SW Engineering CSC648/848 Summer 2019

EnvironMate

Team 01

Team Lead: Michael Gilbert
mgilber1@mail.sfsu.edu

Front End Lead: Rajdeep Riar

Back End Lead: Jonathan Julian

Git Master: Carlos Lopez

Front End: Angie Martinez

Back End: Johnathan Lee

Back End: Sandhya Sankaran

Milestone 4

6 Aug 2019

Version: 1.0

1. **Product Summary:**

Name of the product: EnvironMate
URL or Equivalent: ***54.67.108.149***

Priority 1- Must Have

System-

- 1) shall allow 3 types of users; unregistered user, registered user, admin user.
- 8) Registration form: required for users to register. Contains name, e-mail and. Stored in the database.

Unregistered User-

- 9) shall be able to register.
- 2) shall be able to view all the incidents.
  - Use the search bar to view incidents

  3) shall be able to search incidents by zip code.
- 4) shall be able to search incidents by location.
- 6) shall be able to filter incidents by types of incidents.
- 7) shall be able to filter incidents by status.

Registered User-

- 11) shall be able to perform all functions of unregistered users.
- 12) shall be able to login.
- 13) shall be able to logout.
  - Logout button on every single page top right
- 14) shall be able to create incidents; with one photograph, location, type and description.
  - Form on create incidents page with all these categories
- 15) shall be able to view the incidents created by them.

Admin-

- 18) shall be able to perform all functions of registered users.

2. Usability Test Plan:

Test Objectives-

The purpose of our usability test plan is to ensure a user experience. A simple user experience will allow a wide population of users to be able to post an incident without feeling confused or frustrated by the applications complexity. At EnviroMate, we feel an easy to use website will ensure that the users can perform these tasks effortlessly. We will test for how quickly users are able to post an incident and how satisfied they are with the process.

Test Background and Setup-

To effectively test our posting function, we plan on having users to post a couple incidents. We will test our application by using 2 different types of users: novice, intermediate. By varying the users skill levels, we aim to get the average difficulty of posting an incident. To ensure consistency, all users will complete the same set of tasks on the same laptop. No user will receive external help. After completing the assigned task, users will be given a short questionnaire form, which will ask them to assess their efficiency, effectiveness, and satisfaction levels for the post an incident function.

Usability Task Description-

We will ask the user to post an incident to our application. The first task will be for the user to locate where to post a new incident. Next, the user will select an address. After selecting the address, the user will add a description. Finally the user shall upload a picture and complete the process by posting the incident.

We will measure effectiveness on how fast the user can locate where to post an incident and the amount of time it takes to complete this process. We will monitor the users behavior to see if they are having an easy time posting, or if the user feels frustrated. We will also measure the users efficiency by the number of clicks they user takes and how long it takes to complete the process.

Lickert Subjective Test-

I was able to successfully locate where to post an incident

_Strongly Agree	_ Agree	_Neither Agree nor Disagree	_Disagree	_Strongly Disagree

I was able to post an incident without difficulty

_Strongly Agree	_Agree	_Neither Agree nor Disagree	_Disagree	_Strongly Disagree

I was satisfied with the posting an incident function

_Strong Agree	_Agree	_Neither Agree nor Disagree	_Disagree	_Strongly Disagree

3. **QA Test Plan:**

Test objectives:

The purpose of QA test plan is to ensure that the user is experiencing a reliable and expected behavior as per the requirements. In this QA plan we will be covering scenarios where the user tries to post an incident using EnvironMate application.

HW and SW setup (including URL):
Machine - Macbook pro
OS - Mac OS
Browser - Firefox 68.0.1 (64-bit)
URL: http://54.67.108.149/incidents/report

Feature to be tested:

Posting incidents using EnvironMate application.

QA Test plan:

Test #	Test Title	Test description	Test input	Expected correct output	Test results
1	Post incident - Happy path	Registered user is able to post an incident	**ZipCode**: 94014 **Location**: 343 Sansome St **IncidentType**: Garbage spill **Description**:Garbage not collected in the park **Image**: Upload an existing image	User should be able to submit the form and redirected to incident detail page	PASS

2	Post incident - Error path	Registered user is not able to post an incident when at least one of the required fields(ZipCode, Location, Incident type,description, image) is empty.	**ZipCode**: 94014 **Location**:343 Sansome St **IncidentType:** “” **Description**: “” **Image**: “”	User should be displayed with validation error when the post button is clicked	PASS
3	Post incident - Error path	Registered user is not able to post an incident when all of the required fields(ZipCode, Location, Incident type,description, image) are empty.	**ZipCode**: “” **Location**:“” **IncidentType:** “” **Description**: “” **Image**:“”	User should be displayed with validation error when the post button is clicked	PASS
4	Post incident - Negative path	User should not be able to post an incident without being logged into the application	**ZipCode**: 94014 **Location**: 343 Sansome St **IncidentType**: Garbage spill **Description**:Garbage not collected in the park **Image**: Upload an existing image	User should be displayed with the Login Popup	PASS

4. **Code Review:**

For coding style we are adopting the Java/JavaScript notation.

```
router.get('/report', function (req, res, next) {
  let _zipcodes      = [];
  let _locations     = [];
  let _incidentTypes = [];
  let _status        = [];
  let _userId        = null;

  if (req.cookies && req.cookies.user){
    _userId = req.cookies.user.id;
  }

  // fetch necessary stuff from db
  models.zipCodes.findAll()
  .then( results => {
    results.forEach((zipcode) => {
      _zipcodes.push(zipcode.dataValues);
    });
    return models.location.findAll()
  })
  .then( results => {
    results.forEach((location) => {
      //console.log(location.dataValues);
      _locations.push(location.dataValues);
    });
    return models.incidentType.findAll();
  })
  .then( results => {
    results.forEach((incidentType) => {
      _incidentTypes.push(incidentType.dataValues);
    });
    return models.incidentType.findAll();
  })
  .then(results => {
    results.forEach((incidentStatus) => {
      _status.push(incidentStatus.dataValues);
```

Peer Review:

RE: Code review for Post incident

JJ **Jonathan Natale Julian <jjulian2@mail.sfsu.edu>**
4:13 PM

To: Sandhya Sankaran

Sandhya,

Your code looks good with the overall structure. The only thing that I see missing are comments in the code especially header comments including your name, date, and general functionality of what this class provides. Other comments would be what the routes themselves provide – and any error checking. This is a pretty easy fix and can be completed in no time.

Please get this resolved before the end of the day tomorrow. Also validate other classes have the same information.

You're doing great work!

Thanks Jonathan.

Sent from Mail for Windows 10

From: Sandhya Sankaran <ssankaran@mail.sfsu.edu>
Sent: Monday, August 5, 2019 3:56:18 PM
To: Jonathan Natale Julian <jjulian2@mail.sfsu.edu>
Subject: Code review for Post incident

Hey Jonathan,

Please find attached the file that contains the code for post-incident. Could you please review it and let me know if any changes are to be made.

Thanks,
Sandhya Sankaran

5. **<u>Self-check on best practices for security:</u>**

- **Secured assets**
  - User data
    - User Id
    - Email
    - Password
      - Using password encryption
  - Database
    - Limit what user can access
      - Write permissions
        - No write permission for unregistered user.
        - Registered users can write only on the incident they are creating.
      - Read permissions
        - Credentials
          - Unregistered users have no read permission on any credentials.
          - Registered users can access only their own credentials.
        - Incidents
          - Everyone can see posted incidents
  - Server
    - Prevent unauthorized individuals from accessing our cloud server
      - Disabled password authentication to the server
        - Password authentication is considered to be less secure
      - Require .pem key to login to cloud server
        - Using .pem key is considered to be more secure.

- Password is encrypted using an open-source hashing library called **bcrypt.**
- Input validation are strictly enforced
  - Search is limited to 50 alpha-numeric characters
  - Minimum password length is 8 characters
  - Email is validated using built-in functions
  - When posting an incident, all possible zipcodes are pre-defined and stored in a drop down.
  - When posting an incident, all possible locations are pre-defined and stored in a drop down.

6. Self-check:

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).

   - ON TRACK

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

   - ON TRACK

3. Selected application functions must render well on mobile devices.

   - ON TRACK

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

   - DONE

5. No more than 50 concurrent users shall be accessing the application at any time.

   - ON TRACK

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

   - ON TRACK

7. The language used shall be English.

   - **DONE**

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

   - **DONE**

9. Google analytics shall be added

   - **ON TRACK**

10. No email clients shall be allowed

    - **DONE**

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

    - **ISSUE** (not applicable in our application)

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

    - **DONE**

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

    - **DONE**

14. The website shall prominently display the following exact text on all pages *"SFSU Software Engineering Project CSC 648-848, Summer 2019. For Demonstration Only"* at the top of the WWW page. (Important so as to not confuse this with a real application).

- DONE