CSC 648-848
Milestone 1
Team 2
Summer 2019

Executive Summary:

Safe SF is a website that helps raise awareness of environmental hazards in San Francisco. Users can check for hazards in an area to help them make safer decisions. And the website informs city management on where hazards are so that they can be removed quickly and efficiently. The goal of **Safe SF** is to create a safer, cleaner, and more environmentally friendly San Francisco.

Users registered with the site can report hazards. They are required to identify the type of hazard and mark it on a map. Additionally they can submit pictures of the hazard or write a description. Any user (registered or unregistered) can browse through the reports. This can be done by looking at a map with all of the hazards marked. Clicking on one of the hazards will show the user more information on the report. Alternatively the user can look through a list of the hazards with different filtering options to help them find what they are looking for.

City managers have admin privileges on the site. Admins are responsible for updating the status of the report. Once a cleanup crew is sent to remove a hazard, the status of the report is changed to "in progress." And once the cleanup is done, the status is set to "completed." This lets users know that a hazard is being taken care of.

There are other sites that also provide a hazard reporting service. But the majority of these sites don't let users view the hazards other people have reported. And the sites that do are confusing and have a confusing UI.

Safe SF solves these issues by providing a fast and easy way for users to report hazards, view reports, and has admins that keep users up to date on the status of reports. Because of this, **Safe SF** will have a farther reach and lead to more removed hazards than competing sites.

The development team behind the **Safe SF** project is a student startup at San Francisco State University. We are very environmentally conscious and was inspired to create **Safe SF** after seeing the limited resources out there for reporting and finding environmental hazards.

Personae and Main Use Cases:

Persona 1: Dylan - City resident and full-time worker

Attitude:

- Likes going to recreational areas around the city on weekends and likes to take long hikes.
- Is willing and capable of posting about hazards that he finds.
- Does not feel the need to browse for hazards before going anywhere.
- Likes the city and wants to keep it safe for everyone.
- Is attached to his phone 24/7.

Skills:

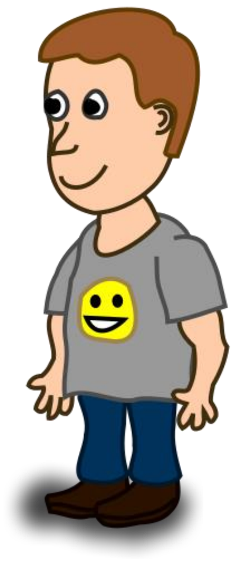
- Comfortable with technology.

Limitations:

- Not always near a computer, so posting from a mobile device is useful.

Pain points:

- Gets frustrated with slow or unresponsive websites. At least expects a loading indicator within a few seconds of clicking a link.



<u>Persona 2: Patricia and Eddie</u> - Baby boomer couple, retirees

Attitude:

- Like to go out for walks together at nearby parks.
- Not too concerned with environmental hazards in general, but would like to report dangerous hazards that may cause immediate safety concerns (such as downed branches on vehicles/roads, heavy trail damage, downed power lines, etc).
- Patient.
- Have no interest in browsing for previous reports.

Skills:

- Very basic computer usage skills:
    - Clicking links.
    - Use url bar to search the web.
    - Transfer photos from their phones to a PC for uploading.

Limitations:

- Not very good at using technology.

- Require specific instructions for most tasks on the web.
    - Poor tech troubleshooting skills, so usage instructions needs to be clear and concise.

Pain points:

- See limitations. If instructions are not clear, they will not bother with using the website or else may input invalid information.

Persona 3: Jenna - Environmentally conscious outdoors enthusiast

Attitude:

- Goes to parks almost daily.
- Would like to do follow-ups on previously reported hazards in whatever area she happens to be visiting for the day.
- Is attached to her phone 24/7.

Skills:

- Comfortable with technology.

Limitations:

- Not always near a computer, so posting from a mobile device is useful.

Pain points:

- Gets frustrated with slow or unresponsive websites. At least expects a loading indicator within a few seconds of clicking a link.

Use Case 1: Dylan goes to a beach

Users skills:
Bachelor’s Degree and very capable of using phone services.

Work setup:
Gets in hiking attire and goes for hiking. Drives to China Basin, parks the car, and start hiking.

Work steps:
Finds a dead sea lion. Opens the website in a browser and clicks on the post tab to report the event.

Difficulties and errors:
Checks to see if there is an already existing report. Then successfully creates a new post.

Performance:
45 seconds per post.

Reliability:
Interface takes more than usual because of low network.

User satisfaction: Very frustrated. Tires easily. Couldn’t make a post at the trail so does it at home.

Dylan decides to go to China Beach this weekend. He finds a deceased sea lion that has washed up on the beach and wants to post about it, but waits until he gets home to do it. At home on his PC, he checks for an existing post about the incident by using the hazard browsing feature. Once he confirms that no one has reported it yet, Dylan clicks the report button to create a new report. He is given clearly labeled input boxes to fill out for the location name, hazard category, location on a map, an optional image, and an optional description. On submission, if he is already logged in, the post is created. If not, he is prompted to register or login before he can post.

Use Case 2: Patricia and Eddie go to The Panhandle

<u>Users skills:</u>
Is not very capable of using technology.

<u>Work setup:</u>
Goes to the local park for a walk.

<u>Work steps:</u>
Finds a broken tree on the trail and wants to report through mobile web. Opens the website in a browser and clicks on the post tab to report the hazard.

<u>Difficulties and errors:</u>
Successfully creates the post.

<u>Performance:</u>
45 seconds per post.

<u>Reliability:</u>
Browser responds within 2 seconds and user is able to post the content.

<u>User satisfaction:</u>
Very satisfied with the service.

The couple go to the Golden Gate Park Panhandle. They come across a downed tree and would like to report on it. They take a picture of the tree and make a post when they get home. On their PC, they create a new hazard report. They are given clearly labeled input boxes to fill out for the location name, hazard category, location on a map, an optional image, and an optional description. On submission, if they are already logged in, the post is created. If not, they are prompted to register or login.

Use Case 3: Jenna goes to Fort Funston

Users skills:
Has a Bachelor’s degree and very capable of using technology.

Work setup:
Goes to the Fort Funston for a hike.

Work steps:
Opens the browser to check for the reported hazards.

Difficulties and errors:
Successfully discovers a hazard on the trail she planned to take and decides to take another route.

Performance:
20 seconds to search for a post in her area.

Reliability:
Browser responds within 2 seconds and user is able to check for the report.

Jenna wants to go to Fort Funston today. Since the Fort Funston has record of closing down due to environmental hazards, Jenna wants to check to see if there are any problems before going out. She goes to the hazard browsing page of the site. Jenna then uses the filtering options to restrict the results to Fort Funston. She finds one that there was a mudslide on the trail she was planning to take and alters her course.

Main Data Items and Entities:

- **Reports**
  - Report_id **(the id assigned to the report upon creation)**
  - Report_image **(the picture, if any, taken by the user)**
  - Report_status **(unassigned, assigned, completed)**
  - Category **(type of environmental hazard)**
  - Report_details **(additional details added by the user)**
  - Report_user_id **(user_id of the person who created the report)**
  - Location_name **(general name of the location)**
  - Loc_lat **(latitude of the location being reported)**
  - Loc_long **(longitude of the location being reported)**
  - Report_insert_date **(timestamp of report create date)**
  - Report_update_date **(timestamp of report updates)**

- **Users**
  - User_id **(id assigned to the user)**
  - Display_name **(user display name)**
  - Email **(email of user)**
  - Password **(hashed password of user)**
  - Report_count **(how many reports user has made)**
  - Join_date **(join date of the user)**
  - Status **(registered, worker)**

Functional Requirements:

- Unregistered users shall be able to view existing reports and filter through them based on category.
- Unregistered users shall be able to pan through a map view of San Francisco to see the reported hazards which are marked by pins.
- Registered user shall be able to report environmental hazards.
- Admins shall be able to change the status of reported hazards after assigning them.

Non-functional Requirements:

- The website shall respond within 2 seconds to an input.
- Files sent to the client shall be under 5 MB.
- The website shall require registration and login to report hazards on the site.
- Captcha authentication shall be used on the registration page.
- Management tools shall be used to manage the team.
- Each page shall have official company logo in the upper left corner.

Competitive analysis:

SF Safe provides a convenient online platform for reporting hazards to the city of San Francisco. Our product gives the user the ability to request a variety of city services as well as browsing other outstanding hazards that have been reported. As a non-registered guest most features of **SF Safe** are still available for use, however it makes future reporting and following-ups more expedient.

The closest competitors to **SF Safe** are the services provided by the city of San Francisco itself, the **Sf Public Works** website and the **SF 311 App**. Ultimately our service will get routed to the same place in public works department to request cleanup, but we offer a superior and more consistent user experience for reporting hazards.

The **SF Public Works** website offers a comprehensive list of services, but is not designed with mobile users in mind. This is not ideal because the average user is looking to report a hazard at the time and place it was found, not at some later time from a computer. The **SF Public Works** offers reporting of hazards, guest access, and the ability to call support directly. No hazard browsing feature is offered, and a map is available to see outstanding events, but is not easily accessible to the user.

The **SF 311 App** is available from the **SF Public Works** website, and is a mobile friendly option for the web service. This app is **SF Safe's** most direct competitor, looking to provide pretty much the same functionality that we offer. **SF 311** allows for the reporting of hazards, a link to call support, strong mobile friendly interface, and guest access. Limited hazard browsing is offered but no map. These are the primary features with which **SF Safe** looks to differentiate from the competition. Customer criticism of the **SF 311** app is often centered around inconsistency and slow response time. Other services such as the reporting of broken parking meters are available on the website, but have no analog in the mobile app. **SF Safe** aims to provide a more thorough and consistent experience.

Lastly another prominent service is the **San Francisco Department of Public Health** Website, which encompasses every manner of possible health reporting and is a top result in google. From the standpoint of hazard reporting this website is far inferior to the others. This is because the site is so cluttered with services a user would have to navigate very deep in to find comparable functionality, and it would be near impossible to use on mobile. It allows for a range of issue reporting including pests, asbestos, abandoned vehicles, and overgrown foliage, however in practice it routes you to 311 to file a complaint.

Feature	**SF DPH**	**SF Public Works**	**SF 311 App**	**SF Safe**
Report	+	++	++	++

Hazards				
Browse Hazards	-	-	+	++
Call Support	+	++	+	+
Mobile Access	-	+	++	++
Map	-	+	-	++
Guest Access	+	+	+	+

+ : feature exists ++ : feature is superior - : feature does not exist

High-level system architecture and technologies used:

High-level system architecture:

Server Host: AWS ec2 (1vCPU, 1 GB RAM)
Operating System: Ubuntu (16.04 Server)
Database: MySQL (5.6.40)
Web Server: NGINX (1.10.3)
Server-Side Language: Javascript

Additional Technologies:

Web Framework: NodeJS (10.16.0), Express (4.17.1)
IDE: VS Code (or developer preference)
SSL Certification: OpenSSL (1.1.1c) <- may change
Front End: Bootstrap (4.3.1)

Team:

- **Alex Wolski**
    - Team Lead
    - Back End Developer
    - Document Master

- **David Stillwagon (Ryan)**
    - Back End Lead

- **Evan Guan**
    - Front End Developer

- **Harshveer Saini (Harsh)**
    - Github Master
    - Front End Developer
    - Back End Developer

- **Lidiya Jebessa**
    - Back End Developer

- **Sunny Srijan**
    - Front End Lead

- **Tristan Mclennan**
    - Front End Developer

Checklist:

Team found a time slot to meet outside of the class
DONE

Github master chosen
DONE

Team decided and agreed together on 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