

SW Engineering CSC648/848 Fall 2020

Company Name: The Dream Team

Application Name: Public Health and Safety in California

Section 02

Team 6

Lauren Wong: lwong15@mail.sfsu.edu Team Lead

Jair Gonzalez: jgonza38@mail.sfsu.edu Full Stack Developer

Alvaro Maroto: amarotoyepes@mail.sfsu.edu Support

Nathaniel Munger nmunger@mail.sfsu.edu Backend Lead

Yann Sainson yann.sainson@gmail.com Frontend Lead

Duy Nguyen dnguye36@mail.sfsu.edu Github Master

Milestone 4

09/22/2020

1. Product summary

Product name: Public Health and Safety in California by The Dream Team

Itemized list of ALL major committed functions:

- Search by county
- Info selection (Covid or Wildfires)
- Data Admin can input data information.
- Registration
- County Covid-19 data
- County Wildfire data
- Data filter
- Map interaction
- Alert system

The main appeal of our site is that it is a one stop shop for wildfires as well as coronavirus without having to sign up for several programs. It is also full state with options for which area to receive alerts. Map interaction will also help with visualization of data, this feature is often separate from actual alert systems.

URL:<https://sfsucsc648.com/DataEntry>

2- Usability test plan

- Test objectives:

The function that is going to be tested is the Data Entry function for both Covid-19 and WildFire data.

This function is going to be tested because it is a very important function in our application as it is the base function for all the data that is stored in the database, and subsequently shown in the screen.

This usability test plan is going to measure the effectiveness, efficiency, if it is error tolerant and if it is easy to learn.

- Test background and setup:

System setup: Dell inspiron 7559 with Windows 10 operating system using the Chrome browser.

Starting point: The current state of our system focuses on implementing the functionality and testing each component. UI and site flow/appearance is minimal. Our site is composed of the main page which our testing will begin. This page contains links to the other components we are testing as the user would either go to Corona or fire data input from the homepage.

Intended users: Our intended users are split into three central groups. The biggest by far is the average user who holds a basic account and visits the website to check the status of either the fire or Covid situations in their areas. The Second group would be the county

administrators who would report the relevant information daily to the website that the average user would rely upon. The last and smallest group would be the system administrators/government officials who would review the input data and approve it to go forward to the public. The group we focused on would be the county administrators and the ability to navigate and input the data. This we thought would be a critical component to the system that warranted more testing.

URL of the system to be tested: <https://sfsucsc648.com/>

- Usability Task description:

TASK	DESCRIPTION
Covid data for San Francisco	Enter Covid19 data for San Francisco county for 11/10/2020
Covid data for Alameda County	Enter Covid19 data for Alameda County for 11/10/2020
Covid data for Calaveras County	Enter Covid19 data for Calaveras County for 11/10/2020
WildFire data for San Francisco	Enter Covid19 data for San Francisco county for 11/10/2020
WildFire data for Alameda County	Enter Covid19 data for Alameda County for 11/10/2020
WildFire data for Calaveras County	Enter Covid19 data for Calaveras County for 11/10/2020

- Lickert subjective test:

1. I found the GUI easy to use:

- ☐ Strongly disagree
- ☐ Disagree
- ☐ Neither agree or disagree
- ☐ Agree
- ☐ Strongly agree

2. It was easy to complete the data form:

- ☐ Strongly disagree
- ☐ Disagree
- ☐ Neither agree or disagree
- ☐ Agree
- ☐ Strongly agree

3. It was easy to submit the data form:

- ___ Strongly disagree
- ___ Disagree
- ___ Neither agree or disagree
- ___ Agree
- ___ Strongly agree

3- QA test plan

- Test objectives:

The objective of the QA test plan is to check if SW performs to specs. This QA will be for the Data Entry function for both Covid-19 and WildFire data.

This test is going to focus on the data forms and is going to test the input of the data focusing on the different parameters, testing if they are in range, out of range, invalid input data, etc.

The test is not going to evaluate

- HW and SW setup (including URL):

Dell inspiron 7559 with Windows 10 operating system using the Chrome browser.

URL: <https://sfsucsc648.com/DataEntry>

- Feature to be tested:

The principal features to be tested are the input forms for both covid and wildfires and the interaction between these forms and the database.

- QA Test plan: - table format:

test #	test title	test description	test input	expected correct output	test results
01	Good Covid data	Test what happens when data that satisfies the parameters is submitted.	San Francisco 3 7 23 153 11/10/2020	Get a new entry in the database that contains the data that has been submitted.	PASS
02	Negative Covid data	Test what happens when negative numbers data is submitted.	San Francisco -3 -7 -23 -153 11/10/2020	The new entry on the database should be rejected because the data	PASS

				contains negative numbers and should not be added.	
03	Invalid Covid data	Test what happens when invalid data is submitted.	0 0 0 0 11/10/2020	The new entry on the database should be rejected because the data contains invalid data as a number is added to a name parameter.	PASS
04	Empty Covid data	Test what happens when no data is submitted.	*empty* *empty* *empty* *empty* 11/10/2020	The input parameters are required so it should alert that you have to complete the information before it submits.	PASS
05	Good Wildfire data	Test what happens when data that satisfies the parameters is submitted.	11/10/2020 11/09/2020 1 San Mateo County 10.0 Test Fire True	Get a new entry in the database that contains the data that has been submitted.	PASS
06	Negative Wildfire data	Test what happens when negative numbers data is submitted.	11/10/2020 11/09/2020 -1 San Mateo County -10.0 Test Fire False	The new entry on the database should be rejected because the data contains negative numbers and should not be added.	PASS
07	Invalid	Test what	11/10/2020	The new	PASS

	Wildfire data	happens when invalid data is submitted.	NONE -1 San Mateo County -10.0 Test Fire False	entry on the database should be rejected because the data contains invalid data as a number is added to a name parameter.	
08	Empty Wildfire data	Test what happens when no data is submitted.	*empty* *empty* *empty* *empty* *empty* *empty* *empty* *empty*	The input parameters are required so it should alert that you have to complete the information before it submits.	PASS

4. Code Review

The image displays two screenshots of a Gmail inbox, illustrating a code review process. The top screenshot shows an email from 'to Ichiyokowong, me, amarotoyepes' with the subject 'Hello Colleagues'. The email body contains a request for a code review, mentioning attached screenshots and a GitHub commit. It lists several file paths: `application/frontend/pages/FireDataEntry.tsx`, `application/frontend/pages/Login.tsx`, `application/frontend/public/CountyData.ts`, `application/frontend/pages/Register.tsx`, and `application/frontend/public/airqranges.ts`. The bottom screenshot shows a reply from 'duy bul' to 'Jair', providing detailed feedback on the code. The feedback includes comments on the `airqranges.ts` file (praising its straightforward nature but suggesting more comments), the `CountyData.ts` file (noting a lack of comments), the `FireDataEntry.tsx` file (critiquing its structure and naming), and the `Login.tsx` and `Register.tsx` files (praising their simplicity). The reply concludes with an overall positive assessment and suggestions for UI improvements and better component organization. Both screenshots show a Windows taskbar at the bottom with various application icons and a system clock indicating the time is 3:35 PM on 11/10/2020.

Mail - Duy Kevin Nguyen - C... x CSC 640-848 Milestone 4 Be... x CSC648-848 Fall 2020 Miles... x SFSU CSC648-848 Fall 2020... x CSC648 Code for Review - d... x JavaScript Comments x + -

mail.google.com/mail/u/0/#inbox/FMfcgxoKjTMNIwPjpcbplk/QBpRgKj

Apps How to make your... What I learned after... What is the BEST W... Why is every indie... Land of the Lustrou... New folder

Gmail Search mail

Compose

Inbox 11

Starred

Snoozed

Sent

Drafts

More

Meet

New meeting

Join a meeting

Hangouts

duy +

No recent chats

Start a new one

Files Code review.zip

to Ichiyokowong, me, amarotoyepes

Hello Colleagues,

Please see the attached file to see code that I would like to submit for code review. I have also included some screenshots of the UI when running the code in development.

This code is also pushed onto github on the latest branch: FrontendDev1

File paths:

application/frontend/pages/FireDataEntry.tsx

application/frontend/pages/Login.tsx

application/frontend/public/CountyData.ts

application/frontend/pages/Register.tsx

application/frontend/public/airqranges.ts

Regards,

Jair Gonzalez

duy bul

to Jair

Hello,

here is the code review with some comments to improve:

application/frontend/public/airqranges.ts Looks good, it is straightforward as it sets the default values for the air quality we can expect. Could use a few more comments to explain what it will be used

application/frontend/public/CountyData.ts /Also looks good sets the values for each county. One note for both these pages is the lack of comments, as the person who built it make sense, but the pe

application/frontend/pages/FireDataEntry.tsx Looks good but the structure and variable naming is messy and can be cleaned up. Once again keep the people who work on the code after in mind oth

application/frontend/pages/Login.tsx

application/frontend/pages/Register.tsx both pages look good, the code is simple and straightforward as there isn't anything complex going on in the front end pages. One note is that everything is ve and dashboard.

Overall I think the code looks really good though it can use some tidying up. Also the lack of UI needs to be addressed and the previous critique on style and appearance should be kept in mind wher drop down selections are clean and work well. They all come from their own folder which reduces clutter and overall everything looks neat for the most part. Though the components you use for the b again the lack of comments. As someone actively working on this project it makes sense but to a newcomer as clean as the components look when they are grouped so close together it is a headach

Reply Forward

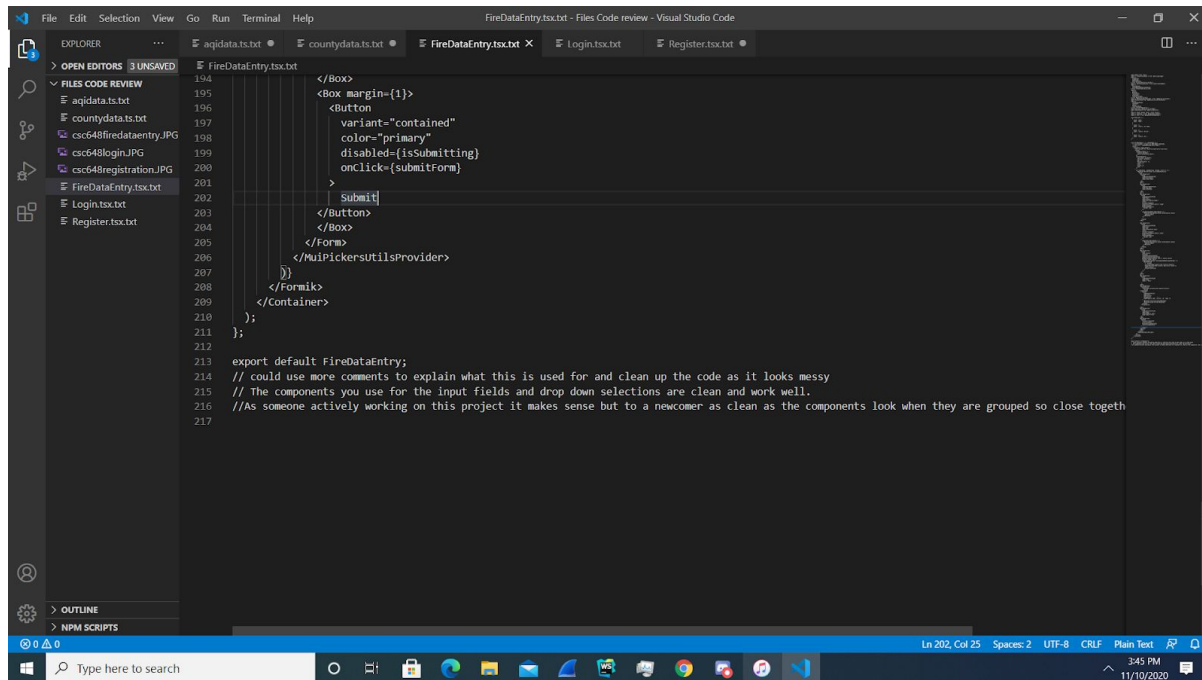
Files Code review.zip

Type here to search

3:35 PM 11/10/2020

Type here to search

3:44 PM 11/10/2020




```
92 </Box>
93 <br />
94 <Box margin={1}>
95   <Button
96     variant="contained"
97     color="primary"
98     disabled={isSubmitting}
99     onClick={handleSubmit}
100   />
101   Submit
102 </Button>
103 </Box>
104 </Form>
105 </MuiPickersUtilsProvider>
106 </Formik>
107 </Container>
108 </Formik>
109 </Container>
110 </Formik>
111 </Formik>
112 </Formik>
113 export default Login;
114 // could use more comments to explain what this is used for and it looks unorganized, add some UI
```

```
26 {county: 'Merced', CountyCode: '24' },
27 {county: 'Modoc', CountyCode: '25' },
28 {county: 'Mono', CountyCode: '26' },
29 {county: 'Monterey', CountyCode: '27' },
30 {county: 'Napa', CountyCode: '28' },
31 {county: 'Nevada', CountyCode: '29' },
32 {county: 'Orange', CountyCode: '30' },
33 {county: 'Placer', CountyCode: '31' },
34 {county: 'Plumas', CountyCode: '32' },
35 {county: 'Riverside', CountyCode: '33' },
36 {county: 'Sacramento', CountyCode: '34' },
37 {county: 'San Benito', CountyCode: '35' },
38 {county: 'San Bernardino', CountyCode: '36' },
39 {county: 'San Diego', CountyCode: '37' },
40 {county: 'San Francisco', CountyCode: '38' },
41 {county: 'San Joaquin', CountyCode: '39' },
42 {county: 'San Luis Obispo', CountyCode: '40' },
43 {county: 'San Mateo', CountyCode: '41' },
44 {county: 'Santa Barbara', CountyCode: '42' },
45 {county: 'Santa Clara', CountyCode: '43' },
46 {county: 'Santa Cruz', CountyCode: '44' },
47 {county: 'Shasta', CountyCode: '45' },
48 {county: 'Sierra', CountyCode: '46' },
49 {county: 'Siskiyou', CountyCode: '47' },
50 {county: 'Solano', CountyCode: '48' },
51 {county: 'Sonoma', CountyCode: '49' },
52 {county: 'Stanislaus', CountyCode: '50' },
53 {county: 'Sutter', CountyCode: '51' },
54 {county: 'Tehama', CountyCode: '52' },
55 {county: 'Trinity', CountyCode: '53' },
56 {county: 'Tulare', CountyCode: '54' },
57 {county: 'Tuolumne', CountyCode: '55' },
58 {county: 'Ventura', CountyCode: '56' },
59 {county: 'Yolo', CountyCode: '57' },
60 {county: 'Yuba', CountyCode: '58' },
61 </>
62 // could use more comments to explain what this is used for
```


5. Security self-check

The application uses a number of recommended best practices for security. JSON web tokens are used for user authentication, and they are refreshed by refresh tokens for an easier user experience. Access tokens are only stored in memory, never to disk, and refresh tokens are stored as HTTP Only cookies. Refresh tokens can only provide new access tokens, eliminating the harms of Cross Site Request Forgery attacks.

Using React as a framework helps mitigate many security risks, as React escapes all content written into JSX, and forbids direct manipulation of the DOM. This, along with use of proper encoding libraries for query parameters and the like, greatly reduces the risk of Cross Site Scripting attacks.

Using a query builder instead of building raw SQL query strings eliminates the danger of SQL injection, as injected script will throw a type error instead of execute.

The application also follows best practices regarding user data. No passwords are stored, only salted hashes, and the only potentially sensitive user data stored by the application is an email, so even in the event of a breach, the effects are not catastrophic.

Basic self-checks confirm that the API requires access tokens with the appropriate role. Middleware protects all sensitive endpoints, and rejects requests that lack tokens, requests that have outdated tokens, and requests that do not have the correct cryptographic signature.

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: all tools that were planned to be used have been used.

2. Application shall be optimized for standard desktop/laptop browsers e.g. must render correctly on the two latest versions of two major browsers

DONE: works on Chrome and Firefox

3. Selected application functions must render well on mobile devices

ISSUE: data display page has issues and there is no clear plan to fix.

4. Data shall be stored in the team's chosen database technology on the team's deployment server.

DONE: yes the data is stored-- but it is dummy data, so ISSUE if we need actual data.

5. No more than 1000 concurrent users shall be accessing the application at any time

ISSUE: there is no way of limiting concurrent users and no clear plan to fix

6. Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.

ON TRACK : privacy is protected but privacy policies are not communicated

7. The language used shall be English.

DONE: yes

8. Application shall be very easy to use and intuitive.

ON TRACK: some fine tuning is needed but we have plans

9. Google maps and analytics shall be added

ISSUE: we used leaflet and open street maps and not google maps and analytics. We need to talk to CTO

10. No e-mail clients shall be allowed

ISSUE: we did use e-mails but this was approved by the CTO-- should still be cleared up though.

11. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.

DONE: there are no such features

12. Site security: basic best practices shall be applied (as covered in the class)

DONE: see part 5

13. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development

ON TRACK: yes

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).

ON TRACK : need to add some text

