**CEN 4010 Principles of Software Engineering**

**Fall 2019**

**Team 4**

**Milestone 4 Project Proposal and High-Level Description**

**The Freelancers**

**Campus Snapshots**

**Members:**

**Aaron Haim (Scrum Master) – ahaim2016@fau.edu**

**Daniel Fradlin (Front End Lead) – dfradlin2017@fau.edu**

**Richard Urena (Back End Lead) – urenar2016@fau.edu**

**Yiyang Liu (Product Owner) – yliu2015@fau.edu**

**Created November 18, 2019**

**Product Summary:**

*Campus Snapshots* is a web system that people can access and inform users about issues and events around campus. It provides real time snapshots via a map that shows the campus status. If an issue arises, it can allow administration to provide help as needed to those markers located on the map. It can also be used to report events happening on campus such as games, concerts, activities, etc. *Campus Snapshots* is made with the intention of allowing campuses to more easily communicate items in an instantaneous fashion throughout the area. It can be used as an itemized list of problems to solve, a way to better organize what is happening, or as a way to send out notifications about specific problems that cannot be solved at that current moment. It is, in a sense, a one stop shop to know any major developments within a campus.

There are some great advantages to implementing such a system. One major advantage is that the student to administrator ratio is about 22:1. With more eyes, more issues can be reported quickly. It allows for the administration to have extra eyes and stay up to date on the markers. Also, with the ability to upload photos of the reported issues, it will allow for issues to be validated and for there to be no false reporting.

Another great part of this system is that if students decide to host their own events, they can add that event to the site for other students to join if they are interested. FAU doesn't have a club for everything so this system makes it easier for students to find others that share the same hobbies/interests.

**Major Committed Functions**

* Users can create accounts if they do not already have one
* Users can login to their account
* Users can create markers (events/issues)
* Users can attach photos to created markers

**URL:** <http://lamp.cse.fau.edu/~cen4010fal19_g04/CampusSnapshotBeta/CampusSnapshot/index.php>

**Demo:**

<https://youtu.be/EVQ2qaTrb_c>

**Usability Test Plan:**

The function that will be tested is the Reporting an Issue function. We will test the functionality of the page itself with all the dropdowns. It is important to make sure the submission of the issue is recorded. The issue board must show the submitted issues.

The starting URL is: <http://lamp.cse.fau.edu/~cen4010fal19_g04/CampusSnapshotBeta/CampusSnapshot/ReportAnIssue.php>

We will begin by testing the button itself to make sure the it works across all browsers and to make sure that the design of it is responsive. Next, we will test the page that it leads us to, making sure that it is the correct page and that it is properly designed. On this page we will test the dropdowns and text input to make sure they are functional. We will also make sure that the image upload is functional. Before we leave this page, we will submit the response and make sure that it has been recorded. Finally, we will make sure that the response is visible on the front page. Also, we will test for overflow, to make sure that if there are too many issues, the page doesn’t break.

Is the page aesthetically pleasing across all screen sizes?

Do all the functionalities perform as intended with no bugs?

Is the page user friendly?

**QA Test Plan:**

This will test the functionality of the Reporting an Issue page. There is a dropdown menu, a text entry, an image upload, and two buttons. There are three tests to perform: functionality, bounding fields, and error pointers.

The user must open the browser of his/her choice and navigate to <http://lamp.cse.fau.edu/~cen4010fal19_g04/CampusSnapshotBeta/CampusSnapshot/ReportAnIssue.php>. From there, the user will go over each feature below to be tested to see if they are all met all on this page. They can view the original page to see if whatever they uploaded actually did.

First, the functionality of each section will be tested. The drop-down menu must be displayed on screen, scrolled through, and able to select an option that gets saved. The text entry box must store text to be saved. The image upload must be able to select an image to upload. The first button must test if it actually submits the query provided the information is correct. The second button must test if it leads the user back to the main page of the project.

Second, the bounding fields of each section must be tested. The drop-down menu cannot be returned or selected as a null value. The text box entry must include a bound between nothing and the max amount of characters allowed. The image upload must only take in png, jpeg, and jpg extensions. The first button must be spammed to see if it stores multiple instances of the data being uploaded causing copies.

Finally, the error pointers must be tested. The drop-down menu should throw an error if no valid location is submitted (if applicable). The text box entry must throw an error if the text entered goes over the character limit. The image upload must throw an error if an incorrect file type is uploaded. When the data is submitted, there should be no storage failures in data.

Here are three test cases:

Test Case 1 – ‘Functionality Test: Check if the drop-down menu can be displayed properly, scrolled through, and store the saved option to be submitted.

Test Case 2 – Bound Fields Test: Check if the image upload only takes in png, jpeg, and jpg extensions.

Test Case 3 – Error Pointers Test: Check to see if the text box entry throws an error if the text entered is nothing or goes above the character limit.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test # | Title | Description | Input | Expected Output | Result |
| 1 | Chrome Functionality Test | Check if the drop-down menu can be displayed properly, scrolled through, and store the saved option to be submitted. | Click | Drop-down location in marker | PASS |
| 2 | Chrome Bound Fields Test | Check if the image upload only takes in png, jpeg, and jpg extensions. | Image File | Image upload in marker or error thrown if file type is valid or not respectively | PASS |
| 3 | Chrome Error Pointers Test | Check to see if the text box entry throws an error if the text entered is nothing or goes above the character limit. | Text Field | Throws error | PASS |
| 4 | Edge Functionality Test | Check if the drop-down menu can be displayed properly, scrolled through, and store the saved option to be submitted. | Click | Drop-down location in marker | PASS |
| 5 | Edge Bound Fields Test | Check if the image upload only takes in png, jpeg, and jpg extensions. | Image File | Image upload in marker or error thrown if file type is valid or not respectively | PASS |
| 6 | Edge Error Pointers Test | Check to see if the text box entry throws an error if the text entered is nothing or goes above the character limit. | Text Field | Throws error | PASS |

**Code Review:**

The code was written relatively well for the purpose that it served. There were only a few minor corrections to be made and a few simplifications that could have made the program look cleaner overall.

ReportAnIssue.php:

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width,initial-scale=1">

<title>Report an Issue</title>

<script src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js"></script>

<!--Bootstrap CSS Core-->

<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css" integrity="sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T" crossorigin="anonymous">

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js" integrity="sha384-JjSmVgyd0p3pXB1rRibZUAYoIIy6OrQ6VrjIEaFf/nJGzIxFDsf4x0xIM+B07jRM" crossorigin="anonymous"></script>

</head>

<body>

<?php

include 'ReportAnIssueFunctions.php';

?>

<!--\*\*\*\*REMOVE INLINE STYLING LATER!!! -->

<div class="container" style="background-color: gray">

<div class="row">

<div class="col-sm-12">

<h1>Report an Issue</h1>

<p>Fill out the information below to report an issue</p>

</div>

</div>

<div class = "row">

<div class="col-12">

<form action="" method="post" id="issueForm" enctype="multipart/form-data">

<div class="form-group"> <!--Can be simplified if stored as an insert command from php to reduce code written across pages. However, this should be fine.-->

<label for="issueLocation">Location</label>

<select name="issueLocation" class="custom-select mb-3" name="issueLocation">

<option value="BARRY KAYE HALL" selected>BARRY KAYE HALL</option>

<option value="BASEBALL STADIUM">BASEBALL STADIUM</option>

<option value="BEHAVIORAL SCIENCES BUILDING">BEHAVIORAL SCIENCES BUILDING</option>

<option value="BOOKSTORE">BOOKSTORE</option>

<option value="COMPUTER CENTER">COMPUTER CENTER</option>

<option value="COLLEGE FOR DESIGN & SOCIAL INQUIRY">COLLEGE FOR DESIGN & SOCIAL INQUIRY</option>

<option value="COLLEGE OF ARTS & LETTERS">COLLEGE OF ARTS & LETTERS</option>

<option value="COLLEGE OF BUSINESS">COLLEGE OF BUSINESS</option>

<option value="COLLEGE OF EDUCATION">COLLEGE OF EDUCATION</option>

<option value="COLLEGE OF ENGINEERING AND COMPUTER SCIENCE">COLLEGE OF ENGINEERING AND COMPUTER SCIENCE</option>

<option value="COLLEGE OF MEDICINE">COLLEGE OF MEDICINE,</option>

<option value="COLLEGE OF NURSING">COLLEGE OF NURSING</option>

<option value="COLLEGE OF SCIENCE">COLLEGE OF SCIENCE</option>

<option value="CULTURE AND SOCIETY BUILDING">CULTURE AND SOCIETY BUILDING</option>

<option value="DOROTHY F. SCHMIDT ARTS & HUMANITIES">DOROTHY F. SCHMIDT ARTS & HUMANITIES</option>

<option value="DOROTHY F. SCHMIDT PERFORMING ARTS CENTER">DOROTHY F. SCHMIDT PERFORMING ARTS CENTER</option>

<option value="DOROTHY F. SCHMIDT VISUAL ARTS CENTER">DOROTHY F. SCHMIDT VISUAL ARTS CENTER</option>

<option value="ENGINEERING EAST">ENGINEERING EAST</option>

<option value="ENGINEERING WEST">ENGINEERING WEST</option>

<option value="FAU STADIUM">FAU STADIUM</option>

<option value="FLEMING HALL">FLEMING HALL</option>

<option value="GENERAL CLASSROOM - NORTH">GENERAL CLASSROOM - NORTH</option>

<option value="GENERAL CLASSROOM - SOUTH">GENERAL CLASSROOM - SOUTH</option>

<option value="GLADES PARK TOWERS">GLADES PARK TOWERS</option>

<option value="HERITAGE PARK TOWERS">HERITAGE PARK TOWERS</option>

<option value="INDIAN RIVER TOWERS">INDIAN RIVER TOWERS</option>

<option value="INNOVATION VILLAGE APARTMENTS - NORTH">INNOVATION VILLAGE APARTMENTS - NORTH</option>

<option value="INNOVATION VILLAGE APARTMENTS - SOUTH">INNOVATION VILLAGE APARTMENTS - SOUTH</option>

<option value="LIVE OAK PAVILION">LIVE OAK PAVILION</option>

<option value="LIVING ROOM THEATER">LIVING ROOM THEATER</option>

<option value="MEMORY & WELLNESS CENTER">MEMORY & WELLNESS CENTER</option>

<option value="PARKING GARAGE I">PARKING GARAGE I</option>

<option value="PARKING GARAGE II">PARKING GARAGE II</option>

<option value="PARKING GARAGE III">PARKING GARAGE III</option>

<option value="PARLIAMENT HALL">PARLIAMENT HALL</option>

<option value="PHIL SMITH HALL">PHIL SMITH HALL</option>

<option value="RECREATION AND FITNESS CENTER">RECREATION AND FITNESS CENTER</option>

<option value="RITTER ART GALLERY">RITTER ART GALLERY</option>

<option value="S.E. WIMBERLY LIBRARY">S.E. WIMBERLY LIBRARY</option>

<option value="SCIENCE BUILDING">SCIENCE BUILDING</option>

<option value="SOCIAL SCIENCE BUILDING">SOCIAL SCIENCE BUILDING</option>

<option value="SOFTBALL STADIUM">SOFTBALL STADIUM</option>

<option value="STUDENT HEALTH SERVICES">STUDENT HEALTH SERVICES</option>

<option value="STUDENT HOUSING OFFICES">STUDENT HOUSING OFFICES</option>

<option value="STUDENT SERVICES & CAFETERIA">STUDENT SERVICES & CAFETERIA</option>

<option value="STUDENT SUPPORT SERVICES">STUDENT SUPPORT SERVICES</option>

<option value="STUDENT UNION">STUDENT UNION</option>

<option value="UNIVERSITY VILLAGE STUDENT APARTMENTS">UNIVERSITY VILLAGE STUDENT APARTMENTS</option>

</select>

</div>

<div class="form-group">

<label for="issueDescription">Description:</label>

<input type="text" class="form-control" id="issueDescription" name="issueDescription" placeholder="Describe the issue...">

</div>

<div class="custom-file">

<input type="file" name="issueImage" class="custom-file-input" id="issueImage">

<label class="custom-file-label" for="issueImage">Choose Image...</label>

</div>

<div class="form-group">

<button type="submit" name="issueSubmit" class="btn btn-primary">Submit</button>

</div>

</form>

<button type="submit" name="home" id="home" class="btn btn-success">Home</button>

</div>

</div>

</div>

<?php

if (isset($\_POST['issueSubmit']))

{

$MAX = 100000;

if(($\_POST['issueDescription']!="")) //If-Else only checks if description is not null only no overflow check

{

$file = $\_FILES['issueImage'];

$fileName = $\_FILES['issueImage']['name'];

$fileTmpName = $\_FILES['issueImage']['tmp\_name'];

$fileSize = $\_FILES['issueImage']['size'];

$fileError = $\_FILES['issueImage']['error'];

$fileType = $\_FILES['issueImage']['type'];

$fileExt = explode('.',$fileName);

$fileActualExt = strtolower(end($fileExt));

$allowed = array('jpg','jpeg','png');

if(in\_array($fileActualExt,$allowed)){ //Checks file and moves it to correct location is uploaded

if($fileError===0){

if($fileSize < $MAX){

$fileNameNew = uniqid('',true).".".$fileActualExt;

$fileDestination = 'IssuesImages/'.$fileNameNew;

move\_uploaded\_file($fileTmpName,$fileDestination);

$issueLocation = $\_POST['issueLocation'];

$issueDescription = $\_POST['issueDescription'];

InsertIssue($issueLocation,$issueDescription,$fileDestination);

}else{

echo("Image is too large! Must be less than 10Mb!");

}

}else{

echo("Error.. try again"); //Very unspecific, what would be the error?

}

}else{

echo("File type not accepted!");

}

}

else{echo("Please fill out the missing fields!");}

}

?>

<script>

// The name of the file appears on the upload field

$(".custom-file-input").on("change", function() {

var fileName = $(this).val().split("\\").pop();

$(this).siblings(".custom-file-label").addClass("selected").html(fileName);

});

document.getElementById("home").onclick = function () {

location.href = "http://lamp.cse.fau.edu/~cen4010fal19\_g04/CampusSnapshotBeta/CampusSnapshot/index.php";

};

</script>

</body>

</html>

ReportAnIssueFunctions.php

<?php

function InsertIssue($issueLocation,$issueDescription,$fileDestination){

include '../Login/Login.php';

$db = new mysqli($dbhost, $dbuser, $dbpass, $dbname);

/\*Tests that the database connects\*/

if($db->connect\_errno > 0) {die('Unable to connect to database [' . $db->connect\_error . ']');}

$result = $db->query("INSERT INTO IssuesOnCampus(issueLocation,issueDescription,issueImagePath) VALUES ('$issueLocation','$issueDescription','$fileDestination')");

if($result)

{

echo "Success!";

}

$db->close();

}

?>

**Security Check:**

We need to protect account information and to make sure that we are encrypting them. The only major assets would be a user’s account access as to avoid stealing another person’s account of mal use of the product.

Currently, the password is encrypted via SQL’s DES encryption and decryption protocols. We are currently working on a more secure encryption algorithm that would be hashed on password creation/input and then stored within the database rather than have the encryption and decryption go fully on in the database. This would occur right after login but before the checks are done to make sure it is valid.

For input data validation, we check to make sure there is a username and password being compared against. However, this system is still a temporary placeholder as the encryption is being implemented so there is only basic validation within the code to make sure that the username exists somewhere.

function UserLogin($username){

include '../Login/Login.php';

$db = new mysqli($dbhost, $dbuser, $dbpass, $dbname);

/\*Tests that the database connects\*/

if($db->connect\_errno > 0) {die('Unable to connect to database [' . $db->connect\_error . ']');}

$reorder\_query="SELECT \* FROM Users ORDER BY studentID";

$reorder\_result = $db->query($reorder\_query);

$userNameExists = 0;

while(($row = $reorder\_result->fetch\_assoc())) {

$temp = $row['username'];

if($username==$temp)

{

echo("You are logged in!");

}else{

echo ("Username does not exist. Create an account to sign-in!");

}

}

$db->close();

}

**Adherence to original Non-Functional Specs:**

1. Users should be able to create an account if they do not already have one. - DONE
   1. A valid email address is required to create an account. – ON TRACK
   2. A username must be at least 6 characters long. - DONE
   3. A password must be at least 8 characters long. - DONE
2. Users will be identified by their campus provided email. – ON TRACK
   1. Any duplicate emails shall not be permitted to create an account. – ON TRACK
3. Users should be able to attach a photo to their reporting and creation of marker. - DONE
   1. The picture must be a PNG or JPG file format, all other types will not be taken. - DONE
   2. The picture must be at least 480p so the image can have decent quality. - DONE
4. Users should be able to comment on existing markers. – ON TRACK
   1. Comments will be limited to 100 characters. – ON TRACK
   2. Students can only post one comment per marker and zero on markers created by that student. – ON TRACK
5. Users should be able to report inappropriate posts. – ON TRACK
   1. Students that report inappropriate posts must be reviewed by Administration. – ON TRACK
6. Administration should be able to delete/modify markers and comments. – ON TRACK/DONE
   1. Any deleted markers/comments will be completely removed from the web system. – ON TRACK
7. Administration should be able to provide status updates of reported issues. – ON TRACK
   1. Status updates will either be in the form of comments or its own issue resolver system. – ISSUE (The system of comments have not been created yet)
   2. If in the form of comments, previous status updates will be deleted from the system. – ISSUE (The system has not been implemented to see if this is a necessary feature)
8. Students should be limited to a specific number of posts per day to avoid spamming the system. - DONE
9. Users can only have a single account. - DONE
10. Posts should indicate whether it has been modified by Student or Administration. – ON TRACK
11. Users should be able to use the system without instructions. - DONE
12. JavaScript, SQL, HTML, and CSS will be the main focus languages used to develop *Campus Snapshots*. - DONE
13. Markers can be sorted/displayed by location on campus. – ON TRACK