**CEN4010 Principles of Software Engineering Summer 2018**

**Milestone 1: Project Proposal and High-level description**

DAFJ Ninjas, Owl-iView

Team number 1

Daniel Gross

Dgross13@fau.edu

Andrew Newland

Anewland2015@fau.edu

Filipe Catarcione

Fcatarcione2012@fau.edu

Jonathan Ortiz Collazo

Jortizcollaz2016@fau.edu

**Documentation**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Author** | **Description** | **Date** |
| 1.0 | Daniel Gross | Document Creation | 6-18-2018 |

Table of Contents

1 Executive summary 3

2 Competitive Analysis 4

3 Data definition 5

**4 OVERVIEW, SCENARIOS, USE CASES………………….……………………………………………….5**

**5 HIGH LEVEL FUNCTIONAL REQUIREMENTS..………………………………………………………….5**

**6 NON FUNCTIONAL REQUIREMENTS..………...………………………………………………………….5**

**7 HIGH LEVEL SYSTEM ARCHITECTURE………………………………………………………………….6**

**8 TEAM AND CHECKLIST………………….………………………………………………………………….6**

# Executive Summary

Florida Atlantic University (FAU) requires a web system be assembled to deliver a snapshot of FAU’s college grounds and campus happenings. Specifically, we will permit society to observe an instance of the campus in real time, which provides information about events happening around campus and details the reported issues that need attention on the campus grounds.

DAJF Ninjas will build a custom web system, called “Owl-iView” with our team of front end and backend developers. We will develop our project using the C# language and asp.net core as our framework. The needs for Owl-iView are:

* Issues reported in a web-form filled out with either photos, text, building code, severity level, or notes. Requires University administrators to check system for any reports of issues to fix.
* Secure login page with a new account registration link and present the privacy policies to users. Data to store securely in the database on the server.
* Ease of use, intuitive, attractive, and media rich system. View status of reported, in progress, and completed issues. Site must be in English, compatible on major browsers, searchable by major search engines. View and comment on existing posts and make a report thread.

System constraints:

* System is usability is limited to the English language, while performance and speed depends on the type of browser used.
* System storage is limited to the allocated space on the server.
* System usability is limited reports cannot be exported.
* System security requires just a username and password.
* System accessibility is limited to account holders only.

Top 3 requirements or design objectives (in order of importance) is:

* Deliver product to customer specification of creating an easy and attractive web system for users to report an issue via an online form and.
* Users can view a report and is able to distinguish the issues and events clearly.
* Secure login page.

The strategy initially is to build an attractive site with the imposed constraints. We will design and develop the site to satisfy all of the needs and objectives. The design will be evaluated by the customer after providing releases of the site in order to gain feedback. Users will use the site’s upload and reporting feature. Key features are that our framework creates a dynamic web page since asp and html work well together, better application security with asp’s built in windows authentication, faster because asp is a server side technology where the code is executed before sending to browser. A novelty of out project is the reporting feature because it shows the good and the bad on campus grounds. The values of the project are that it will improve the quality of life for students with the ability to communicate issues fast on campus.

# Competitive Analysis

|  |  |
| --- | --- |
| Owl-iView | Competitors |
| Post an issue with picture or text | Post issue with text |
| Commenting on posts, like and threads | Comment. |
| View and append a severity to a post | View posts |
| View reported, in progress, and completed issues on campus | View reported issue |
| Cross Platform (Linux, Windows, Apple) | Not cross platform |
| Attractive and dynamic | Non attractive. |

To summarize, our planned advantages compared to what we have will give us an edge over competitors because our site will have better performance, ease of use, and very attractive. The system will prove useful because of the real time snapshot and event features.

# Data Definition

# Database=System for storing all place for data needed

SQL = Structured Query Language to locate specific and to the database for data.

Entity Framework (EF) Core = EF Core can serve as an object-relational mapper (O/RM), enabling .NET developers to work with a database using .NET objects, and eliminating the need for most of the data-access code they usually need to write.

# University administrators=University Campus grounds maintenance and management.

# Users= People who posts on the website.

# Overview, Scenarios and Use Cases

FAU clubs or services may use this web system to upload photos for the system administrators to view and fix. The web system is beneficial because it is fast and provides a direct report to the person in need.

Students may upload photos of recent events or club activities for visitors and users to see.

From an end user perspective, users will need to fill out a form online to report a good or potentially broken service that requires fixing.

Users will be uploading a report of an issue they find fit to be directed to maintenance for fixing. Users and administrators on campus may receive a report for campus ground happenings and events.

# High Level Functional Requirements

High level functionalities are:

* Create an Events page to display a list of posts related to Events.
* Create an issues page to display a list of posts related to Issues only,
* Ability to create a post.
* Ability to add comments to a post.
* Ability to upload an image to a post.
* Allow users to create a post from any browser across any platform.

# Non Functional Requirements

System performance and speed will be an advantage with asp.net core. Storage will be on an Azure server. System usability is limited reports cannot be exported. The web system protects user accounts by a password. System accessibility is great with cross platform capabilities. By using the Azure services, we can expect a heavy load of data easily.

# High Level System Architecture

Our main software product includes visual studios 2017 for development and Azure to host our web system. Also, asp.net core and C# for our framework provide fast user interface and fast storage of data in the database located in Azure. We will use tools and our language will be in C#. The framework support modern browsers such as Chrome, Mozilla, and Safari. For any external code we plan to use, its source and license will be listed.

# Team and Check List

Group 1 is comprised of Daniel Gross, Product Owner and front end developer, Andrew Newman, Front/Back end developer and Scrum Master, Filipe Catarcione, lead developer, and Jonathan Ortiz Collazo, lead developer.

a) Team decided on basic means of communications DONE

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

c) Front and back end team leads chosen DONE

d) Github master chosen DONE

e) Team ready and able to use the chosen back and front-end frameworks DONE

f) Skills of each team member defined and known to all DONE

g) Team lead ensured that all team members read the final M1 and agree/understand it before submission DONE