

Spill

SW Engineering CSC648/848

Section 01 Spring 2018

Team 06

Milestone 1

2/28/2018

Revision History:

Date	Comments
3-11-2018	Revised according to feedback from Dragutin Petkovic, CEO

Peter Mutch (peter2mutch@gmail.com)

Satjit Bola

Alaric Gonzales

Lorraine Goveas

Albert Fernandez Saucedo

Sandhu Harpreet

Executive Summary

Reporting environmental issues is hard to do, and finding out about them is even harder! The EPA manages big issues, but small neighborhood concerns often slip under their radar. City- and county-level websites are often poorly designed and hard to navigate, and it can be difficult for private citizens to be sure their concerns are being addressed.

Spill aims to change that. Spill is a website where users share information about environmental issues. Users can report a new issue, check up on old ones, or find issues specific to an area, whether close to them or far away. Whether you're a corporate whistleblower, concerned parent, or local official, Spill is the place for your environmental concern.

Spill is made by a team of six students from San Francisco State University studying computer science, as part of Computer Science 648: Software Engineering under Dragutin Petkovic and Anthony Souza. For more information, check out <http://sfsuteam06spilldemo.dnsd.info/>.

Use Cases

1. Report an issue

- **Who:**
 - Registered User
 - **Skill Level:**
 - Basic
 - **Process:**
 - Gollum is a neighborhood dad who is concerned about broken glass he saw on the street corner by his favorite park. He opens Spill to post a photo he took earlier that day, and is asked to create an account. He does so and posts the photo, marks its location, and posts a short comment about it.
-

2. Search for an issue

- **Who:**
 - Registered / Unregistered User
 - **Skill Level:**
 - Basic
 - **Process:**
 - King Solomon is interested in seeing what environmental problems there are in San Francisco, California. He opens Spill, searches for San Francisco, and sees a variety of reports, particularly of litter in nearby parks. He wonders if there is this much litter in other cities, and starts a new search for litter. Now King Solomon can see reports of litter from anywhere, no matter how far.
-

3. Check for updates on an issue

- **Who:**
 - Registered User
 - **Skill Level:**
 - Basic
 - **Process:**
 - John wants to know if the litter he reported last week has been cleaned up yet. He finds the report, and can see that although it hasn't been cleaned up yet, it is marked as In Progress. He leaves a comment asking when the work will be completed and thanks the municipal agency for taking care of his city.
- =====

4. Resolve issues

- **Who:**
 - City Employee
 - **Process:**
 - Galadriel, a municipal employee, has just been told that cleanup has finished on the banks of a local river. She searches for that river and sees a variety of posts regarding trash on the banks. She marks them as resolved.
- =====

5. Remove post from website

- **Who:**
 - Admin
- **Process:**
 - Samwise sees an inappropriate or inaccurate post that has been made itself onto the website. As an admin, it is his responsibility to keep the website safe for all ages while still serving valid content.

Using the admin module, he locates the post, removes it from the website, and a message is sent to the user to alert him/her of the events that just transpired.

Data Definitions

Unregistered User: A person who utilizes the website that is not registered with the database. This user can only access the website as “read-only”. This type of user has the least amount of privileges.

Registered User: A person who utilizes the website that is registered with the database. This user is granted special privileges such as posting a new post, editing a previous post submitted by the same user, or deleting a previously created post, as well as all features available to unregistered users.

Admin: A registered user with special permissions geared toward keeping the website running as intended and removing inappropriate content. Site admins are also responsible for tracking and removing/merging multiple posts about the same post.

Issue: A user report of an issue they noticed. May include an image, location, time posted, username of postee, and description of the issue.

Status: The current state of an issue. An issue can be verified, in progress, planned, resolved, or locked.

Verified: In this state, the issue has been approved by an as an accurate representation of a real-world issue.

Planned: In this state, the issue is planned for cleanup or resolution but is not being actively treated.

In Progress: In this state, the issue is being actively cleaned, treated, or otherwise handled.

Resolved: In this state, the issue has been resolved by the appropriate authorities/facilities. The issue is changed from in progress to resolved by an admin or the user who originally submitted it.

Locked: In this state, only admins can make changes to the post. The locked state is used for any posts that have been previously verified but now has complications.

Agency: The establishment, company, or government branch in charge of resolving or fixing the post in question

Comment: A text message posted by a registered user on an issue

User Profile: User picture, username and activity. Can be edited by user in the user panel

User Panel: Allows the user to view and update their profile and account information and settings

Admin Panel: Allows administrators to help mediate content submitted by users on the website

List of Functional Requirements

1. **Unregistered Users** shall be able to search for **incidents** by location.
2. **Unregistered Users** shall be able to view **incident** reports.
3. **Registered Users** shall be able to **post** a report of an **incident**.
4. **Registered Users** shall be able to **comment** on **unlocked posts**.
5. **Registered Users** shall be able to mark the **status** of their post as **resolved**.
6. **Registered Users** shall be able to mark the location of an **incident** they are posting.
7. **Registered Users** shall be able to **upvote** a **post**.
8. **Agency** user shall have access to reports they are responsible for resolving.
9. **Agency** user shall be able to **comment** on **issue record** under their authority.
10. **Agency** user shall be able to update the **status** of an **incident** under their authority.
11. **Admin** shall be able to **lock** posts.
12. **Admin** shall be able to **remove** posts.
13. **Admin** shall be able to view user's **email addresses**.

List of Nonfunctional Requirements

1. Spill shall be developed, tested and deployed using tools and servers approved by Class CTO and as agreed in M0
2. The website shall be optimized for standard desktop and laptop browser.
3. Application shall have responsive UI code so it can be adequately rendered on mobile devices but no mobile native app is to be developed
4. Data shall be stored in Mongo DB
5. Application shall be media rich (at minimum contain images and maps)
6. No more than 50 concurrent users shall be accessing the application at any time
7. Privacy of users shall be protected and all privacy policies will be appropriately communicated to the users.
8. The language used shall be English.
9. Application shall be very easy to use and intuitive.
10. Google analytics shall be added
11. No e-mail clients shall be allowed
12. Pay functionality, if any (e.g. paying for goods and services) shall not be implemented nor simulated.
13. Best practices shall be used to ensure the website is secure.
14. Modern SE processes and practices shall be used as specified in the class, including collaborative and continuous SW development
15. The website shall prominently display the following exact text on all pages "SFSU Software Engineering Project, Spring 2018. For Demonstration Only" at the top of the WWW page.
16. The website shall be protected by SSL encryption
17. The database shall require authentication to do any and all writes

Competitive Analysis

Feature	US EPA	Ireland EPA	Broward.org	Spill (Future Product)
Text Search	+	+	+	+
Interactive Map	+	+	+	++
Online Complaint Form	+	+	+	+
reCaptcha/check	+	+	+	+
Image Uploading	-	-	+	+

+ : feature exists | ++: Superior | - : feature does exist

Spill will be competitive in the current market by providing a superior interactive map and user image uploading. Spill's interactive map shall allow for local, regional, or national views as appropriate, and filter results according to user queries. Spill's user focus will allow users to view issues most relevant to them. Spill's image uploading functionality will provide additional information, allowing Agencies and users to have a clearer understanding of the incident.

High-Level System Architecture

1. Spill will be developed using the MERN software stack. Data will be stored in Mongo DB. The backend will be written in Node.js using the Express.js framework. The front end will be written in React.JS.
2. Spill will be hosted in Google Compute engine and use nginx as the server.
3. Spill will be optimized for Google Chrome and Mozilla Firefox browsers, the most recent version and one version previous.
4. The source code for Spill will be hosted on a GitHub repository and developed independently, collaboratively and simultaneously by all Team 06 developers.
5. Map integrations will be handled by utilizing the Google Maps API.

Team:

Team Lead: Peter Mutch (peter2mutch@gmail.com)

Backend Lead: Satjit Bolas

Frontend Lead: Alaric Gonzales

Team Members: Lorraine Goveas, Albert Fernandez Saucedo, Sandhu Harpreet

Checklist:

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

Github master chosen: **DONE** (Team Lead: Peter Mutch)

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 it: **DONE**

Team Lead ensured that all team members read the final M1 and agree/ understand it before submission: **DONE**