PROJECT INFORMATION

Project Title

(Try to choose a catchy title. Max 20 words).

Examining COVID-19 Health Services Use in UCI Health

Team Information

Team Name: **Team # on Canvas:**	**UCI Health** **15**
Team member 1 (Team Lead) Tang, Ryan; 38859342; tangrs1@uci.edu	
Team member 2 Sajo, Suraj; 89963984; sajos@uci.edu	

Team member 3
Zhou, Haining; 95309476; haininz@uci.edu

INTRODUCTION

Motivation/Background

(Describe the problem you want to solve and why it is important. Max 300 words).

There is a need for an improved understanding of treatment effectiveness for patients based on their age and overall health. We need to utilize data from the UC Data Health Warehouse to examine medication utilization among patients with high-risk comorbidities diagnosed with COVID-19 and correlate with health services utilization to determine which medications work most effectively based on patient characteristics. This is an extremely important task that would help doctors and other medical professionals better treat their patients and prevent reactions to the medicines.

State of the Art / Innovation

(Describe how the problem is solved today (if it is), and the innovation/advance of your project. Max 200 words).

With the COVID-19 pandemic raging on for the past year, the need for good data extrapolation is necessary to see how patients react to certain treatments. Currently, there are no great solutions to this problem because of how new and current the data is. The difference between outpatient and hospital is a glaring issue especially when health experts are trying to understand the current state of patients in the hospitals.

Project Goals

(Describe the project general goals. Max 200 words).

Our solution has two main phases, constructing the database and extracting analytics from the cleaned data. The first phase will be us grabbing the relevant data from the UCI Health data warehouse which will then be processed by our scripts. We expect to use Python, SQL, and R to make and add the necessary infrastructure for the database. Some of the milestones we expect for this first milestone are to get access to the data warehouse (there is a little bit of a delay in getting our credentials), to understand what data is relevant to the team, and how we can extract the correct data points with the raw data set.

The second phase of the solution will be creating insights for the data. After organizing and putting the data into a more accessible database, we will then need to understand the trends in the data. We expect to use Python or R to make the data extractions and use Python to create some possible visualizations to show our insights. Some of the milestones would be making sure which data points are relevant to us, building out the visualizations, and vetting the data so that it reflects the research correctly.

Assumptions

(Describe the assumptions (if any) you are making to solve the problem. Max 180 words).

We are assuming that this end product will be used by medical professionals.
We are assuming that the data is organized differently from different UC campuses.
We are assuming all data is accurate.
We are assuming data is updated on a weekly basis.

SYSTEM ARCHITECTURE OVERVIEW

High Level Diagram

(Provide an overview of the entire proposed system, identifying the main components that would be developed/used for the product and their interfaces. Max 500 words or 1 page).

The main component that will be developed is a MySQL database that will be used by the UCI Health Department to run statistics in order to see which medications impact the COVID-19 disease. The data for this database will be extracted from a UC dataset that will be processed by our Python script and then be inserted into the new MySQL database that we created.

User Interaction and Design

(Provide an overview of the main interfaces and their design for users' interaction with the proposed product. Max 500 words or 1 page).

Our main interface will be a database system (very likely a relational database system). We are planning to choose among MySQL, SQL Server, and Oracle because these are the 3 most common open-source relational databases, but we need to further discuss with the project sponsor about their considerations on factors like storage, flexibility, and security. For now, we want to mainly focus on the design of the database. For instance, we will make the name of each column (attribute) more brief and intuitive, including setting up an appropriate primary key for each table. We also want to create more relational forms and use foreign keys wisely (meaning not to put a lot of things in a single table) to make things clear. In order to enable users to write queries and get the results much faster, we are planning to also add indexes to appropriate tables. These are all general plans now. We will come up with more specific and targeted designs when we really get to see the CORDS data.

REQUIREMENTS

User Stories and Use Cases

(Describe functional and non-functional requirements of your project by describing user stories and use cases. Please add at least a total of 10 entries (stories or/and cases). Max 1,000 words or 2 pages).

Use Cases

Use Case #1

Use Case: Updating UCI Health Database
Actors: UCI Health Database, CORDS
Precondition: UCI Health Database has access to CORDS data for all UCs
Flow of Events:
Basic Path:
1. UCI Health Database automatically logins to CORDS
2. UCI Health Database filters out and check data that isn't currently in our system
3. UCI Health Database pulls new data from CORDS
4. UCI Health Database organize data
5. …

Alternative Paths:
CORDS has no new data to provide for the UCI Health Database, logs us out of their system
Postcondition: UCI Health Database has been updated and ready for medical professionals to use

Use Case #2

Use Case: Cleaning the UCI Health Database
Actors: UCI Health Database, Script
Precondition: data received from CORDS hasn't been inputted in database
Flow of Events:
Basic Path:
1. UCI Health Database calls our script
2. script sorts out and deletes unnecessary information in each data
3. …

Alternative Paths:
UCI Health Database runs into an error while cleaning up the database, sends developers the error
UCI Health Database logs out of CORDS
Postcondition: new information has been properly cleaned out

Use Case #3

Use Case: Search Patients with Certain Constraints
Actors: Medical Professional, UCI Health Database
Precondition: Medical Professional has logged on to UCI Health Database and select "search patients" option
Flow of Events:
Basic Path:
1. Medical Professional specifies constraints for patients they want to filter
2. UCI Health Database sorts out data through specifications given by Medical Professional
3. UCI Health Database displays data based on specifications
4. ...

Alternative Paths:
If constraints leads to database filtering being empty, database will inform Medical Professional that it is invalid constraint and ask to choose another one
Medical Professional picks constraints that are valid
Postcondition: Medical Professional analyzes the people with the constraints they want and goes on from there

Use Case #4

Use Case: Medical Professional updates database with own patients
Actors: Medical Professional, UCI Health Database
Precondition: Medical Professional is certified and has access to input data
Flow of Events:
Basic Path:
1. Medical Professional clicks “add new patient” selection
2. Medical Professional inputs name, date of birth, preconditions, etc.
3. Medical Professional finalize and click enter to enter data
4. ...

Alternative Paths:
If medical professional inputs name with special characters, database will ask them to restart
database already has patient inside system
Postcondition: UCI Health Database is updated with new patient

User Stories

User Story #1
As a filter, I can filter given data, so that medical professionals can use it to study
- Filter Data: Given constraints, when saved on a API, then data gets filtered for the preferring constraints(test = medical professional wants patients range 40-50, data displays all patients)

User Story #2
As a accessor, I can access CORDS, so I can use in UCI Health Database to update data
- CORDS accessor: Given a username and password, when accessor logins to CORDS, then script depicts which data is new(test = on weekly basis, assessor automatically logins and sees data for new patients)

User Story #3
As a database response, I can retrieve new data, so I can use in UCI Health Database to update data
- Database Response: Given a request to pull data, when new data is found in CORDS, then databases pulls data and organize and stores in UCI Health Database(test = on weekly basis, UCI Health Database receives new data to input in storage)

User Story #4
As a data, I can request data in a fashionable manner, so users can have access to it as soon as possible
- Data Request: Given a request to get data, when weekly submission for CORDS comes in, then script tells accessors to go ahead and go get data for new update(test = on weekly basis, UCI Health Database automatically tells script to go fetch data)

User Story #5
As a displayer, I can display data for user, so user can analyze and make decision with given information
- Data Displayer: Given medical professional, when medical professional asks for data, then displayer displays data they need in an organized manner(test = medical professional wants people who had diabetes before they got Covid, data displays all patients)

User Story #6
As a organizer, I can organize data, so user doesn’t have to read unnecessary information
- Data Organizer: Given new data, when new data is pulled from CORDS, then data organizer cleans unnecessary information from each data(test = database just receive data, goes an cleans it up)

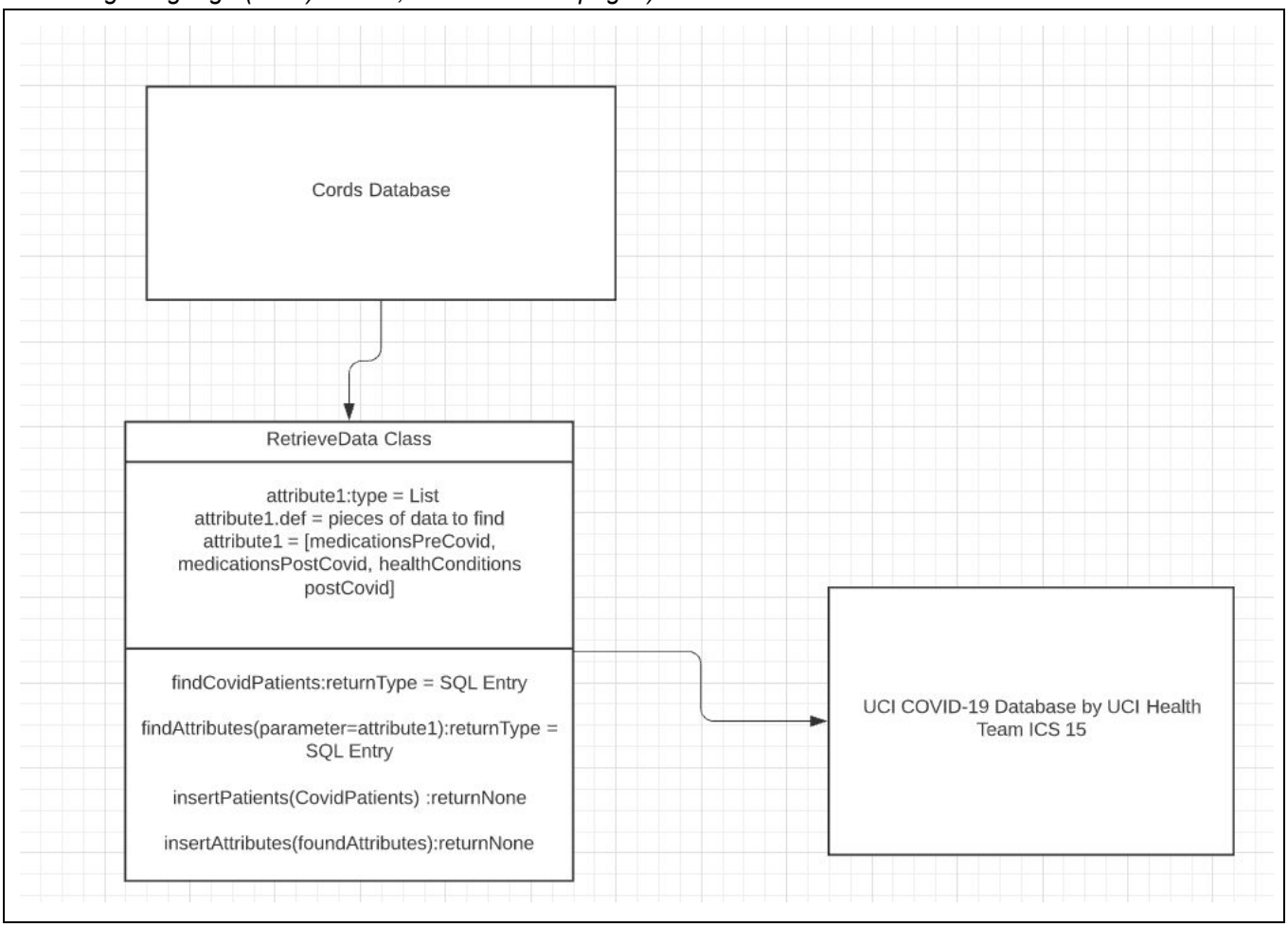
Coding and Testing

(Include links to your code (GitHub commits/issues) and describe testing and testing metrics for your project. Max 500 words or 1 page).

This is the link to our project github page: https://github.com/haininz/CS-190-Capstone-Project.git
Since we are creating a database rather than constructing a new software, there might not be a lot of things to test. Our testing will mainly focus on the logical correctness and the efficiency of the database. We will first make sure each row and column is the same as what we expected. We could do that easily by writing queries for each column and see if there will be an error (like wrong entry name). Another important thing we want to make sure is the response time. Clearly, we do not want the user to write a query and then wait for half an hour to get the results. In other words, we will be testing on the design of our database. We will do that by writing down some queries first, and then run each one against one of the potential designs of the database and compare the performance for each. Right now we do not know how large the database will be, so unfortunately we are not able to give an ideal response time.

SYSTEM MODELS

(Provide a visual design of the solution by showing contexts, sequences and behavioral states using The Unified Modeling Language (UML). Max 1,000 words or 2 pages).



APPENDICES

(List/describe the platforms and technologies you plan to use (or currently using) to develop the solution). Max 200 words).

- Python
- Pycharm
- SQL
- MySQL
- R
- CORDS Datahouse
- Github
- Trello