule for their appointment

- b. The scheduling system needs to update the correct field in the database in order to make sure that the wrong appointment does not get reserved.
2. **Operability:** The system must provide an interface to the user that is easy to operate and navigate (Van Vliet, 2008, pp.128). If the user is not able to perform any operations on the system then they won't be able to schedule any appointments and the system will have no use.

References

Van Vliet, Hans. (2008). Software Engineering: Principles and Practice, 3rd edition, John Wiley & Sons

Vaccine Scheduling System

Software Design Specification

Yuyao Zhuge, April 12th, 2021 - v 1.0

Table of Contents

1. SDS Revision History	1
2. System Overview	2
3. Software Architecture	2
4. Software Modules	5
4.1. Searching Module	5
4.2. Scheduling Module	6
4.3. Database Module	8
5. Dynamic Models of Operational Scenarios	10
6. References	11
7. Acknowledgements	11

1. SDS Revision History

Date	Author	Description
4-11-2021	Yuyao	Created the initial document.
4-12-2021	Yuyao	Finished the first SDS draft.
4-13-2021	Yuyao	Finished recreating diagrams, minor changes in words.

2. System Overview

2.1. Services of System

- This system aims to help people find a Covid-19 vaccination appointment that meets all of their needs when it comes to the vaccine type, distance to the vaccination clinic, and day/time of the appointment

2.2. Functionalities of System

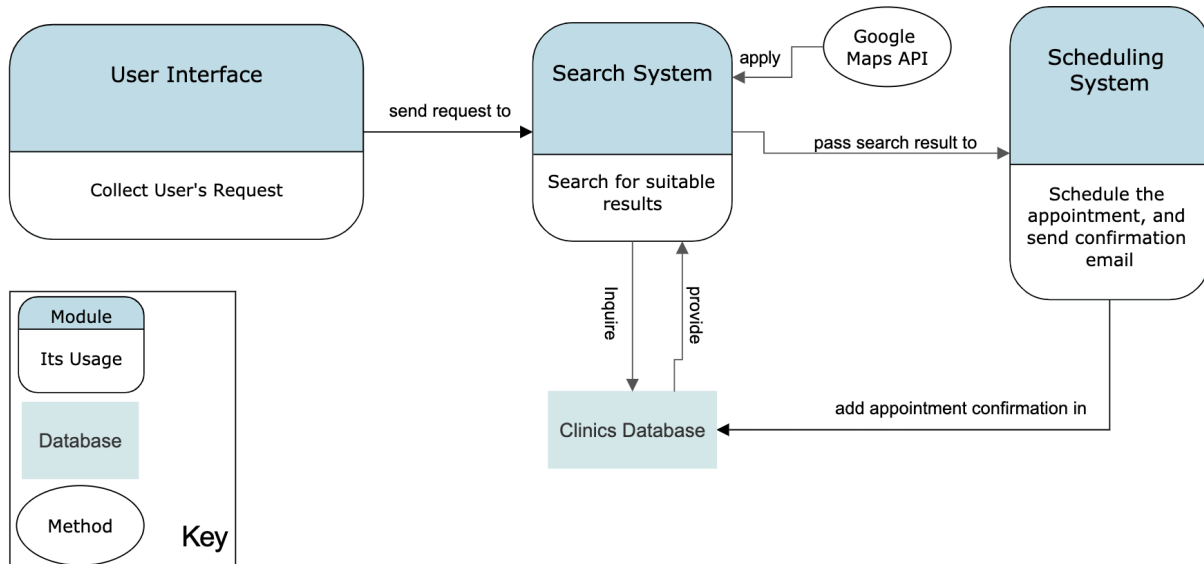
- In terms of the users' choices of vaccination types, preferred times, and distance to the clinics, our search system will list any appointments that meet these criteria. Then, the users can pick one to schedule an appointment. Our schedule system will reserve the appointment in the database and display a confirmation number to the user.

3. Software Architecture

3.1. The set of components.

- User Interface
- Search System
- Scheduling System
- Clinics Database

Software Architecture Diagram



3.2. The functionality provided by (or assigned to) each component.

- User Interface/Website

Vaccine Scheduling System

This is the official vaccine scheduling site. To schedule, fill out your home address and your email address. We will use these to determine your location relative to other clinics, and to contact you to message you your confirmation code. Once completed, we can use the information you provided to help with your search for a vaccine appointment. Please select the vaccine type, distance requirements, and your available times to search for an appointment. Thank you, and stay safe!

Personal Info:

Address: City: State:

ZIP code: Email address:

Search Parameters:

Vaccine Type: Maximum Distance:

Preferred Times:

☐ Weekday/AM

☐ Weekday/PM

☐ Weekend/AM

☐ Weekend/PM

- The user interface interacts with the user to collect the info needed for the searching system. The information collected consists of the search criteria that is needed for the searching system to work. This is shown in the picture above.

- The user interface will show the search results for all of the available appointments that matched the search criteria. The user can select one of the appointments from the given list to schedule.
- The user interface shows the confirmation page that displays a confirmation number for the scheduled appointment.
- Appointment Searching Module
 - The search system inquires data from the Clinics Database, and then according to the user's input, it analyzes the data to exclude unmatched appointments.
 - For calculating the distance between the user's address and the clinic's address, we apply Google Maps API.
 - After completing, the search module will return a list of available appointments that matched the search criteria to the User Interface.
- Appointment Scheduling Module
 - After the user picks one of the matched appropriate appointments, the scheduling system will generate a confirmation number for this appointment and save it to the database.
 - Finally, it will send the user a confirmation email with specific info for this scheduled appointment.
- Database
 - This database saves and organizes preloaded clinics data which are necessary for searching and scheduling purposes.
 - It saves the confirmation number when the appointment is scheduled.

3.3. Which modules interact with which other modules.

- Searching Module
- Scheduling Module
- Database
- User Interface
- The searching module invokes the clinics database, collects the user's input from User Interface and returns a satisfying result list on User Interface.
- User Interface displays the result; passes the chosen appointment result to the Scheduling Module.

- The Scheduling Module generates and then saves confirmation numbers into the Database; show confirmation page through User Interface.

3.4. The rationale for the architectural design.

We chose this design because it allowed us to split the project in several parts so that each team member can have clear jobs. The modules are designed to link with one another while keeping the whole structure not very complicated.

Each of the components were chosen by determining the main things we would need in order to create a vaccine scheduling system:

- A database to store all of the available appointments
- A user interface to allow the user to interact with the system
- A searching system to give the user ability to find appointments that work for them
- A scheduling system to reserve an appointment that the user chooses.

4. Software Modules

4.1. Searching Module

The module's role and primary function.

This module's role is to search for available clinics for users. It invokes clinics' data from the database, provides them to Google Maps using API integration, and then shows the result on User Interface.

Interface Specification

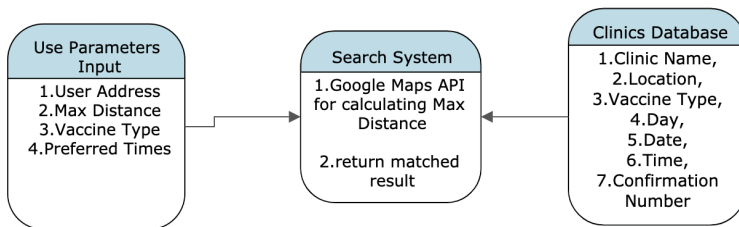
This program will interact with User Interface and the database. It invokes the Clinics Database with attributes:

- 1) str(Clinic name)
- 2) str(location)
- 3) str(vaccine type)
- 4) str(time)

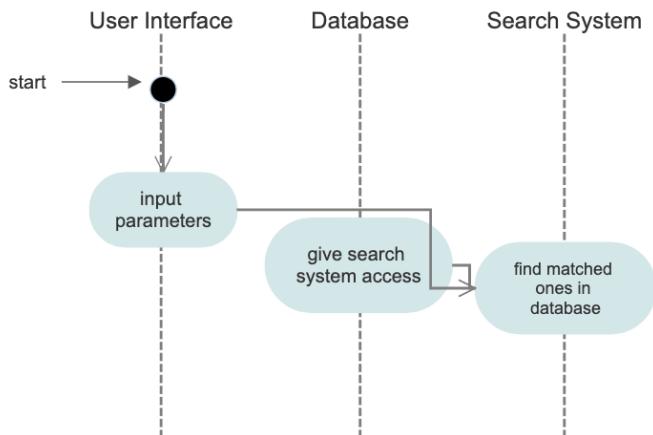
- 5) str(day)
- 6) str(date)
- 7) int(confirmation number)

A Static and Dynamic Model

Searching System UML Object Diagram



Search System UML Sequence Diagram



Design rationale

This module is designed to achieve a searching ability for convenience to find the locations of the clinics. Finding clinics is the main function of this program. We will apply Google Maps API as a powerful assistance and interacting tools with the users.

4.2 Scheduling Module

The module's role and primary function.

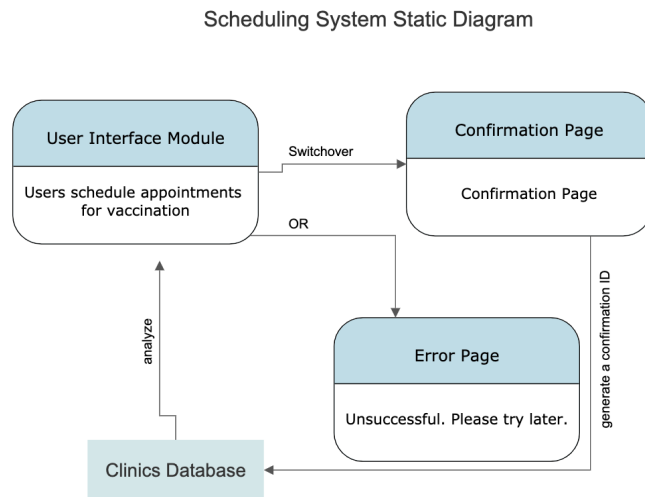
This program aims to schedule an appointment with clinics for the users. It invokes clinics' data and the users' data from the database. It gets feedback from the users about which clinics they choose, and a specified time for vaccination. Then, it tries to match these inputs with the clinics' available_time. If it matches, then it will generate a confirmation receipt.

Interface Specification

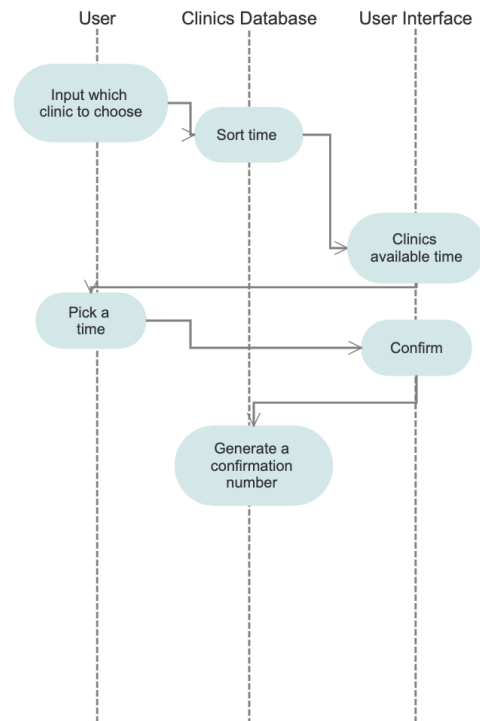
This program will interact with the front-end interface and the database. It invokes the Clinic Class from the database with attributes:

- 1) Location,
- 2) Vaccine Type,
- 3) Day,
- 4) Date,
- 5) Time,
- 6) Confirmation Number,
- 7) Clinics Name

A Static and Dynamic Model



Scheduling System UML Sequence Diagram



Design rationale

This module is designed to achieve a scheduling ability for the users to schedule an appointment with clinics. It achieves one of the most important operations to simulate making appointments (or make connections with the clinics server). As mentioned, we will apply Google Maps API as a powerful assistance and interacting tools with the users.

4.3 Database Module

Interface Specification

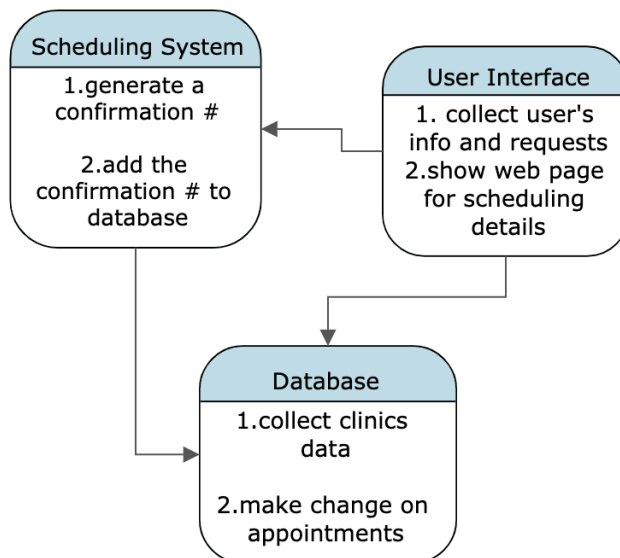
This module uses python class structure to specify and organize all kinds of data. It includes classes of Clinics with attributes:

- 8) Clinic Name,
- 9) Location,

- 10)Vaccine Type,
- 11)Day,
- 12)Date,
- 13)Time,
- 14)Confirmation Number

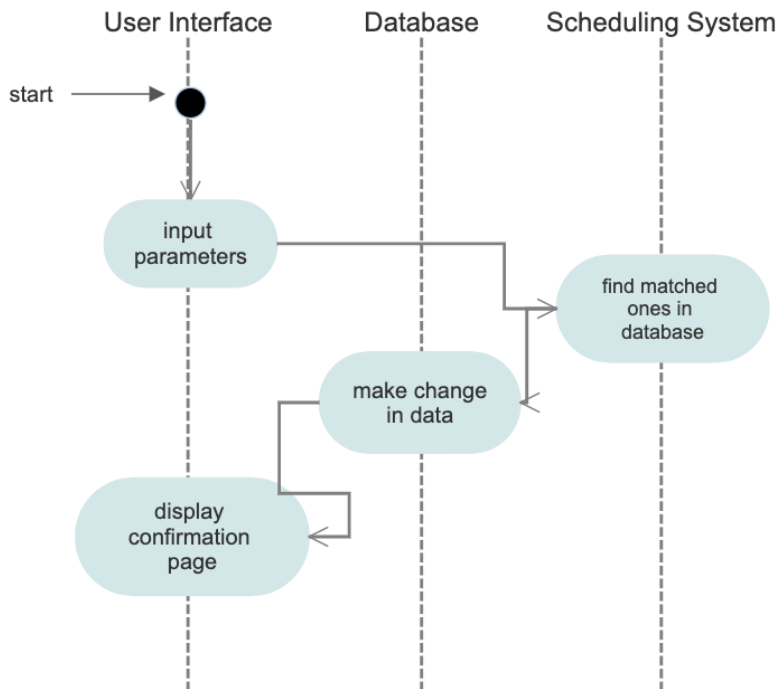
Static Model

Database UML Object Diagram



Dynamic Model

Search System UML Sequence Diagram



Design rationale

The database module is an essential and indispensable part of our software. It appears in any achievable functions of our website. If needed, we will add more attributes to these classes.

Alternative designs

1. Considering there are risks that the security module is not implemented adequately, we could switch user login mode to immediate scheduling mode, which indicates that the users don't need to provide personal info but personal email address for receiving confirmation email. In this case, we will not need to create the User Login Class or People Class in our database but a user's preferred name and email.
2. Currently, our project is aiming to create a website that can be used practically to schedule an appointment with clinics, which it requires us to contact with clinics

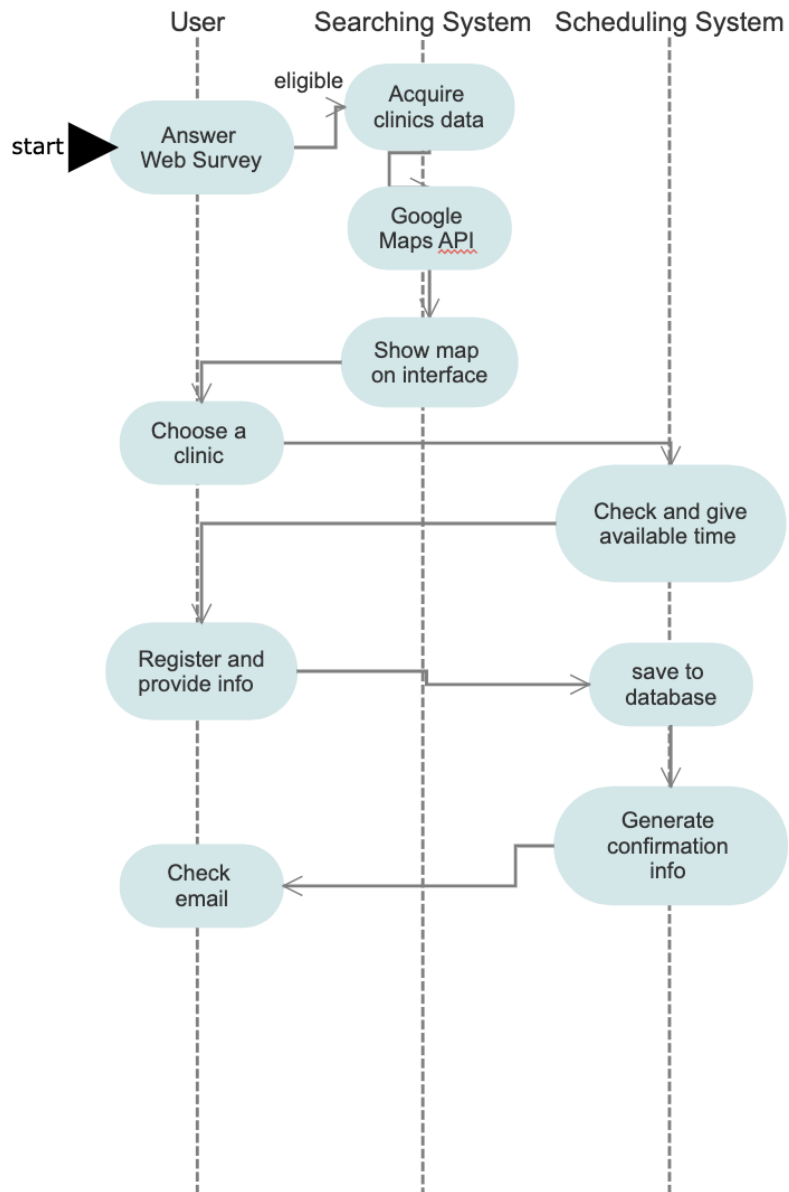
for authorization (This seems most likely). If so, we will create a simulator and mimic the clinics' website's functionality for the users to "schedule an appointment".

3. By the time we have the website working, all people in the US might be eligible to receive the COVID-19 vaccine. In this case, we would get rid of the web survey part, and directly lead users to the next Searching Module.

5. Dynamic Models of Operational Scenarios

This project involves two major functionalities as mentioned: searching and scheduling. The following is the UML sequence diagram.

UML Sequence Diagram



6. References

Sileika, R. (2014). Pro Python System Administration. United States: Apress.

7. Acknowledgements

This template was modeled after Anthony Hornof's base Software Requirement Specification (SRS) given to the UO class CIS 422 in Spring 2021.

This template builds slightly on a similar document produced by Stuart Faulk in 2017, and heavily on the publications cited within the document, such as IEEE Std 1016-2009.