CEN 4010: Principles of Software Engineering

Fall 2019

**Milestone 4**

**Beta Launch and Final Project Reviews**

**Campus Live**

**Code Nxt**

Team 3

Team Members

|  |  |
| --- | --- |
| Sharon Freystaetter | [sfreystaette2015@fau.edu](mailto:sfreystaette2015@fau.edu) |
| Jeff Campbell | [jeffcampbell2016@fau.edu](mailto:jeffcampbell2016@fau.edu) |
| Francois Joseph | [josephf2014@fau.edu](mailto:josephf2014@fau.edu) |
| Kevin Lewitzke | [klewitzke2017@fau.edu](mailto:klewitzke2017@fau.edu) |

Document Versions

|  |  |  |
| --- | --- | --- |
| **No.** | **Revision** | **Date** |
| 1.0 | Initial version | 09/23/2019 |
| 2.0 | Detailed requirements, architecture, and vertical software prototype | 11/04/2019 |
| 3.0 | Usability/QA test plans, code review | 11/18/2019 |

# 

# Product Summary

Campus Live is a new, innovative method of easy communication between faculty and students on a large college campus. No longer will there be a barrier separating students from administration. Any issues on campus can be reported by students on Campus Live, allowing for action to be taken without any delay and students to feel heard. With the implementation of Campus Live, students are given a voice to let faculty know exactly what is happening on campus and administration is given an outlet to let all students know about upcoming events.

Major functions of Campus Live include:

* Ability for any student or faculty member to report maintenance problems and upcoming events on campus
* Pertinent details for each problem or event are captured and tracked, including: title, detailed description, date and time, and status
* Students and faculty are able to review reported problems and browse upcoming events. Search, sort and filter tools are available.
* Photos may be uploaded for each problem and event
* Comments may be added to each problem and event
* Maintenance problem and system reports may be generated by authorized users

The application may be accessed at this location:

[http://lamp.cse.fau.edu/~cen4010fal19\_g03](http://lamp.cse.fau.edu/~cen4010fal19_g03/view_problems.php)

Application demo user names:  
fjoseph2014, jcampbell2016, klewitzke2017, sfreystaette2015

The password for all demo user names is *password*

# Usability Test Plan

The major function to be tested for usability is the functionality to report a new maintenance problem on campus.

## Test Objectives

The objectives of the usability test plan are to ensure that the functionality to report a new maintenance problem is intuitive for inexperienced users within the Campus Live application. The user should be able to successfully enter a new maintenance problem on their own, without any assistance. The user should feel comfortable performing the functionality within the system, and confident that their action was completed successfully. If there are usability aspects of the system which have not been considered during design, it is expected that they will be revealed during usability testing.

## 

## Test Plan

|  |  |
| --- | --- |
| **System setup:** | Windows 10 operating system  Chrome browser |
| **Starting point:** | Login page |
| **Task to be accomplished:** | Report a maintenance issue on campus |
| **Intended user:** | Any student or faculty user |
| **Success criteria:** | Maintenance issue has been successfully reported |
| **URL to be tested:** | <http://lamp.cse.fau.edu/~cen4010fal19_g03/report_problem.php> |
| **Steps:** | 1. User enters his or her credentials and clicks the “Log In” button 2. User navigates to the “Report a Campus Problem” page 3. User completes all required fields on the page 4. User successfully submits the form |
| **Notes:** |  |

## Questionnaire form

1. It was easy to navigate to the intended location within the application

Strongly Agree Agree Neutral Disagree Strongly Disagree

1. The destination page form was easy to understand and use.

Strongly Agree Agree Neutral Disagree Strongly Disagree

1. I am interested in this product and would use it during my time on campus.

Strongly Agree Agree Neutral Disagree Strongly Disagree

# QA Test Plan

## Test Objectives

The goal for the quality assurance testing is to confirm that all software functionalities of Campus Live perform as it has been intended. Testing will confirm that the website Campus Live is a product to be used as a form of communication between students and faculty of a campus. Product functions provided in the product will be validated in this testing process. The test plan will also provide verification that all functionalities provide accurate outputs to the users. All functions will be tested against multiple input cases in order to provide a product that meets the requirements of the customer and provides little to no errors during use.

## Hardware and Software Setup

The hardware setup for quality assurance testing will include accessing the application from both a desktop or laptop computer, as well as a mobile device or tablet. The computer and mobile device or tablet will have an internet connection. The application will be tested using the Apple Safari, Mozilla Firefox, Google Chrome, and Microsoft Edge web browsers on both devices. The application will be hosted on the FAU LAMP stack and accessed using a standard internet connection.

## Tested Features

The following features will be tested:

1. A user will only be able to log into the website as long as they provide the correct username and password pairing.
2. All users will be able to create a new entry on the Campus Live website.
3. All users will be able to view all reported problems and events. Each entry is classified as either issue or event.
4. Each new entry will not submit unless the required information (title, location, and description) is inputted. Once entry is submitted, a unique ID will be created for it.
5. The search functionality will provide desired results for inputs without case sensitivity and partial inputs.
6. Only Faculty / Staff, Maintenance and System Administrator users are able to edit existing entries.
7. No existing entries older than 30 days can be reopened by any user. A new entry will be required.
8. Maintenance users and System Administrators will be able to generate on-demand reports.
9. All users will be able to view their own entries, comments and photos in one place.

## Test Cases

### Test Case #1: User Login

Tester will login on the Campus Live website with correct credentials (a valid username and password pair), incorrect credentials (both username and password are not in the system), and a set of credentials that is half correct (username is correct and password is incorrect, username is correct and password is incorrect, or both username and password are in the system but they are not pairs of each other). The test will pass if the tester is able to access the system only after entering correct credentials. The use of any partially correct or incorrect credentials returned an error message.

### Test Case #2: Submit New Entry

Tester will submit a new entry (reporting a problem or creating a new event) on the Campus Live website. The application shall not allow a form to be submitted unless all the required fields have been completed. The form shall be submitted multiple times, each time with one piece of required data missing, to ensure an error message is displayed to the user. The form shall be submitted both with and without a photo. If the form is submitted with a photo, the photo shall successfully be uploaded to the application. The user shall receive a confirmation message after each successful submission, and the details of the form must be captured accurately by the application for the test to be successful.

### Test Case #3: View and Filter Previous Entries

Tester will log in as a student, and filter through the reported issues. The test shall be completed using all possible filtering combinations. Case insensitivity on search fields shall be tested by entering search terms in a different case than the stored data, and ensuring results are returned. The test shall also be completed using data that is expected to return no results. Accessing a results page with no filters applied shall return all records.

## Results

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test #** | **Test Title** | **Test Description** | **Test Input** | **Expected Correct Output** | **Test Results** |
| 1 | User Login | Test the user login function of the Campus Live software. | Username:  sfreystaette2015  password:123 | Error message: ‘Incorrect username/password’ | Error message: ‘Incorrect username/password’ |
| 2 | Submit New Entry | Test the user’s ability to submit a new entry to the database. | Empty title input  Location: MLC | Error message: ‘Location required’ | Error message: ‘Location required’ |
| 3 | View and FIlter Previous Entries | Test the user’s ability to find past entries using a filtering system | Location filter: ‘LIB’ | Two outputs with the location of library | Two outputs with the location of library |

### Test Case #1: User Login

When the tester ran this test case they were only able to log in using their correct credentials. If the tester put in the username with the wrong password, the software would give an error message ‘Incorrect username/password.’ If the tester only gave an input for the username and no password input, the software would give the error ‘Password required.’ Based on these results, the software for Campus Live passed QA test case 1.

### Test Case #2: Submit New Entry

When the tester rand this test case they were only able to report a problem when both a description and location were input. If the tester only gave the title and no location, the software would produce the error message ‘Location is required.’ If the tester only gave the location and no description, the software would produce the error message ‘Description required.’ Based on results being produced as intended, the campus live software passed the QA test case 2.

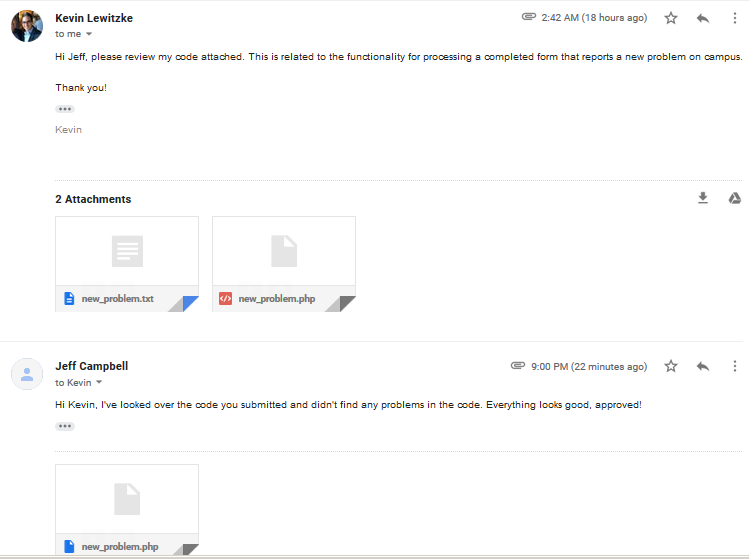
### Test Case #3: View and Filter Previous Entries

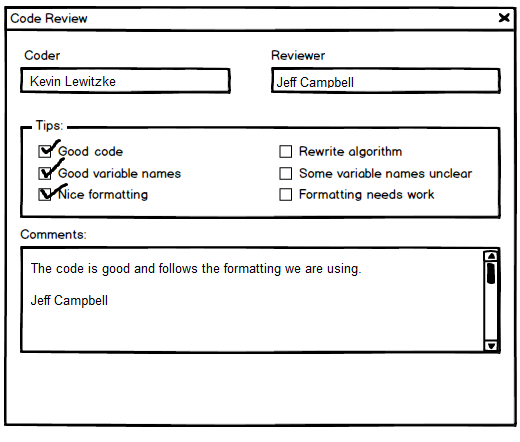
The tester was able to filter through all past entries and produce the expected list of results from the database. In the database there were previously two inputted reported problems located at the library. The same two entries showed when the tester searched through the results using the location filter: ‘Library’, ‘LIB’, and ‘libr’. If the tester gave an input that was not in the database, such as the reported by filter ‘John Smith’, the Campus Live website would produce an empty list. The test results for test case 3 were produced as expected.

# 

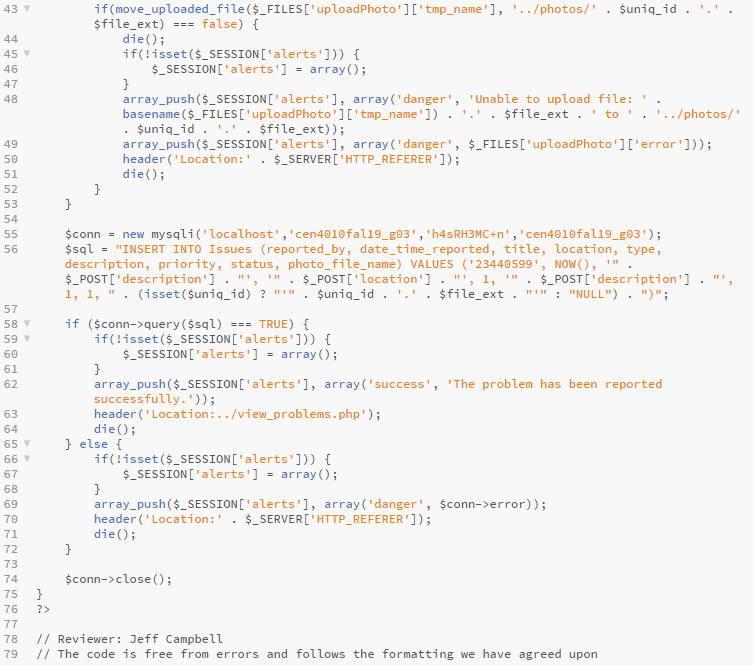
# Code Review

The coding style used in the application is 1TBS or “One True Brace Style”, a variant of the K&R (Kernighan and Ritchie) style.









# 

# Best Practices for Security

Assets to protect include:

1. File system
2. Database
3. Student and faculty details
4. Passwords

All passwords are hashed using the PHP password\_hash function, and the hashes are stored in VARCHAR fields of the MySQL database. The password\_hash function is a strong, one-way hashing algorithm. The algorithm specified in the application is *PASSWORD\_DEFAULT*, which is currently the bcrypt algorithm, but is designed to change over time as new and stronger algorithms are added to PHP. Specifying *PASSWORD\_DEFAULT* as the hashing algorithm ensures that the latest algorithms are automatically used by the application as PHP is upgraded in the future.

The password\_hash function automatically salts passwords at the time they are hashed. It also allows the algorithmic cost to be increased. Passwords entered by the user verified against the stored hashes using the PHP password\_verify function.

Input data is validated both client-side and server-side. Client-side validation is performed using HTML attributes. For example, HTML will automatically prevent a form from being submitted if an input with the *required* attribute is empty. Other client-side validation includes ensuring the input data format matches the input type. For example, HTML will not permit alpha-characters (letters) to be entered into a date field.

Client-side validation is designed to guide the user, but it is an ineffective security practice. Server-side validation is performed using PHP. All form attributes (required fields, minimum/maximum length, input data types, etc.) are checked again during processing on the server, and any form is rejected if invalid data has been entered. The user is returned to the form and an error message is displayed. All form data is cleansed to prevent SQL injection or malicious files from being uploaded to the system.

# Adherence to Original Non-Functional Specs

|  |  |  |
| --- | --- | --- |
| **Req #** | **Requirement** |  |
| 1 | Campus Live shall be compatible with the current, non-beta versions of Mozilla Firefox, Safari, Google Chrome, and Microsoft Edge web browsers, including the versions of these browsers designed for mobile and tablet devices. | On Track |
| 2 | The user’s operating system shall not be relevant in the operation or performance of the web application. | Done |
| 3 | All users shall be able to report an issue, look up existing issues, create an event, and look up existing events, without any prior training. | On Track |
| 4 | The system should have the latest up-to-date security features for encrypting and securing all users information and data. | On Track |
| 5 | The system should comply with all outside legal and regulatory agencies to the best of its ability. | Done |