

SW Engineering CSC648/848 Spring 2023

SafetyHub

Section: 02

Team: 03

Diego FLORES - Team Lead - Database - dfloresflores@mail.sfsu.edu

Carmelo DE GUZMÁN - Front End Lead - cdeguzman@mail.sfsu.edu

Colline SEGURET - Github Master - cseguret@sfsu.edu

Sumith SHRESTHA - Front End - Sshrestha12@sfsu.edu

Adrian VAZQUEZ - Back End Lead - Database -

avazquez10@mail.sfsu.edu

Isabel FALCON - Front End - ifalcon@mail.sfsu.edu

Milestone 1

02/28/2023

Revision History Table

Description	Date Submitted	Date Revised
Initial Document Outline	02/21/23	
Functional Requirements	02/21/23	03/14/23
Executive Summary, Team & Roles, and Checklist	02/25/23	03/24/23
Main Data Items and Entities	02/26/23	03/09/23
Personas & Main Use Cases	02/26/23	03/14/23
Competitive Analysis	02/26/23	
High Level System Architecture and Technologies Used	2/27/23	
Non-functional Requirements	2/27/23	
Document Submission	02/28/23	

1. Executive Summary:

In recent years, California has seen a rise in all kinds of emergencies, from public health to natural disasters. But at the same time, California has also seen a surge in tech start-ups that shaped technology and how the world operates. California is home to a lot of talent but also all kinds of dangers. We cannot keep ignoring such a dichotomy. Keeping California citizens informed of natural disasters and public health concerns is pivotal to saving lives. Currently, the majority of information that people consume regarding an emergency is shared through social media like TikTok, Facebook, Instagram, and Twitter. These technologies are helpful but are not tailored to spread governmental information fast and accurately. Most information comes from the aftermath of an emergency. We need technology that alerts the public about possible emergencies as soon or even before they happen so the public can take the necessary precautions. We need an app that works alongside governmental institutions to display the most up-to-date and accurate numbers and offers users a good and relatable user experience.

The entire state of California is the addressable market for our app. With a population of over 39 million, a useful app to notify people about emergencies and public health issues is really needed. Our app's significance goes beyond simply saving lives because it has enormous commercial potential. There is a sizable market in California for goods that might assist people in getting ready and staying safe because the state frequently experiences natural catastrophes including wildfires, earthquakes, and floods.

Our team seeks to fill the gap and create an app that benefits California citizens. We are committed to offering the best user experience to the public. We want to create a free-to-use and easy-to-use app that shall keep the public informed about different situations happening around them. We want to offer governmental institutions special access to our app. It is our priority to ensure what we display to the public is the most up-to-date information. Our project seeks to inform the user about different weather emergency levels and infection numbers of COVID-19 in California. Users of our app shall be able to sign-up for notifications that alert them about dangers in their county. We understand the public is drawn to visuals. Therefore we shall display maps for better reference. We want to give the user the freedom to search for the status of different counties within California. We are open to changing and or adding features to our app based on user feedback. All of the features mentioned shall be executed with the user in mind, and our team shall extract all complexity away from the user, so our user base is as large as it can be.

Our team is composed of six members that come from different backgrounds. Our skill levels vary, but we all have one goal in common. We are committed to bringing the California citizens a product they can trust.

2. Personas and Main Use Cases:



Parents



Elders



Living or Visitors in California

Personas:

- **California Residents/Visitors**

About:

- Live in California
- Traveling or visiting California
- Live in areas that experience extreme weather conditions
- Live in different cities that have enforce different health regulations

Scenarios:

- Can to track weather storms and effects of certain weather conditions in their area
- Might need to travel by car for work or school and usual route can be closed due to weather
- If covid or similar virus increase in infection rate in a particular area, they might need to close down or follow certain health regulation that neighboring cities/counties don't have to

- **Parents**

About:

- Living in or are visiting California
- Aware of their children's health
- Can have children with weak immune systems
- Have jobs and children attending school
- Tend to be tech-literate

Scenarios:

- Their babies/young children are most vulnerable to covid effects thus need to be aware of areas with high reported cases
- Extreme weather conditions can affect school/work attendance

- Weather conditions can affect access to electricity, water and gas so need to be updated on storms

- **Elders**

About:

- According to care.com, elders are people who are 65 years old or older
- Live in California
- Tend to be vulnerable to illnesses/viruses
- Might not be tech-literate

Scenarios:

- Are considered priority in case of evacuations so if needed, need to be alerted to weather conditions that lead to evacuations
- Need to be alerted to high virus infections in their area to ensure they can protect themselves

- **Admin**

About:

- SFSU students
- CS majors
- Skills vary but all know beginning level software development knowledge
- Using project to practice skills with database, back-end, and front-end, and github

Scenarios/Roles:

- Designing the UI and adding extra features if needed
- Will address errors that have missed or arise later

Use Cases:

1. Messages

Users with accounts will be able to receive message notifications. The messages will involve information about warnings given to counties. This “chat” box is only one-sided meaning the website is the only one sending out messages.

2. Emergency push notifications

Users do not need an account in order to receive email notifications. As long as an email is registered, notifications about extreme weather in your county will be sent to your email and in application. Similarly, when in areas with high infection reported rates, users will be notified.

3. Maps

No special skills needed to navigate maps. As long as the location is known (either through access to the user's location or by manually entering a location), the user will be able to see if the location is experiencing any extreme weather conditions and alerts for people in that area. There will also be a map used to visually show covid rates throughout California.

4. Resource Page

No skills needed by users to use or access this page. The users will be able to use this page to find links to updated information for covid, extreme weather evacuations and related volunteer groups and donation areas/sites. There will also be related contact info (i.e. emergency lines) displayed for quick access. This page will also be simple for people with different tech level skills so that people who can easily search the web and people who have trouble moving through sites can find what they are looking for.

3. List of Main Data Items and Entities:

For our public protection web application, we shall focus on three main entities that are associated with each of their data items. These entities shall be in the form of users (civilians), administrators (county/ governmental agencies), and technical experts (our team project members), each with their own account privileges to access certain functionalities of the website.

User Controls and Actions shall be specified as follows:

- In terms of the user space, all users shall be able to view and access the main site as well as interact with certain public features such as viewing the map for Covid and weather alerts and searching throughout the website for any search inquiries a user shall request.
 - As for receiving text/email alerts regarding any weather/Covid-related incidents, users shall have to register for an account to receive these push notifications.
 - Civilian users shall also need an account to gain access to previous emergency data as only the current emergency information will be available to view on the website.
- In terms of administrative privileges, county/governmental agencies will have full access to the website when logged in with their respective administrative account.
 - These administrative users shall have the ability to input data metrics onto the site such as Covid-19 infections by the number of cases, as well as weather information that can lead to potentially severe cases of flooding, fire, or freezing in a specific county.
- For the technical experts, these users shall also have administrative privileges, however, they shall have full control over the website such as approving data metrics and public alerts before going live on the site, as well

as moderating content posted on the site and deleting users not adhering to usage compliance.

In terms of the data items, specifically in the database, we shall have two tables, a user's table, and a post table, which contains a list of all registered users from the website and a table for all data inputted by administrators, respectively:

The following two database tables shall be specified as follows:

- For the user's table, we shall have a userID variable with datatype INT that is unique to each registered user. This shall make it efficient for the team project members to manage all users as it shall only require them to search by userID rather than by username.
 - This table shall also have a username and account type field, both of which shall be a variable datatype with a 20-character limit. Specifying each user account as either standard or administrator shall assist in determining if a user can or cannot access a certain feature of the website.
 - Finally, we shall have a createdAt field with the datatype DATETIME as an indicator of when a user registered for an account.
- Comparatively, on the posts table, we shall have a postID variable with datatype INT as each data metric posted by a county official shall have a unique posting ID.
 - We shall also have a username field with a variable datatype of a 20-character limit and a postedAt field with a datatype DATETIME of when the data metric was published to the website.

Regarding our website's data structure, we shall be implementing a hash table/hash map that shall be responsible for mapping Covid metrics and weather-related information to each county. We decided to use a hash map for managing all these data metrics because this type of data structure employs the main concept of key-value pairs, making this a great way to store a collection of various data for various counties:

- In the hash map, the different counties in the Bay Area shall act as *keys* to the table while Covid metrics or weather-related information shall represent *values* associated with each key.
 - As for the data types, we shall have the keys as 20-character fields while the values are of a floating type.

4. Functional Requirements:

1. Users shall be able to receive notification alerts
2. Users shall be able to view entities on the map (such as Covid, weather, etc.)
3. Users shall be able to zoom in/out of county map
4. Users shall be able to search through categories or by text
 - a. the search will be “Fuzzy” using %Like
5. Users shall be able to filter by type of emergency (covid, weather, etc.) to avoid cluttering the map
6. Users shall be able to find information about resources
7. Registration form shall require users to register and shall contain fields such as name, email, etc. (Stored in the database)
8. All users shall have login authentication using sessions.
9. System shall have authorization levels (user, admin)
10. System shall contain historical data (to show past emergencies)
11. System shall implement an API for Covid metrics
12. System shall implement an API for weather-related updates
13. System shall allow users to reset their password
14. Admin shall be able to input metrics on a management dashboard
15. Admin shall be able to review/approve alerts before going live on website
16. Admin shall be able to remove registered users if not in compliance with TOS
17. System shall have Social Media Bot integration
18. System shall have multilingual support
19. System shall be able to export data in CSV or PDF format
20. System shall be able to display the user’s recent search history

5. Non-functional Requirements:

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).
2. Application shall be optimized for standard desktop/laptop browsers e.g., must render correctly on the two latest versions of two major browsers
3. Selected application functions must render well on mobile devices (this is a plus)
4. Data shall be stored in the team’s chosen database technology on the team’s deployment server.
5. Privacy of users shall be protected, and all privacy policies will be appropriately communicated to the users.
6. The language used shall be English.
7. Application shall be very easy to use and intuitive.
8. Google maps and analytics shall be added
9. No email clients shall be allowed. You shall use webmail.

10. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated in UI.
11. Site security: basic best practices shall be applied (as covered in the class)
12. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
13. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project CSC 648-848, Spring 2023. For Demonstration Only" at the top of the WWW page. (Important so not to confuse this with a real application).

6. Competitive Analysis:

Competitive Products Link:

- i. <https://www.fire.ca.gov/programs/communications/red-flag-warnings-fire-weather-watches/>
- ii. <https://www.weather.gov/fire/>
- iii. <https://www.leadingageca.org/emergency>

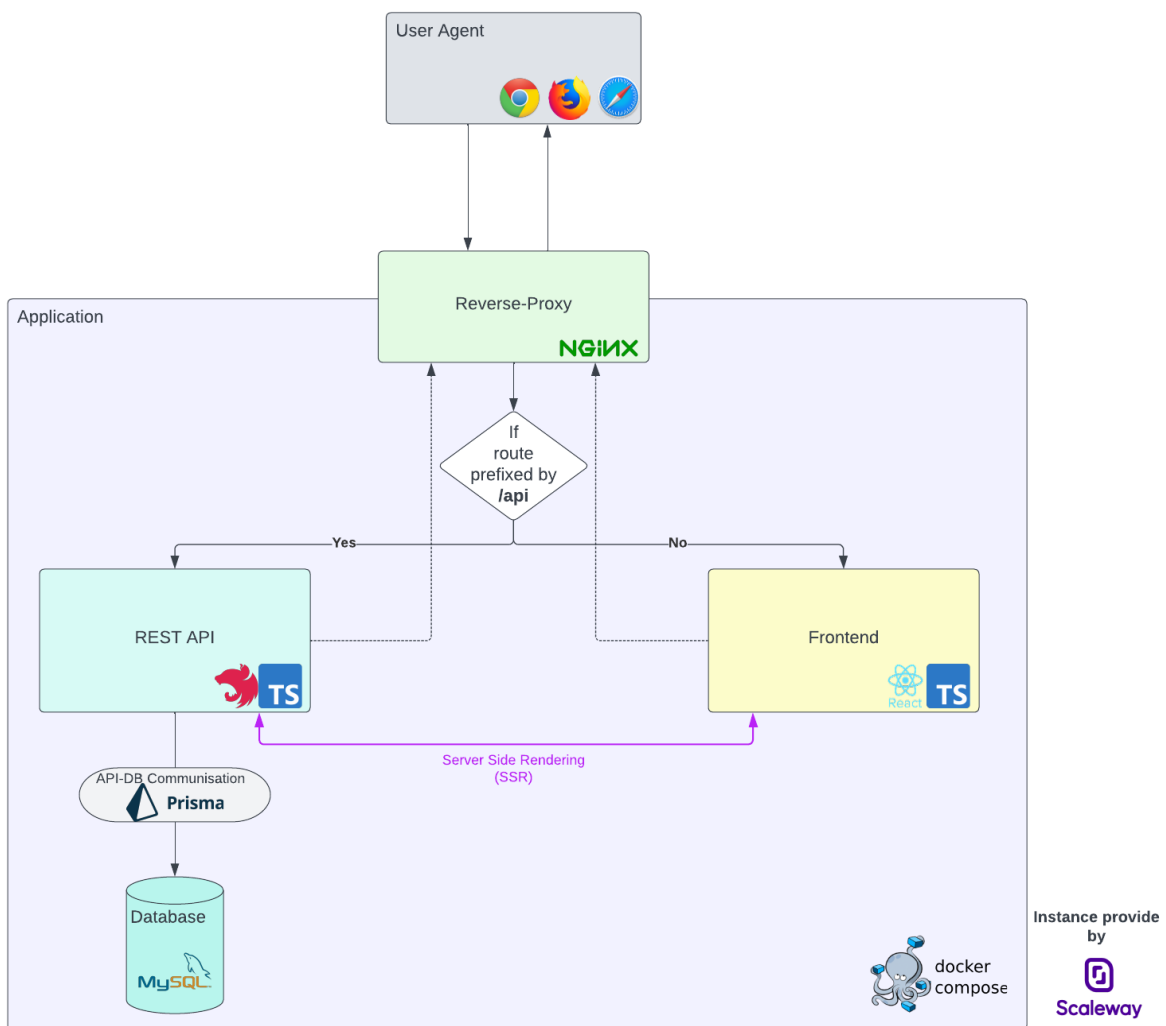
Features	SafetyHub	LAC	NWS	Cal Fire
UI Simplicity	✓✓	✓✓	✓	✓
Sign In	✓✓	✓	X	X
Weather condition	✓✓	✓	✓✓	✓✓
Health	✓✓	✓	X	X
Contacts Department	✓✓	X	X	✓✓
Notification/Alerts	✓✓	X	X	X
Donation	X	✓✓	✓✓	✓✓

Summary:

SafetyHub is a web app that provides assistance during emergencies. It provides crucial information regarding weather, wildfire, health, covid, and security. The weather tab contains weather updates of the location whereas the health tab

navigates the hospital contacts and covid updates. Secondly, the wildfire section navigates through fire situations and updates on nearby locations. Lastly, the security tab navigates through the emergency departments such as 911, fire departments, and so on. This app is mostly for emergencies and it doesn't require users to be tech experts, we have decided to make our app user friendly. The user interface will be simplified, only focusing on necessary information. In comparison to our competitors apps, which includes more-than-necessary features, thus making a challenging navigation for users. Whereas, our app's main focus is on user friendliness.

7. High-level System Architecture and Technologies Used:



All the technical stack will be written in [Typescript](#)

Backend:

- [NestJS](#) A progressive Node.js framework for building efficient, reliable and scalable server-side applications.
- [Prisma](#) Next-generation Node.js and TypeScript ORM.
- [MySQL](#) The World's Most Advanced Open Source Relational Database.

Frontend:

- [ReactTS](#) library for building user interfaces.
- [ChakraUI](#) modular and accessible component library that gives you the building blocks you need to build your React applications.
- [Axios](#) Promise based HTTP client for the browser and node.js.

DevOps:

- [Docker & Docker Compose](#)
- [Nginx](#)
- [Github Actions](#) makes it easy to automate all your software workflows, now with world-class CI/CD.
- [Scaleway](#) Cloud Provider.
 - [PLAY2](#)-NANO instance used

Supported browsers:

- [Chrome](#)
- [Safari Apple](#)
- [Firefox](#)

APIs:

- [Google Maps](#)
- [OpenWeather](#)

8. Team and roles:

- Diego FLORES - Team Lead - Document Master
- Carmelo DE GUZMÁN - Front End Lead
- Colline SEGURET - Github Master
- Sumith SHRESTHA - Front End
- Isabel FALCON - Front End
- Adrian VAZQUEZ - Back End Lead

9. Checklist:

- Team found a time slot to meet outside of the class - **DONE**
- Github master chosen - **DONE**
- Team decided and agreed together on using the listed SW tools and deployment server - **DONE**
- Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing - **DONE**

- Team lead ensured that all team members read the final M1 and agree/understand it before submission - **DONE**
- Github is organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.) - **DONE**