SW Engineering CSC648/848

Fall 2020

Code: “Coronafirus”

Section 02

Team 01 Members:

Thomas Abadines (Team Lead) (tabadines@mail.sfsu.edu)

Vito Gano (Project Manager) (vgano1@sfsu.edu)

Souhib Trabelsi (Github Master)

Junhao Zhai (Backend)

Amily Wu (Document Master)

Trung-Duong Pham-Vu (Front End)

Milestone 4

11/10/2020

Revision: v.1.0

(Submitted: 11/10/2020)

History:

Revision: v.1.0 - Released (11/11/2020)

1) Product summary(e.g. how would you market and sell your product–about ½ page)-(Junhao)

- The name of the product that we are advertising is called Coronafirus. **DONE**
- Explicit itemized list of ALL major committed functions, 1 line per function (your FINAL P1 functions for which you will be graded) your team shall actually deliver (and test for).This is your FINAL functional commitment e.g. failure to deliver on some of P1 functions results in reduced grade. Please write it in the list format(each item 1 line of text) so it is easy to check. List of functions to be written in regular English and not like specs. You can stay with the list recommended in M3 **KINDA DONE**
- Global team only: say what P1 functions (P1 functions or back-end vs. UI) will you develop, and which will be developed by Filda team –the best you know at this point.
- Say what is unique in your product (if anything) **Done**
- URL to your product accessible to instructors, on deployment server **not done**

The name of the product that we are advertising is called Coronafirus. It combines two of our main focuses, Coronavirus and Wildfires in California. It's a website that has an interactive map that when you click on one of the county areas, information about it, either covid or wildfire, will pop up. We have a toggle button on the navigation bar that can switch the UI focus to either Covid or Wildfire, with Covid having a green-based theme and wildfire having a red-based theme to differentiate. A possible useful feature in our product is getting alert messages. If the general user wants, we have an optional button in our navigation bar to get alerts by signing up with their email and selecting their county, they are not required to make an account or password in our website to receive alerts. Since our website isn't a phone app, it's not as convenient to receive alerts through our website, so email would be more easily accessible especially because most people have smartphones and a default email app available. For directors they have premade accounts and we made the tasks they need to do very simple with our UI. We focus our product on usability to make it simple for our users to read data related to wildfire and coronavirus. In a few clicks a user can see data related to his county.

As for our website, we provide the following first priority to our website user.

1. Allow user to search data about Covid and Wildfire with certain Keyword
2. Allow user to view the Interactive map with names of county
3. Allow user to view more detailed after the choose the general target
4. Allow director from different department to login with premade account
5. Allow the administrator to login to the account to approve data updates.
6. Allow user to obtain alert to their desired email account

Itemized list

1. View Interactive Map (Search)

2. View Data from using Interactive Map
3. Login (Covid Director)
4. Login (Fire Director)
5. Login (Administrator)
6. Covid Metrics Input (Infected)
7. Covid Metrics Input (Recovered)
8. Covid Metrics Input (Death)
9. Fire Metrics Input (Acres Burned)
10. Covid Alert, County Health director can send alerts to their county.
11. Fire Alert, Fire Director can send an alert to their county.
12. Posting Items and Data (Administrator), items will need to be approved by the administrator before posting it online
13. Register to get alerts (Users)

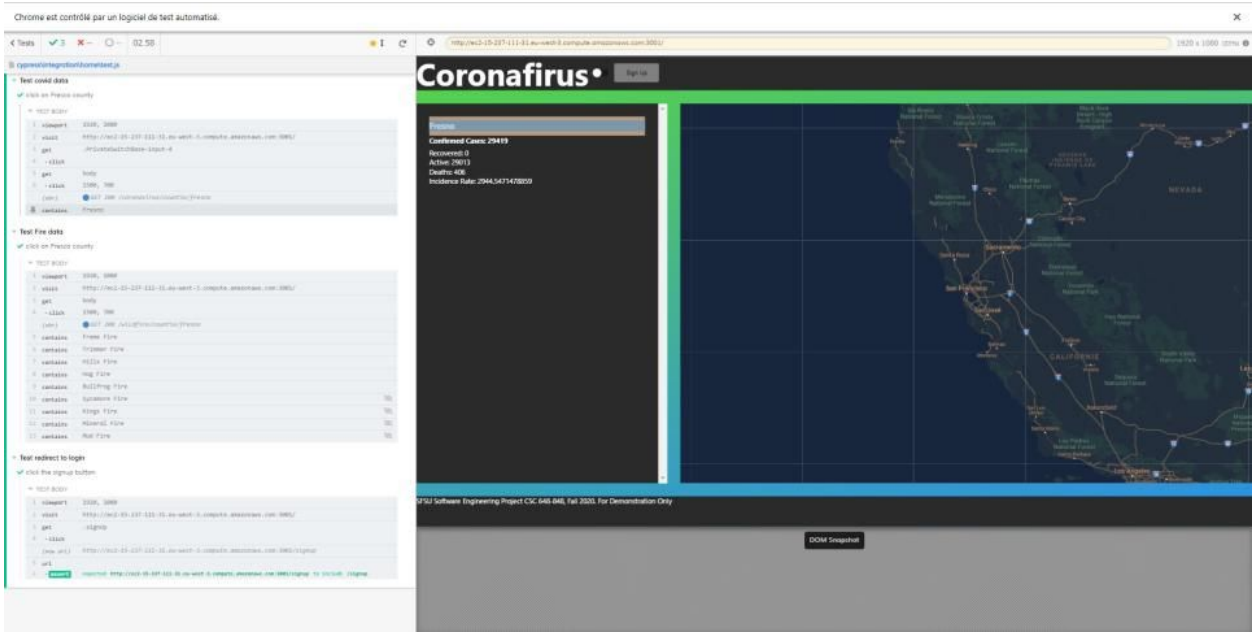
Our products provide users with the following special functions:
1. We allow users to use the toggle button in the upper left corner to choose to watch Coronavirus or Wildfire information.
2. In order to ensure the accuracy of the data, we pass the data through the department director and administrator inspections.
3. We used multiple unordered passwords to determine the security of the account.
4. We separate the login interface of administrator and director from the login interface of ordinary users.

URL to our product: http://coronafirus.team:3001/

2) Usability test plan

Test objectives-The main function being tested is the interactable map that is usable by all users. This test will attempt to measure how easy it is to use the map and if there are any possible confusing aspects that may increase the difficulty of using the map. Since the map will be the main way users will find and interact with the data being provided by the backend, it is essential that it is both easy to use and understand.

Test background and setup -
System setup: Cypress, Google forms



Starting point: Users will begin from the landing page of the Coronafirus webpage. From there users will attempt to find the county that they wish to observe and the data that they are interested in.

Intended users: The intended users of this function will primarily be the general users of the app, with a secondary focus on the county and fire directors if they wish to find out the status of nearby counties.

URL of the system to be tested:
http://coronafirus.team:3001/
https://forms.gle/5UxtpEyzVh5JJEBJ8

What is to be measured:

1. How easy is it to navigate the UI? Is it easy to understand how to use the website?
2. How intuitive it is to use the map to find the information the user is looking for. How clear is the information presented and is it relevant to your search?
3. Was it easy to quickly find the area or information you were looking for?

The main intended users of this function will be the general users, since it will be the easiest way for them to quickly get information about their area. In this test, how

quickly the information that the users seek, as well as the clarity of the information found, will be judged with the reviewers rating the map on a sliding scale.

Usability Task description-
The usability task will consist of mainly four parts. The first task will be to find the county that the user is interested in and click on county. The next part of the test will be to search for covid results of their selected county. After finding the covid data of the selected county, the testers will use the map to search for fire results of the same county. Finally the testers will select another county and compare the results to their selected county to ensure that the information being shown is comparable.

Effectiveness:
Effectiveness will be measured by three factors:
How many testers are able to find the county they want to observe?
How many sources of confusion are there for navigating maps?
How long does it take testers to find the information they are looking for?

Efficiency:
Efficiency will be measured by three factors:
How many clicks does it take for testers to reach final results?
How long does it take for testers to get to the desired page/results?
How many transitions does it take to get all the information about their county?

Lickert subjective test:

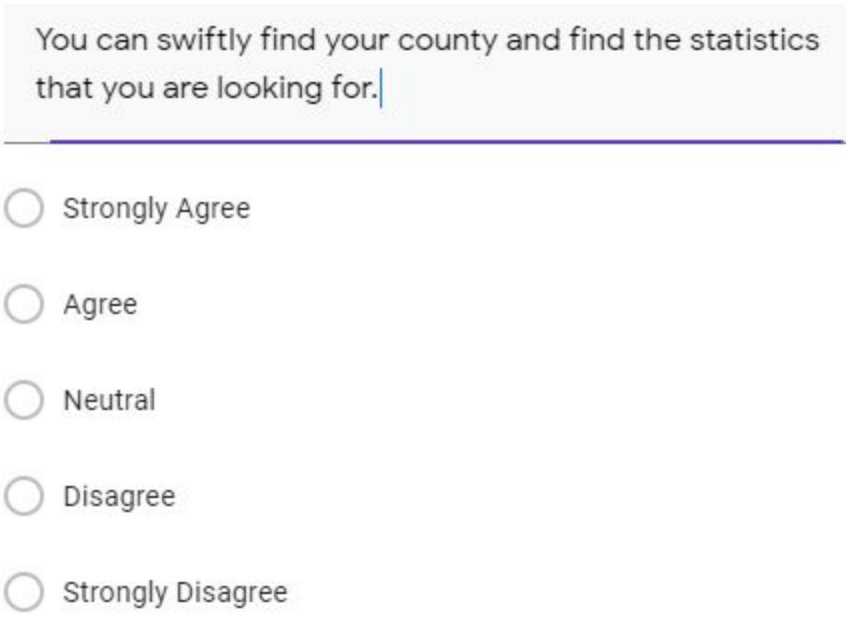
https://forms.gle/5UxtpEyzVh5JJEBJ8

The website was easy to navigate and it was intuitive to use.

- ○ Strongly Agree
- ○ Agree
- ○ Neutral
- ○ Disagree
- ○ Strongly Disagree

It is easy to find the information you are looking for and the information is relevant to your query.

- ○ Strongly Agree
- ○ Agree
- ○ Neutral
- ○ Disagree
- ○ Strongly Disagree



3) QA test plan-max 2 pages

For the same function you chose for the usability test,write a QA test plan(check class slides), with sections as follows

-Test objectives:

-what is being tested

-HW and SW setup(including URL):

-Feature to be tested

-QA Test plan:

-table format: 3 test cases and results of testing them on your system:appr. 1 page. Use tabular formats as in the class, You must provide a QA test plan in the format of the easy to read tabular form allowing easy reading and analysis by management e.g. like presented in the class slides on SW QA.

Suggested format for QA Test Plan Table: table columns are:

- test #;
- test title;
- test description;
- test input;
- expected correct output;
- test results (PASS or FAIL for each tested browser)

b) You also must perform the testing as per the plan above and record the results in a form above. Apply the above test on 2 browsers of different type and record it in the above QA test plan table

---

Test Objective - Mapview

What is being tested?

We are testing the features of the main page

Software setup(including URL)-

http://coronafirus.team:3001/

Feature to be tested-

Interactable map

QA Test plan: -table format:

Chrome

Number	Title	Description	Input	Expected correct output	PASS/FAIL
1	Test covid data	Click on Fresco county	Clicks	Display Fresco on page	PASS
2	Test fire data	Click on Fresco county	Clicks	Display wildfires in Fresco county	PASS
3	Test signup button	Click on signup button	Clicks	Redirection to sign up	PASS

Firefox

Number	Title	Description	Input	Expected correct output	PASS/FAIL
1	Test covid data	Click on Fresco county	Clicks	Display Fresco on page	PASS
2	Test fire data	Click on Fresco county	Clicks	Display wildfires in Fresco	PASS

				county	
3	Test signup button	Click on signup button	Clicks	Redirection to sign up	PASS

4) Code Review:

a. By this time you should have chosen a coding style. In the report say what coding style you chose.

b. Choose the code (substantial portion of it) related to the feature you used for QA and usability test. You need to submit an example of the code under review ,(or part of it–2 pages or so MAX) for this function to be peer reviewed, and document this as follows:

1. One team member should submit code to other team member(s) for peer review.
2. Peer review should be performed by other group member(s)(1 review is OK).
3. Peer review is to be done by e-mail and comments are to be included in the code
4. Submit the email containing or screen shot of the peer review and commented code and e-mail communication related to this in your Milestone 4 document.

```
import React, { useState } from 'react';
import axios from 'axios';
import { Route, Link, Redirect } from 'react-router-dom';
import '../styles/frontendTemplate.css';
import '../styles/listStyle.css';
import MapView from '../components/MapView.js';
import DataView from '../components/DataView.js';
import Secret from '../data/Secrets';
import Switch from '@material-ui/core/Switch';


function MapPage() {
    const [results, setResults] = React.useState(null);
    const [mode, setMode] = React.useState(true); // true for Wildfire,
false for Covid
    const [selected, setSelected] = React.useState(null);
    const [redirect,setRedirect] = React.useState(false);
    const [path, setPath] = React.useState("");

    React.useEffect(() => {
      if(selected) {
      filterFunction(selected);
      }
    }, [selected]);

    React.useEffect(() => {
```

```
      if(selected) {
        filterFunction(selected);
      }
    }, [mode]);

    const checkbox = (event) => {
      setMode(event.target.checked);
    }

    const routeChange = () => {
      setRedirect(true);
    }

    async function filterFunction(i) {
      var input;
      if(!i) {
        input = selected.toLowerCase();
      } else {
        input = i.toLowerCase();
      }
      console.log(mode)
      if (!mode) {

axios.get('http://ec2-15-237-111-31.eu-west-3.compute.amazonaws.com:5000/c
oronavirus/countie/' + input)
        .then((res) => {
          console.log(res);
          setResults(res.data);
        })
        .catch(() => {
          setResults(null);
        });
      }
      else {

axios.get('http://ec2-15-237-111-31.eu-west-3.compute.amazonaws.com:5000/w
ildfire/countie/' + input)
        .then((res) => {
          console.log(res);
          setResults(res.data);
```

```
        })
        .catch(() => {
          setResults(null);
        });
      }
  }
  if(!redirect){
    return (
      <div>
        <div className="container">
        <div className={mode ? 'background' : 'background fade'} />
          <div className="wrapper foreground">
              <div className="main-head foreground">
                <h2>Coronafirus</h2>
                <Switch
                  style={{ float: "right" }}
                  checked={mode}
                  onChange={checkbox}
                  name="checkedA"
                  inputProps={{ 'aria-label': 'secondary checkbox' }}
                />
                <button className="signUp" onClick={routeChange}>Sign
Up</button>
              </div>
              <div className="map foreground">
                  <MapView mapsSecret={Secret.GoogleMaps.ApiKey}
filterFunction={filterFunction} mode={mode} setSelected={setSelected}/>
              </div>
              <div className="side foreground">
                  <DataView results={results} mode={mode}/>
              </div>
              <div className="main-footer foreground">SFSU Software
Engineering Project CSC 648-848, Fall 2020. For Demonstration Only</div>
          </div>
        </div>
      </div>
    );
  } else {
      return (<Redirect to = "/signup" />);
  }
```

```
// here




}

export default MapPage;
```

Comments for the code presented above:

1. The code is well formatted in a way that I can understand what are the requested variables and specifications.
2. The code should have some comments as it may confuse other coders who might have a hard time understanding the thought process of how you did your code
3. I noticed that you used spaces in your indentation which is the preferred indentation method which is consistent with coding practices
4. I also noticed that there were too many html tags in the map page. Typically its better to split the html tags into different components so that the pages are composed of react components.
5. In addition to the previous statement, this applies to the JSX tags which should be split into components.
6. I noticed that you forgot an unnecessary comment in your code, don't forget to remove unnecessary stuff in your code

5) Self-check on best practices for security–½ page (Souhib)

We are protecting metric inputs, alert inputs, account handling, and account user information.
The metric inputs will be sent to the Administrator to approve or deny.
The alert inputs will be sent to the Administrator to approve or deny. Information will not be able to go live until the administrator approves the data.
We are using MD5 encryption for the passwords in the database.

MD5 encryption is a one-way encryption, which means that even if someone has access to all encrypted passwords in the database, there is no way for that person to get the non-encrypted password.
We are not using search bar for our users, we allow them to search for their county directly through a map

6) Self-check: Adherence to original Non-functional specs–performed by team leads

1. Application shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0 (some may be provided in the class,some may be chosen by the student team but all tools and servers have to be approved by class CTO). **DONE**
2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers **ON TRACK**
3. Selected application functions must render well on mobile devices **ON TRACK**
4. Data shall be stored in the team's chosen database technology on the team's deployment server. **DONE**
5. No more than 1000 concurrent users shall be accessing the application at any time **ON TRACK**
6. Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users. **ON TRACK**
7. The language used shall be English. **DONE**
8. Application shall be very easy to use and intuitive. **ON TRACK**
9. Google maps and analytics shall be added **DONE**
10. No email clients shall be allowed **DONE**

11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI. **DONE**
12. Site security: basic best practices shall be applied (as covered in the class) **ON TRACK**
13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development **ON TRACK**
14. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Fall 2020. For Demonstration Only" at the top of the WWW page. (Important so not to confuse this with a real application). **DONE**