

# Personal Quarantine Management (Health-focused)

---

## Phase III

### **Designed and Implemented by**

Feras Alshehri  
Luis Orozco  
Saurabh Kularni

### **Advised and Guided by**

Dr. Renne Renner

This report documents the team project for **CINS 370** - Fall of 2020.

# Table of contents

---

<b>0. TEAM MEMBERS &amp; MEMBERS RESPONSIBILITIES</b>	<b>2</b>
Team Members	2
Members Responsibilities	2
<b>1. PROBLEM STATEMENT &amp; REQUIREMENTS ANALYSIS</b>	<b>3</b>
Problem Statement	3
Requirements Analysis	3
<b>Big Picture</b>	<b>3</b>
<b>Data Users</b>	<b>3</b>
<b>General Features &amp; Functionality</b>	<b>3</b>
<b>2. CONCEPTUAL MODEL &amp; LOGICAL MAPPING</b>	<b>5</b>
ER Model	5
Relational Database Schema	5
<b>3. DATABASE CREATION</b>	<b>8</b>
Create Script (DDL Script)	8
Test Data Injection Script (DML Script)	8
Clear Script	8
<b>4. DATA ACQUISITION</b>	<b>9</b>
Personal Data	9
Public Data Consumption	9
Sample Data	10
<b>5. USER INTERFACE</b>	<b>12</b>
Use Case	12
<b>User use cases</b>	<b>12</b>
<b>Data use cases</b>	<b>12</b>
Wireframes	14
<b>User Registration and Login</b>	<b>14</b>
<b>Track body temperature, heartbeat rate, oxygen level. Track Test Results.</b>	<b>16</b>
<b>Track weight, height and age.</b>	<b>17</b>
<b>Track Exposure (Location History)</b>	<b>18</b>
<b>Track symptoms, visualize test cases and hospitals</b>	<b>19</b>

## **0. TEAM MEMBERS & MEMBERS RESPONSIBILITIES**

### **0.1. Team Members**

- Saurabh K. (Team leader)
- Luis O. (Assist Lead)
- Feras A.

### **0.2. Members Responsibilities**

- Problem statement: Saurabh, Luis, and Feras
- Requirement analysis: Luis
- ER model: Saurabh, Feras, Luis
- Relational schema: Feras
- DDL, DML, and dropAll SQL scripts: Feras
- Data plan: Saurabh
- Use cases: Feras
- Meta Data: Luis

# 1. PROBLEM STATEMENT & REQUIREMENTS ANALYSIS

## 1.1. Problem Statement

For someone under self-quarantine (e.g. due to a spread of an infectious disease), this project aims to provide a tool to give individuals information that could help them better track their own health, and assess the conditions in a specific region to take precautions as appropriate with regards to the self-quarantine cause.

### Our Primary goals are:

1. Provide a way for individuals to monitor their own symptoms and metrics in a personal profile.
2. Provide a way for individuals to make an informed decision to set their alert level in accord to the surrounding conditions based on public data.
3. Provide a way for individuals to track and manage their exposure to infectious disease

## 1.2. Requirements Analysis

### 1.2.1. Big Picture

This database brings data and analysis to one location throughout communities for individuals. A person will be distinguished using a Person\_ID to help each individual to track and manage their exposure to infectious disease. All tests will be tracked locally within hotspots using their corresponding Person\_ID. The initiative is created to help communities in self quarantine keeping communities safer and aware by providing additional resources such as Hospitals with details on their building capacity, for those individuals who may need additional support. These resources are an important component in response to this self quarantine exertion.

### 1.2.2. Data Users

As we expect people exercising self quarantine as a response to infectious disease, and or other health issues, this database can be a value to our individual users. These users are considered before we build outside of our scope. The ADMIN will be able to manage and update the entire database, while our main users will be able to view, manage and update their own data only.

### 1.2.3. General Features & Functionality

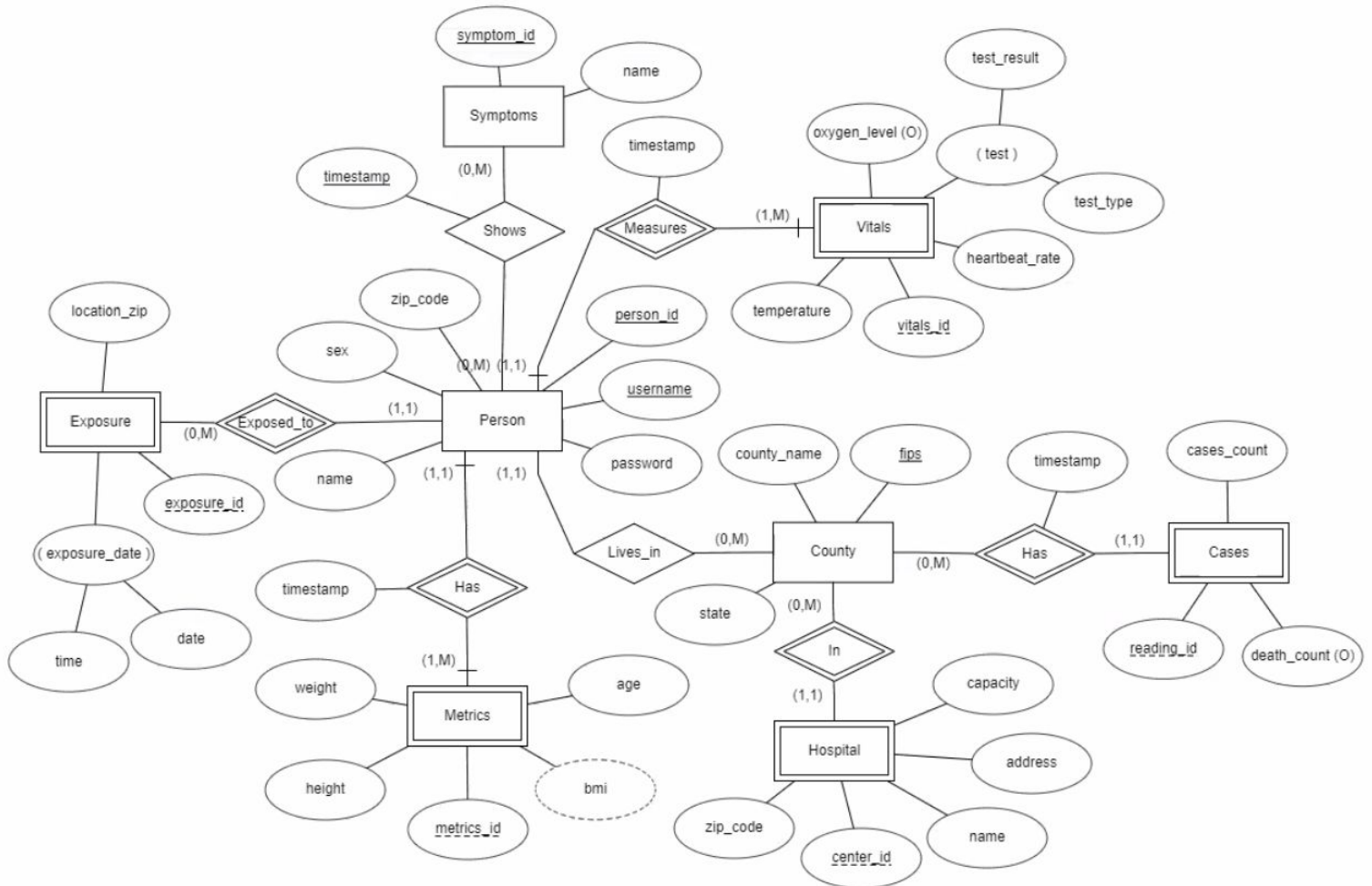
Overriding features for the database will include a breakdown of a single individual in the quarantine process that consists of Person, Metrics, Vitals, Hospital, Exposure, and Symptoms.

This type of data will provide a visual of how fast any infectious diseases are spreading and which areas users should avoid if the number of cases are too high for that specific zip code.

This database has been assembled for individuals exercising self quarantine and provides awareness of quality public health. Users will be able to upload data regularly, and self monitor for symptoms such as: fever or chills, cough, loss of taste or smell, sore throat, nausea or vomiting. If a person gets exposed to an infectious disease, they will be able to see the date, location zip code and will have an Exposure ID to identify what disease it is. Each person will be able to monitor their vitals giving results of temperature, heart rate, oxygen level, test results for those who have been tested. This database can also be used as a resource to find local hospitals within the zip code area, name of the hospital, and the capacity of the building. Each hospital will be given a Center ID.

## 2. CONCEPTUAL MODEL & LOGICAL MAPPING

### 2.1. ER Model



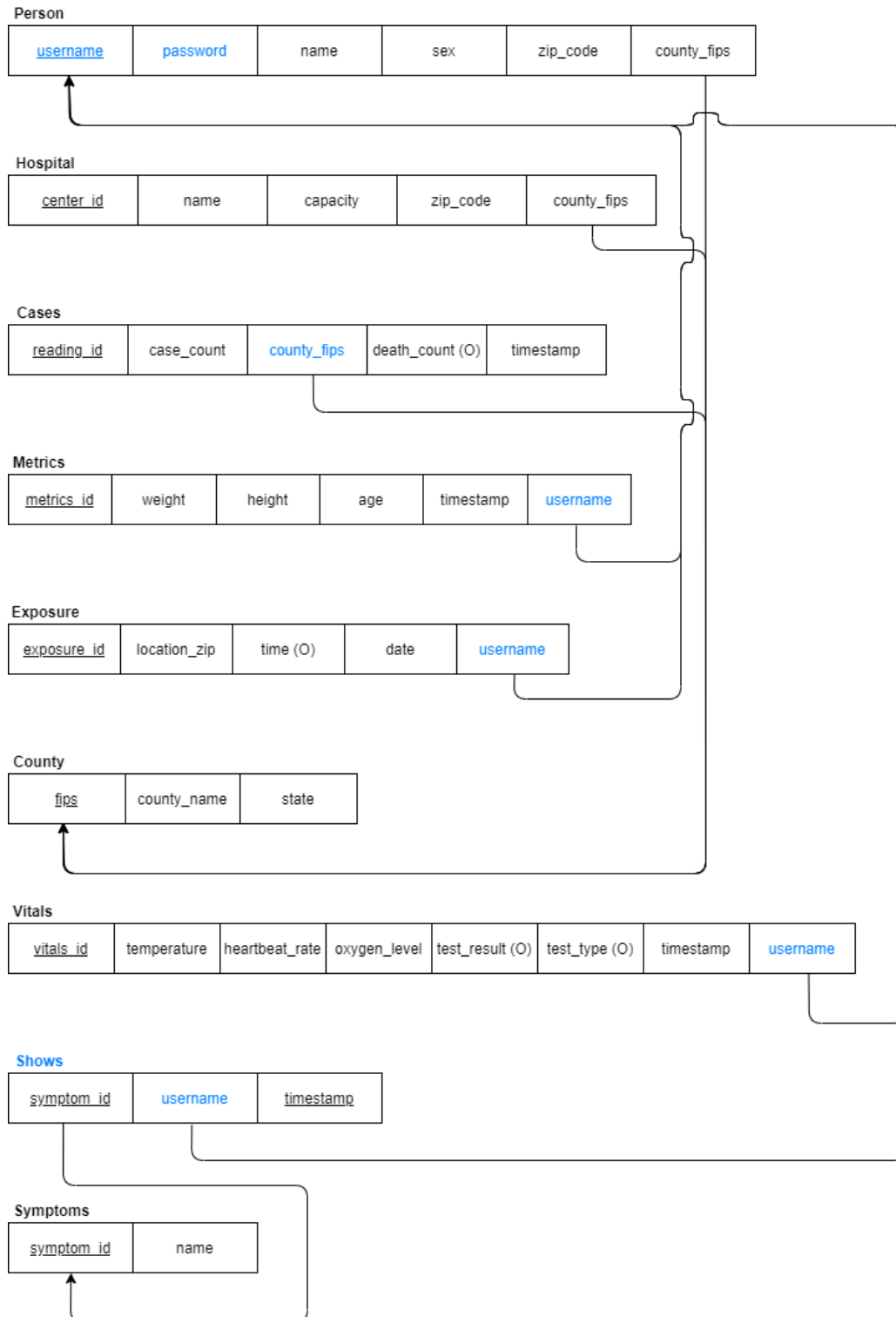
Changes from Phase 2 -

We moved the Hospital and Cases entity as a child of Person to as a child of County as per your feedback.

## 2.2. Relational Database Schema

### 2.2.1. Relational schema diagram

Changes from the previous revision (phase 2) are highlighted in blue.



### 2.2.2. Meta-Data

Attribute	Entity	Data Type	Null-able?	Description
fips	County	INT	No	The fip code representing a county
state	County	ENUM	No	State name, constrained to the 50 state names fully spelled out
county_name	County	VARCHAR(100)	No	Name of county
username	Person	VARCHAR(100)	No	Username
password	Person	VARCHAR(255)	No	Password
Name	Person	VARCHAR(100)	No	Person name
sex	Person	ENUM	No	Gender F = female M = male
zip_code	Person	INT	No	Person corresponding zip
county_fips	Person	INT	No	Primary key for Person
weight	Metrics	FLOAT	No	Person weight (in Kilograms)
height	Metrics	FLOAT	No	Person Height (in centimeters)
metrics_id	Metrics	INT(Auto increment)	No	Primary Key for Metrics
age	Metrics	INT	No	Person age in whole years
timestamp	Metrics	timestamp	No	Current time for when person checks vitals
username	Metrics	VARCHAR(100)	No	Username of this row's owner
temperature	Vitals	float	No	Person temperature in celsius degrees
heartbeat_rate	Vitals	INT	Yes	Beats in per minute
oxygen_level	Vitals	INT	Yes	SpO2 level, percentage out of 100
vitals_id	Vitals	INT(Auto increment)	No	Primary Key for Vitals



test_type	Vitals	varchar(30)	Yes	Medical test type
test_result	Vitals	ENUM	Yes	Positive or Negative medical test result
timestamp	Vitals	timestamp	No	Current time for when person checks vitals
username	Vitals	VARCHAR(100)	No	Username of this row's owner
center_id	Hospital	INT(Auto increment)	No	Primary Key for Hospital
name	Hospital	VARCHAR(100)	No	Name of hospital
capacity	Hospital	INT	No	Building capacity information
zip_code	Hospital	INT	No	Zip code of hospital
county_fips	Hospital	INT	No	Fip of hospital
exposure_id	Exposure	INT(Auto increment)	No	Primary Key for Exposure
location_zip	Exposure	INT	No	Zip code of exposure location
time	Exposure	TIME	Yes	Time of exposure event
date	Exposure	DATE	No	Date of exposure event
username	Exposure	VARCHAR(100)	No	Username associated with exposure event
symptom_id	Symptoms	INT(Auto increment)	No	Primary Key for symptoms list of symptoms
name	Symptoms	VARCHAR(30)	No	Name of symptom
timestamp	Shows	TIMESTAMP	No	Timestamp of the symptom being present
symptom_id	Shows	INT	No	The present symptom's id
username	Shows	VARCHAR(100)	No	The person showing the symptom's username
reading_id	Cases	INT(Auto increment)	No	Primary key for Cases
cases_count	Cases	INT	No	number of cases

death_count	Cases	INT	Yes	number on how many deaths
count_fips	Cases	INT	No	The county fip code of the cases' entry

### 3. DATABASE CREATION

#### 3.1. Create Script (DDL Script)

Refer to the file *G6\_PHASE3\_DDL.sql* for the DDL script to create the database schema. Read the comment for more details.

#### 3.2. Test Data Injection Script (DML Script)

Refer to the file *G6\_PHASE3\_DML\_insert.sql* for the DDL script to populate the tables with test data. Read the comment for more details.

#### 3.3. Clear Script

Refer to the file *G6\_PHASE3\_DropTables.sql* for a script that can clear the selected tables in a database. Read the comment for more details.

## 4. DATA ACQUISITION

### 4.1. Personal Data

- 4.1.1. The user must register using username, age, zip code, county and sex, height as well as weight. This will set up the initial profile for the user.
- 4.1.2. In addition to the above details, after logging into the application, the user will need to provide details about their vitals like heartbeat rate, user's body temperature, blood oxygen level, name of the latest health test and result of the test. If the user has travelled during the quarantine period, they will need to provide zip-code of the location, date and time of the visit. All the data related to the user will be taken by manual user input.
- 4.1.3. External data like Symptoms will be updated by the admin manually.

### 4.2. Public Data Consumption

- 4.2.1. Public data includes a list of hospitals based on the zip-code of the user. This data is gathered from the following source and is freely available to download as well as through API endpoint. This database is updated weekly. The source for this data has been updated from Phase - 1 to a different source in order to support a list of hospitals according to zip-code and capacity.

Name of the link - Definitive Healthcare: USA Hospital Beds | COVID-19 Resources

<https://coronavirus-resources.esri.com/datasets/definitivehc::definitive-healthcare-usa-hospital-beds>

Data for the list of counties is available through kaggle.com and is extracted from John Hopkins University. The link to the same is as follows.

Name of the link - COVID-19 US County JHU Data & Demographics

[https://www.kaggle.com/headsortails/covid19-us-county-jhu-data-demographics?select=us\\_county.csv](https://www.kaggle.com/headsortails/covid19-us-county-jhu-data-demographics?select=us_county.csv)

Data related to cases of the ongoing infectious disease is collected from verified github.com profile of New York Times (<https://github.com/nytimes>). This data is updated daily and is available freely through github repository. The link for the same is as follows.

Name of the link - covid-19-data/us-counties.csv at master · nytimes/covid-19-data

<https://github.com/nytimes/covid-19-data/blob/master/live/us-counties.csv>

## 4.3. Sample Data

### 4.3.1. Person

person_id	name	sex	zip_code
1	Saurabh Kulkarni	M	95926
2	Feras Alshehri	M	95926
3	Luis Orozco	M	95926

### 4.3.2. Metrics

metrics_id	height	weight	age	timestamp	person_id
1	155	60	25	11/15/2020 12:50:05	1
2	160	80	30	11/16/2020 2:50:05	2
3	170	90	35	11/17/2020 3:50:05	3

### 4.3.3. Vitals

vitals_id	temperature	heartbeat_rate	oxygen_level	test_type	test_result	timestamp	person_id
1	36	150	97	Covid-19	0	11/16/2020 2:50:05	3
2	37	150	97	Covid-19	0	11/16/2020 2:50:05	2
3	35	150	97	Covid-19	1	11/16/2020 2:50:05	1

### 4.3.4. Hospital

center_id	name	capacity	zip_code
1	Enloe Medical Center	298	95926
2	Oroville Hospital	153	95966
3	Orchard Hospital (FKA Biggs - Gridley Memorial Hospital)	24	95948

#### 4.3.5. Exposure

exposure_id	location_zip	time	date	person_id
1	95926	3:10	2020-11-14	1
2	95926	14:50	2020-11-15	1
3	95966	19:10	2020-11-15	2

#### 4.3.6. Symptoms

symptom_id	name
1	Dry Cough
2	Sore Throat
3	Headache

#### 4.3.7. Shows

timestamp	person_id	symptom_id
11/17/2020 3:50:05	1	1
11/17/2020 3:50:07	1	3
11/17/2020 3:50:30	2	1

#### 4.3.8. Cases

reading_id	county_fips	cases_count	death_count
1	06007	3399	56
2	06019	33809	460
3	06031	9372	87

#### 4.3.9. County

fips	county_name
6007	Butte County
6009	Calaveras County
6011	Colusa County

## 5. USER INTERFACE

### 5.1. Use Case

#### 5.1.1. User use cases

##### 5.1.1.1. Admin updating the list of supported symptoms in Symptoms table

5.1.1.1.1. An admin can update the list of supported symptoms by adding/removing/altering an entry into the symptoms table, to change the list of supported symptoms by the application and the user interface.

##### 5.1.1.2. An individual monitoring their personal health

5.1.1.2.1. A user can create a personal account to build a health profile, and monitor their health and track changes using historical data sets added by the user.

#### 5.1.2. Data use cases

##### 5.1.2.1. Track vitals

5.1.2.1.1. Record body temperature measurement at a given date and time.

5.1.2.1.2. Record heartbeat rate (during a doctor visit, or using an equipped smartphone or smartwatch at home) a given date and time.

5.1.2.1.3. Record blood oxygen level (during a doctor visit, or using an equipped smartphone or smartwatch at home) a given date and time.

5.1.2.1.4. This will allow the user(s) to track their vitals and view the changes in their body over time in a list format.

##### 5.1.2.2. Track weight

5.1.2.2.1. A person tracking changes in their own weight.

##### 5.1.2.3. Track height

5.1.2.3.1. A person, most-likely younger than 18 to 20 years old, could track changes in their height over time.

##### 5.1.2.4. Track age

5.1.2.4.1. A person tracking changes in age to better assess the public data with regards to their respective age group.

##### 5.1.2.5. Track travel locations

5.1.2.5.1. A person tracking geo-locations overtime to assess the possibility of exposure/contraction.

- 5.1.2.6. Track relative symptoms
  - 5.1.2.6.1. Record relative symptoms (such as caught, sore throat, loss of sense of smell... etc) a given date and time.
  - 5.1.2.6.2. This will allow user(s) to track symptoms and when they first were noticed over time.
- 5.1.2.7. Track test results
  - 5.1.2.7.1. Record test results over time.
  - 5.1.2.7.2. This will allow the user(s) to better understand contraction location and/or inform a group of possible spread caused by the user (visiting the workplace or school before testing positive).
- 5.1.2.8. Visualize regional cases statistics
  - 5.1.2.8.1. Fetch public cases data of the local region and visualize it for the user.
  - 5.1.2.8.2. For example, assuming the reason for the self-quarantine is COVID-19, user(s) may choose to visualize the number of cases and/or deaths in the user's county or zip-code.
- 5.1.2.9. Visualize regional hospitals statistics
  - 5.1.2.9.1. Fetch public data of the local region hospitals and visualize it for the user.
  - 5.1.2.9.2. For example, assuming the reason for the self-quarantine is seasonal flu, the user(s) may choose to visualize the capacity of a hospital in the user's county or zip-code.
- 5.1.2.10. Track Exposure (Location History)
  - 5.1.2.10.1. An individual can track the exposure history to see if there is any contact with a high case density zip-code area.
  - 5.1.2.10.2. This will allow the user(s) to be cautious and get tested from the near-by hospitals if possible.
- 5.1.2.11. Update Hospital Capacity
  - 5.1.2.11.1. Admin can update the list of hospitals and the capacities.
  - 5.1.2.11.2. This will allow the system to keep updated with the remote source(s).
- 5.1.2.12. Update regional cases statistics
  - 5.1.2.12.1. Admin will update the data required to visualize the regional cases.
  - 5.1.2.12.2. This will allow the system to keep updated with the remote source(s).



## 5.2. Wireframes

### 5.2.1. User Registration and Login

The wireframe shows a web browser window titled "Personal Quarantine" with the URL "http://lucidchart.com". The main heading is "Personal Quarantine Registration". The form contains the following fields and controls:

- Username:** A text input field.
- Height (cm):** A numeric input field with the value "150" and up/down arrow controls.
- Age (whole years):** A numeric input field with the value "25" and up/down arrow controls.
- Password:** A text input field.
- Weight (kg):** A numeric input field with the value "60" and up/down arrow controls.
- Zip-code:** A numeric input field with the value "123456" and up/down arrow controls.
- County:** A dropdown menu with "Butte" selected and up/down arrow controls.
- Sex:** A dropdown menu with "Female" selected and up/down arrow controls.
- Register:** A blue button with the text "Register".

Personal Quarantine

← → ↻ ⬆

personalq.com

Personal Quarantine Login

Username

Password

LOGIN

### 5.2.2. Track body temperature, heartbeat rate, oxygen level. Track Test Results.

Personal Quarantine

personalq.com

Cases by county

Nearby Hospitals

Vitals

Metrics

Exposure

Symptoms

1

Temperature (C)

36.5

2

Heart rate

100

3

Oxygen Level (%)

95

9

Name of the test

100

9

Was the test result positive?

NO

Update


View History

### 5.2.3. Track weight, height and age.

Personal Quarantine

personalq.com

Cases by county



Nearby Hospitals

Hospital 1, Zip-code - 123456  
Capacity : 10

Hospital 2, Zip-code - 123456  
Capacity : 25

Vitals

Metrics

Exposure

Symptoms

4

Weight (Kg)

50

5

Height (cm)

100

6

Age

25

Update

View History

Your BMI is


25.0

### 5.2.5. Track Exposure (Location History)

Personal Quarantine

personalq.com

Cases by county



Nearby Hospitals

Hospital 1, Zip-code - 123456  
Capacity : 10

Hospital 2, Zip-code - 123456  
Capacity : 25

Vitals


Metrics

Exposure

Symptoms

Location zip-code

Date of exposure



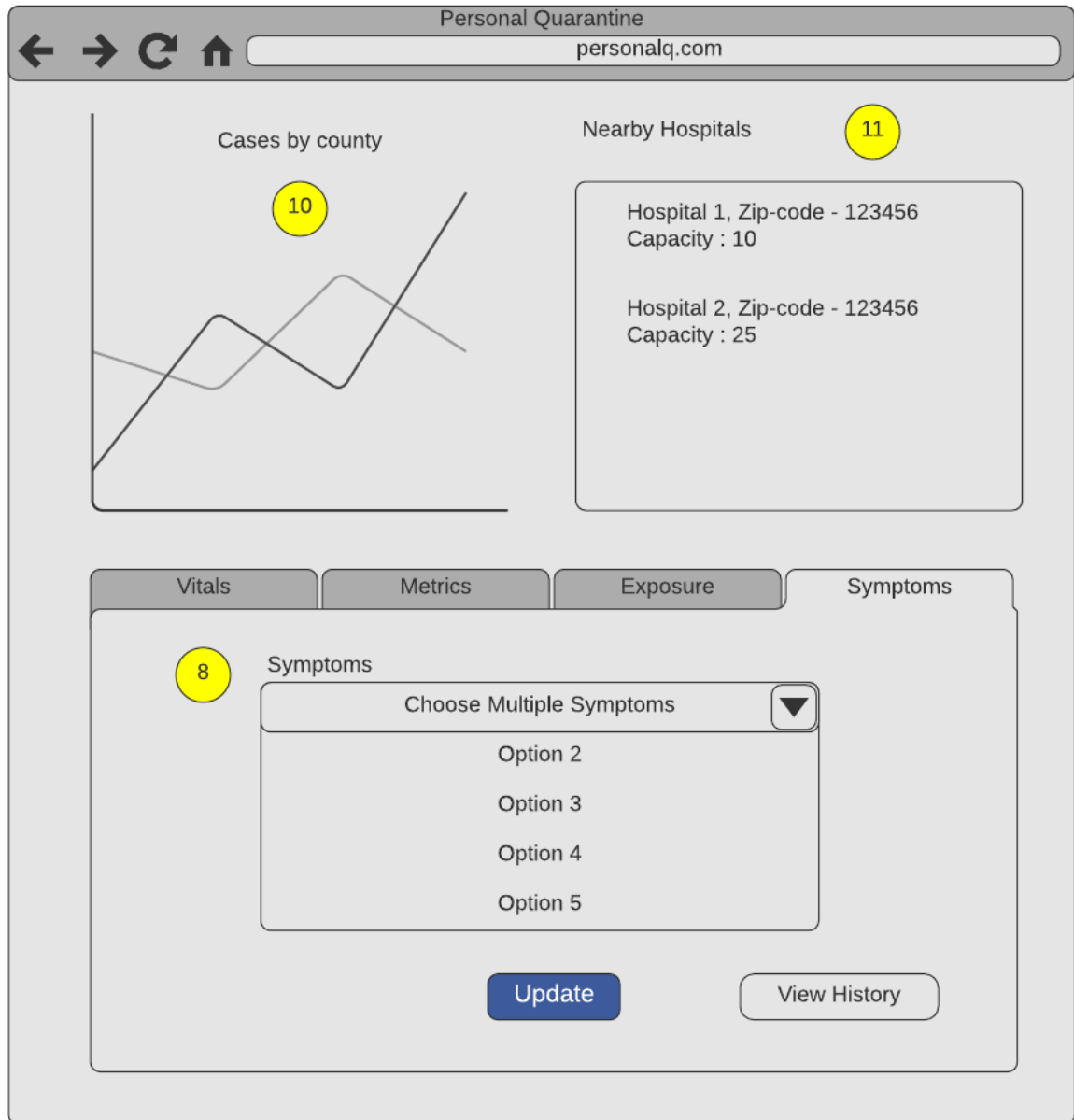
Time of exposure (Optional)

12

Update

View History

### 5.2.6. Track symptoms, visualize test cases and hospitals



## 5.3. High fidelity mockups

### 5.3.1. Admin - Add symptoms

# Personal Quarantine Admin

## Insert Symptoms

Enter password:

Enter symptom:

### 5.3.2. User Registration

## Registration

<b>Username</b>	<b>Height</b>	<b>Age</b>
<input type="text" value="Username"/>	<input type="text" value="Height (cm)"/>	<input type="text" value="Age"/>
<b>Password</b>	<b>Weight</b>	<b>Zip-code</b>
<input type="text" value="Password"/>	<input type="text" value="Weight (kg)"/>	<input type="text" value="Zip-code"/>
<b>State</b>	<b>County</b>	<b>Sex</b>
<input type="text" value="Select State"/>	<input type="text" value="Select County"/>	<input type="text" value="Select Sex"/>

[Already registered? Login here](#)

5.3.3. User Login

# Login

**Username**

Username

**Password**

Password

Login

[New? Register here!](#)

5.3.4. Health Vitals (Add Temperature, heart rate, oxygen level, test results)

Cases by County - Last 7 days  
Average Age Group - 25

Day	Cases
0	100,000
1	350,000
2	250,000
3	450,000
4	300,000
5	150,000
6	400,000

Butte, California

Nearby Hospitals

Hospital 1  
Capacity - 120 beds  
Address for Hospital 1

Vitals

Metrics

Exposure

Symptoms

Temperature (C)

Heart rate (bpm)

Oxygen Level (%)

Temperature

Hearth rate

Oxygen Level

Name of the test

Was the test result positive?

Test name

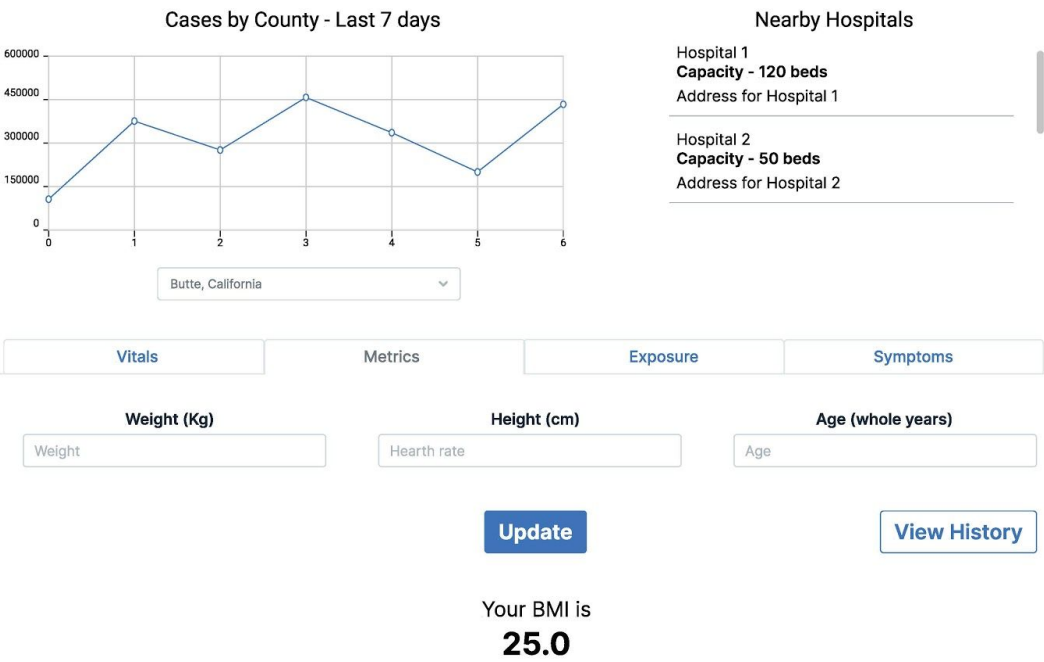
Choose result

Update

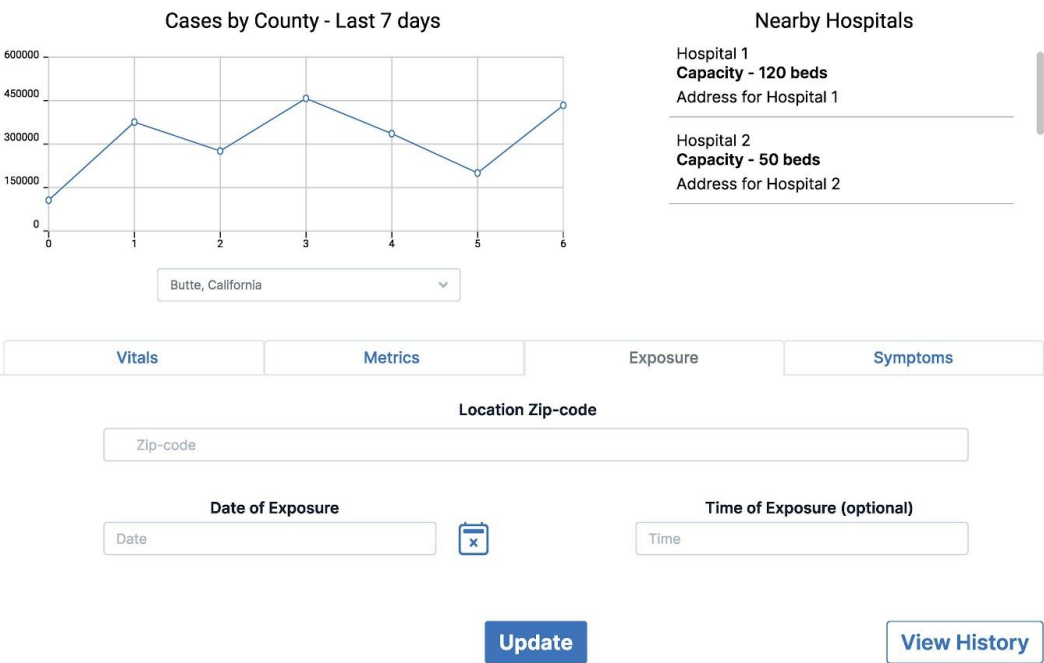
View History



5.3.5. User Metrics (Weight, Height, Age).



5.3.6. Exposure (Add date, time, zip-code)



5.3.7. Add Symptoms

Cases by County - Last 7 days

Day	Cases
0	100,000
1	350,000
2	250,000
3	450,000
4	300,000
5	150,000
6	400,000

Butte, California

Nearby Hospitals

Hospital 1

Capacity - 120 beds

Address for Hospital 1

Hospital 2

Capacity - 50 beds

Address for Hospital 2

Vitals

Metrics

Exposure

Symptoms

Symptoms (Choose multiple if applicable)

Choose symptoms

Dry Cough

Fever

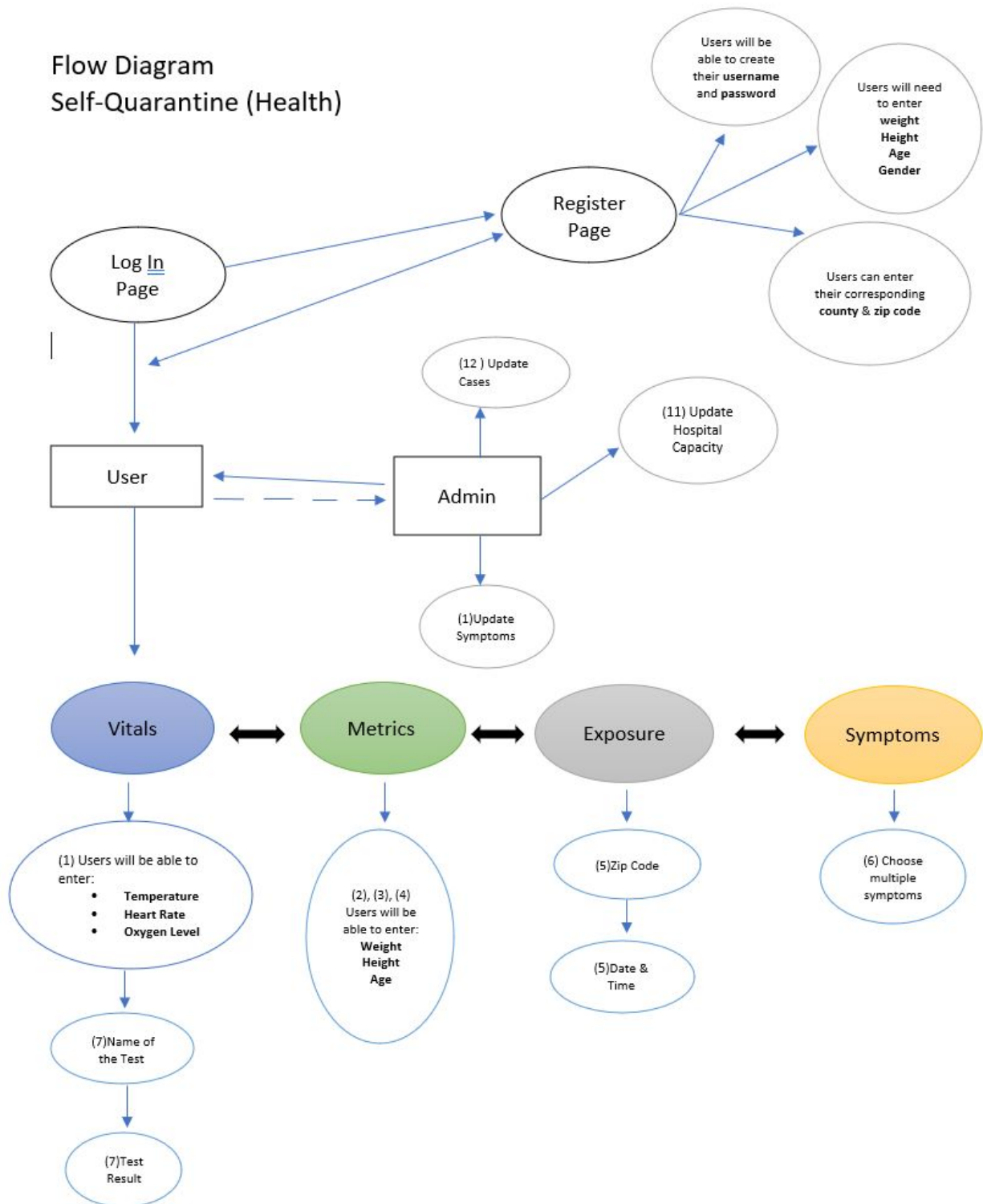
Tiredness

Update

View History

## 5.4. Flow Diagram

Flow Diagram  
Self-Quarantine (Health)



## 6. Implementation Narrative

- 6.1. For the initial design of the application we created mockups for the user interface with LucidCharts .Then our team used HTML with embedded PHP to create the website. PHP was used to establish a connection from the website to MariaDB, so that we could access and display the data stored in our database. SQL queries were written inside of PHP, which was embedded in HTML, so that we could display the data on the website in tables. For ease of user interface designing, Bootstrap 4 open source library was used. Searchable drop down functionality was achieved using bootstrap-select open source library.
- 6.2. To include the 3rd party libraries, a common “root.html” was used. cURL functionality of the PHP was used to fetch the remote data in CSV format from New York Times github repository. This parsed data was constructed as a SQL query to insert all the cases and counties in the appropriate table. This code can be found commented in “home/main.php” file.

### 6.3. Features Table -

# of physical tables	# of views (derived tables)	# of UI pages	# of queries or updates	Total Constraints FKs	Total Constraints !Null	Total Constraints Unique	Total Constraints Other	Tools Summary
9	0	3	6	8	37	9	3	Bootstrap 4, Figma, ERDPlus, LucidChart, Draw.io, Google Docs, HTML, PHP, MariaDB, Charts.JS

### 6.4. SQL Statements - Complex query

- 6.4.1. 1. The goal of the query is to get the average age of the users for the given zip-code area (in this case, zip-code is 89995).

```
select p.zip_code, AVG(m.age) from person as p, metrics as m where m.username = p.username group by p.zip_code having p.zip_code=89995;
```

Output is as follows

```
MySQL [lorozco5]> select p.zip_code, AVG(m.age)
-> from person as p, metrics as m
-> where m.username = p.username
-> group by p.zip_code
-> having p.zip_code=89995;
+-----+-----+
| zip_code | AVG(m.age) |
+-----+-----+
| 89995 | 23.5000 |
+-----+-----+
1 row in set (0.001 sec)

MySQL [lorozco5]> 
```

- 6.4.2. Significant challenges faced by the team was front-end development and fetching data from the remote source. This was overcome by watching LinkedIn Learning videos and w3school.com website as well as taking some assistance from TA Subhed.
- 6.4.3. Project Status - Login and registration part of the project is fully developed with validations. Currently, users can insert their vitals in the database and view the history in list format. Additional time will be required to develop it further for Exposure tracking, Metrics recording and symptoms recording. While the data was successfully fetched from the remote source, additional time is required to integrate it properly with Charts.JS library. Remaining parts includes fetching and parsing of Hospital data to show the list of nearby hospitals and their capacity. Automation of fetching this remote data would be helpful for the user.