**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 4

2 Competitive Analysis 4

3 Data definition 4

**4 OVERVIEW, SCENARIOS, USE CASES………………….……………………………………………….6**

**5 HIGH LEVEL FUNCTIONAL REQUIREMENTS..………………………………………………………….6**

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

**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 create this using a framework in asp.net core and the language in C#. The needs for our “Owl-iView” are:

* Visitors; People to report issues on a form from the system, which contains fields for photos, text, building codes, severity level, and other notes. Requires University administrators to check system for any reports of issues to fix.
* Security; Secure login page with a new account registration link, and communicate privacy policies to user. Data will be stored in the database on the server.
* User Friendly; 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:

* Easy and attractive way for users to report an issue via an online form.
* 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.

Make it an executive summary -- think of answering the question of why you develop this project and target at what market sectors. Assign a product name to your project. This executive summary should be readable to a general audience who is not a computer science specialist. The executive summary is also used to advertise and promote your project.

# Competitive Analysis

|  |  |
| --- | --- |
| Owl-iView | Competitors |
| Post an issue with picture or text | Post 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. |
| View events from campus organizations | View events. |
| 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, easy to use, and attractive. The system will prove useful because of the real time snapshot feature.

# Data Definition

# Our search feature will make it possible to search for your post by id number and sort in ascending or descending order.

# This section serves as the “dictionary” of your document. It defines main terms, data structures and “items” or “entities” at high or logical (not implementation) level (e.g. name, meaning, usage, and NOT how the data is stored in memory) so it is easier to refer to them in the document. Focus on key terms (main data elements, actors, types of users etc.) specific for your application and not on general well know terms. These terms and their names must be used consistently from then on in all documents, user interface, in naming software components and database elements etc. In later milestones, you will add more implementation details for each item. You will later expand this section with more details.

# Overview, Scenarios and Use Cases

This section describes the project overview (in much more details) and likelihood usage scenarios of your product from end users’ perspectives. Focus only on main use cases. Simple text format is OK and preferable – tell us a story about who and how is the application used. Focus on WHAT users do, their skill level, not on HOW the system is implemented. You can expand use cases provided in high level document in future milestones.

# High Level Functional Requirements

This refers to the high-level functionality that you plan to develop to the best of your knowledge at this point. Focus on WHAT and not HOW. Keep the users in mind. Develop these functions to be consistent with use cases and requirements above. Number each requirement and use these numbers consistently from now on. For each functionality use 1-5 line description.

# Non Functional Requirements

For example, performance, usability, accessibility, expected load, security requirements, storage, availability, fault tolerance etc. Number each. When possible, try to quantify these quality attributes.

# High Level System Architecture

Our main software product includes visual studios 2017 to use asp.net framework tools, languages and systems to be used, list of core APIs available at this point, supported browsers etc.

You also have to decide on which frameworks you will use if any. These provide both user interface, as well as cross-platform and cross browser layout/css. All external code you plan to use must be listed along with their license.

# Team and Check List

List student group names, name of Scrum master, product owner and initial roles for each member

For each item below you must answer with only one of the following: DONE, ON TRACK (meaning it will be done on time, and no issues perceived) or ISSUE (you have some problems, and then define what is the problem with 1-3 lines). Reflect these items in your Trello project space:

a) Team decided on basic means of communications

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

c) Front and back end team leads chosen

d) Github master chosen

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

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

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