

SW Engineering CSC648/848

Fall 2023 Section 2 M1

“ CalAwareNow - Digital Shield for California ”

Team Lead: Ameen Safi (asafi1@mail.sfsu.edu)

Github Administrator/Scrum Master: Dev Patel

Backend Lead / Front end: Himal Shrestha

Frontend Lead: Chih Lin Chien

Database Lead: Fasika Abera

Backend Engineer: Trina Haque

9/19/2023

Date Submitted	Date Revised

Contents

Teams and roles	0
Executive Summary	2
Personas and main use cases	3
List of main data items and entities	7
Initial list of functional requirements	11
List of non-functional requirements	13
Competitive analysis	14
High-level system architecture and technologies used	16
Checklist	17

Executive Summary

The vast state of California, with its 58 counties which is known for its beauty and diverse ecosystems, unfortunately faces multiple threats ranging from health crises like COVID-19 to natural disasters including wildfires and extreme weather events. With these multifaceted challenges, it becomes imperative for the public to have access to a centralized up-to-date and reliable source of information that can guide them on the necessary precautions and measures to take.

CalAwareNow - Digital Shield for California

We are developing a state-of-the-art web application crafted with the residents of California in mind. Our platform bridges the gap between the state's county departments and its residents by offering real-time updates on health metrics, weather alerts, wildfire evacuation levels, and security advisories. CalAwareNow offers residents a one-stop-shop for crucial updates. For the everyday Californian, this means access to crucial, location-specific information at their fingertips. Whether it's an outbreak spike in Los Angeles or a wildfire alert in Napa, our users can not only search their county's status but also receive proactive alerts tailored to their region. Beyond its primary functions, There are some Key advantages like

1. **Intuitive User Interface:** Designed keeping user experience at its core, CalAwareNow ensures even those not tech-savvy can navigate with ease.
2. **Unique Features:** While competitors offer similar services, our unique features, like advanced mapping interfaces and media-rich content, CalAwareNow ensure stands out.
3. **Safety alerts:** It will show the safest neighborhoods concept for the security aspect.

About Our Team:

CalAwareNow is being developed by a dynamic team of students from San Francisco State University, bringing together diverse skills and experiences. It is a blend of passionate innovators who believe in the power of technology to effect positive change. Leveraging diverse skill sets, from front-end developer to back-end developer, our team is committed to delivering a product that prioritizes user safety and convenience. With **CalAwareNow** we are not just creating a web application; we are building a community-centric platform to protect and empower every resident of California. In conclusion CalAwareNow is not a webapp it's a mission to ensure every Californian stays safe, informed, and prepared.

Personas and Main Use Cases

Personas:

1. Cersei Lannister is the Alameda County Director of Health in Alameda County
 - i) Attitude: She is very passionate about Public Health and wants the public to have the best information possible related to health issues in the county.
 - ii) Skills: She has a masters in Public Health and is very apt in understanding Public Health issues as well as trends.
 - iii) Limitations: She has very limited technical skills other than entering data in Excel.
 - iv) Pain points: She often has to rely on her assistants to share data to the people using the County Website
 - v) Goals: She wants a user-friendly platform where she can easily share data related to Public Health and prepare the public for any type of health issues.
2. Brienne of Tarth is the Alameda County Director of Security
 - i) Attitude: She is in her early 40s, moved to this position after working at FBI for 10 years.
 - ii) Skills: She is very skilled at technology as she had a degree in Cybersecurity as well as excellent emergency management.
 - iii) Limitations: She is very busy with managing security related issues and doesn't have enough time to fix the website that the county has.
 - iv) Pain points: She thinks the current system is poorly designed and makes it ineffective sharing information about security issues.
 - v) Goals: She wants a user friendly UI that has data visualization charts about the security issues, trends and reports.
3. Ser Davos is County Director of Weather & Fire
 - i) Personas: He is in his late 40s and has held this position for 18 years.
 - ii) Attitude: He is efficient at his job and wants to help the people as much as he can.
 - iii) Skills: He is moderately good at using technology and very skilled at handling

disastrous situations that require quick response.

iv) Limitations: He spends a lot of time in the field and often has limited time handling logistical stuff using technology.

v) Pain Points: Current system of alerting about weather & fire is scattered and not centralized.

vi) Goals: He wants a centralized system that will alert about the weather and fire issues in the county more effectively.

4. Jon Snow is the Sheriff/Security of San Leandro (City in Alameda County)

i) Attitude: He is in his early 30s. He was born in San Leandro and spent his whole life in Alameda county. He really cares about his neighborhood and wants to keep it safe.

ii) Skills: He worked in multiple different police departments before being elected as a Sheriff a few months ago. He is very skilled at patrolling, dealing with law enforcement agencies, and handling subpoenas and court orders.

iii) Limitations: Although he is young, he prefers to be in nature during his personal time and avoids technology as much as he can. As a result, he lacks the basic computer skills and often makes mistakes anytime he has to process any report.

iv) Pain points: For his job, he still needs to import/export crime reports, court orders, subpoenas and other reports that he always dreads.

v) Goals: He wants to be able to do his job effectively in a user friendly system where he can easily share about the crime trend and do any technical tasks his job requires.

5. Sansa Stark is a resident of Alameda county.

i) Attitude: She is in his early 20s and very used to using social media, his phones, and computer to stay connected.

ii) Skills: She has a degree in business and like most of his peers, he is pretty good at using laptop, internet, phone for general purposes.

iii) Limitations: She doesn't have cable and is bad about checking news unless it's trending in social media.

iv) Pain points: She often misses important information about the local news as

she forgets to check for updates on her own.

v) Goals: She wants an alert system from her local county which automatically informs her about earthquakes, fire or other natural disasters in her county as well as update her about any potential dangerous situation nearby.

6. Rhaenyra targaryen is the site

i) Attitude: She is 24 who loves all things technology and exciting projects

ii) Skills: She is very skilled in programming, cybersecurity and data visualization

iii) Limitations: She has ADHD and is difficult for her to keep track of things that require manual checking

iv) Pain points: Since most people in the county don't have technical expertise, she is bombarded with technical requests and a large volume of data

v) Goals: She wants an automated alert system that will can be triggered easily where there is a state guideline for emergencies

Use Cases:

1. Signing up for an alert system on the website.

– Sansa creates an account on the website, selects her county and signs up for the alert system for natural disasters in her county that will notify her about earthquakes, wildfire and other issues in her county.

2. Uploading security reports of different cities in Alameda county.

– Brienne logs into the system, uploads reports/charts about past security issues in Alameda county to be more transparent to her residents. The system accepts the reports, sends it to the site administrator for approval and displays them in a user friendly way, and makes the information easily accessible by the public.

3. Sending alerts to the resident about a wildfire.

– Rhaenyra logs into the system as an administrator, filters the residents of the city of San Leandro, and sends an alert to all the residents to evacuate due to a wildfire in the city. The system sends an alert to all those residents right away.

4. Searching and viewing the County crime report.

– Jon logs into the system, clicks on the crime report tab to view the crime report of the city last month when he was on vacation. He then filters the report by his

county and the date he was on vacation. The system provides him with that report based on those filters.

5. Site administrator approval

– Rhaenyra receives a request to go live with the new data uploaded by the County Health Director. She reviews the document and notices that it's missing some key field. She responds to it by requesting more all the required fields.

Main Data Items & Entities

Entity Description

1. **Users** - represents registered user accounts in the system

- ◆ user_id: unique identifier for each user (key) (e.g., 1, 2, 3)
 - ◆ full_name (e.g., John Doe)
 - ◆ username (e.g., johndoe1)
 - ◆ email (e.g., johndoe@gmail.com)
 - ◆ password (encrypted)
 - ◆ user_type (e.g., county director, public)
 - ◆ county_id (foreign key)
 - ◆ registration_date (timestamp)
 - ◆ last_login (timestamp)
 - ◆ department_id (foreign key)

2. **Roles**

- ◆ role_id: unique identifier for each role (key) (e.g., 1, 2, 3)
- ◆ role_name (e.g., public, administrator, county director)

3. **CA_Counties** - represents counties in California

- ◆ county_id: unique identifier for each county (key) (e.g., 1, 2, 3)
- ◆ county_name (e.g., Orange county, Merced county)
- ◆ state (defaults to california)
- ◆ population (e.g., 815,201)
- ◆ county_status (boolean)(e.g., true, false depending on whether there is any public safety concern)
- ◆ county_longitude: double (e.g., [San francisco](#) → 37.773972)

- ◆ county_latitude: double (e.g., [San francisco](#) → -122.431297)

4. Departments - represents the different county departments responsible for public protection

- ◆ department_id: unique identifier for each department (key) (e.g., 1, 2, 3)
- ◆ department_name (e.g., Health department, Security department)
- ◆ department_location (e.g., 11 abc street, san francisco, CA 90000)

5. Covid_Metrics - stores covid metric for each county

- ◆ health_metric_id: unique identifier for each health metric (key) (e.g., 1, 2, 3)
- ◆ county_id (foreign key)
- ◆ cases_per_100k (e.g., 62.1)
- ◆ deaths_per_100k (e.g., 2.8)
- ◆ recored_at (timestamp)

6. Weather_Metrics - stores weather metric for each county

- ◆ weather_metric_id: unique identifier for each weather metric (key) (e.g., 1, 2, 3)
- ◆ county_id (foreign key)
- ◆ extreme_condition_type (e.g., Thunderstorms, Heat waves)
- ◆ recommended_actions (e.g. Road A blocked due to the weather condition, use Route B)
- ◆ recorded_at (timestamp)

7. Wildfire_Metrics - stores wildfire metric for each county

- ◆ wildfire_metric_id: unique identifier for each wildfire metric (key) (e.g., 1, 2, 3)
- ◆ county_id (foreign key)

- ◆ incident_description (e.g., Fire on areas A, B, and C. People in these areas may be asked to evacuate if the wildfire worsens. The fire is currently not contained and cover 2 acres of Area B)
- ◆ incident_cause (e.g., under investigation)
- ◆ acres (e.g., 2100)
- ◆ incident_status (e.g., true or false depending on containment percentage)
- ◆ incident_containment (e.g., 25 %)
- ◆ recorded_at (timestamp)
- ◆ last_updated_at (timestamp)

8. Security_Metrics - stores security metric for each county

- ◆ security_metric_id: unique identifier for each wildfire metric (key) (e.g., 1, 2, 3)
- ◆ county_id (foreign key)
- ◆ security_concern_type (e.g., protest, vandalism)
- ◆ security_action_description (e.g., concern abc reported at areas A and B, avoid areas A and B until cleared)
- ◆ recorded_at (timestamp)
- ◆ incident_status (e.g., true or false depending on whether area is cleared
Note: should be NN)

9. Alerts - stores all emergency alerts triggered by administrators

- ◆ alert_id: unique identifier for each alert (key) (e.g., 1, 2, 3)
- ◆ county_id (foreign key)
- ◆ triggered_by (Users foreign key)
- ◆ alert_message

Users		Roles	
user_id	integer	role_id	integer
full_name	varchar	role_name	varchar
username	varchar		
email	varchar		
password	varchar		
user_type	varchar		
county_id	integer		
registartion_date	timestamp		
last_login	timestamp		
department_id	integer		
Covid_Metrics		Weather_Metrics	
health_metric_id	integer	weather_metric_id	integer
county_id	integer	county_id	integer
cases_per_100k	double	extreme_condition_type	varchar
deaths_per_100k	double	recommended_actions	long text
recored_at	timestamp	recorded_at	timestamp
CA_Counties		Wildfire_Metrics	
county_id	integer	wildfire_metric_id	integer
county_name	varchar	county_id	integer
state	varchar	incident_description	long text
county_longitude	double	incident_cause	long text
county_latitude	double	acres	integer
county_status	boolean	incident_status	boolean
population	integer	incident_containment	double
		recorded_at	timestamp
		last_updated_at	timestamp
Departments		Security_Metrics	
department_id	integer	securtity_metric_id	integer
department_name	varchar	county_id	integer
department_location	varchar	security_concern_type	varchar
		security_action_description	long text
		recorded_at	timeestamp
		incident_status	boolean

Initial list of functional Requirements

1. **User Registration:** Allow users to register with valid email addresses and confirm their emails.
2. **User Profiles:** Registered users can create and manage their profiles, including contact information and notification preferences.
3. **User Authentication:** Implement secure login functionality for registered users.
4. **Role-Based Access Control:** Assign roles to users (e.g., public, county director, administrator) to control access and permissions.
5. **County Data Entry:** Enable county department directors to input COVID-19, wildfire, security, and weather metrics.
6. **Data Validation:** Validate data entries to ensure accuracy and consistency.
7. **User Feedback:** Allow users to provide feedback or report issues related to the platform or data accuracy.
8. **Alerting System:** Provide a system for sending alerts to registered users based on state guidelines and user preferences.
9. **Alert Preferences:** Allow users to set their notification preferences for different types of alerts.
10. **Search Functionality:** Implement a search feature for users to find county-specific information.
11. **Mapping Interfaces:** Display events like wildfires on interactive maps.
12. **Real-Time Collaboration:** Enable real-time collaboration and chat features for county department directors.
13. **Media-Rich Content:** The platform displays relevant information using media-rich content like images, videos, and interactive graphics.
14. **Privacy Controls:** Users can manage their privacy settings and control the type and frequency of alerts they receive.
15. **Responsive Design:** The web application is responsive and adapts to various screen sizes, including desktops, laptops, and mobile devices.
16. **Social Sharing:** Allow users to share alerts on social media platforms to increase public awareness.
17. **Security Compliance:** Ensure compliance with data privacy regulations, especially for health data.
18. **User Help and Support:** Provide user-friendly help and support resources, including FAQs and a knowledge base.
19. **Security Testing:** Perform security testing to identify and address vulnerabilities.

20. **System Performance:** Ensure the application's performance is optimized to handle a large volume of users and data
21. **Modularity and Scalability:** Develop the system with a modular architecture to facilitate future enhancements and customization for other customers.

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 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, Fall 2023. For Demonstration Only" at the top of the WWW page. (Important so not to confuse this with a real application).

Competitive Analysis

Competitive Products	Competitor Key Features	Our Planned Key Features
https://covid19.ca.gov/state-dashboard/ Covid 19 Tracker for California	<ul style="list-style-type: none"> • Look-up information based on county • Quick access to most recent updates on covid • Easy visual access to view new hospital admissions, deaths, and positive case tests 	<ul style="list-style-type: none"> • We plan to also give access to the most recent updates, but we will do this in a more eye-catching way rather than just the format of simple links. Possible nicer buttons. • On their website you sort of have to scroll in order to see the stats, but ours would be more central to the page it is on.
https://www.airnow.gov/ US Air quality tracker	<ul style="list-style-type: none"> • Allow look up by zip code, city, or state • This site presents live webcam footage of air quality in various areas • Map that shows live updates on where in the country there are wildfires happening 	<ul style="list-style-type: none"> • Perhaps our website could link to the live footage that the government provides. Since it is the government, it shouldn't be against any rules to do this. • We can also implement a live changing map, but ours can have filters which manage what is shown, rather than just wildfires. We can display multiple categories on a map.
https://www.neighborhoodscout.com/securitygauge	<ul style="list-style-type: none"> • Can look up based on address or 	<ul style="list-style-type: none"> • We could implement the safest

Security risk assessment website for people to check crime rates in different areas	<p>coordinates of latitude and longitude</p> <ul style="list-style-type: none"> • Shows you list of safest neighborhoods in any selected city • For any given neighborhood you can get access to a summary, map, trends, risk factors, history and crime forecast 	<p>neighborhoods concept for the security aspect. But we could add them as icons on the map as well, if the user chooses to view them that way, instead of only in list mode.</p> <ul style="list-style-type: none"> • We will utilize the same categories as summary, map, risk, etc. But we should also add live updates, sort of like a news media outlet.
---	---	--

The above products do present useful features for their respective markets. We plan to implement similar features as the ones they provide, but we are determined to improve them, and not just copy. In a nutshell, we strive to provide users with services that help achieve as close to “real-time” updates as possible. This shall also be done while designing a UI that is pleasing to use, and aids in directing the user's attention to the most useful pieces of information, in a direct manner. Our product will also actually be a combination of all 3 products above, providing all of their services, in one grand product. We will utilize graphical representations of stats, as well as map graphics, to provide users with data regarding locations of interest.

High-level system architecture and technologies used

Architecture: MVC

Stack: Node.js, MySQL, HTML/CSS (bootstrap framework)

Google Cloud Compute Engine

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 organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.)

DONE