

Group 5 Requirement Documentation

Index

1. Introduction	2
1.1 Purpose	2
1.2 Scope	2
1.3 References	2
1.4 Overview	2
2. Project Description	3
2.1 System Overview	3
2.2 Client Characteristics	3
2.3 User Characteristics	3
2.4 Product Functions	4
3. Requirements	4
4. Non-Requirements	4
5. Assumptions	5
6. Appendices	6
6.1 Glossary	6
6.2 Author Information	7
6.3 Additional Documents	7

1. Introduction

1.1 Purpose

The purpose of this document is to create a simplistic yet modern website that will merge all of the content on all three of the client's current websites as well as allow assignment submittals. This project will be used primarily by the client for personal blog style postings and posting courses material. It will also be accessible for students looking to access information regarding their courses and submit assignments and followers of the client's blog postings

1.2 Scope

The purpose of this website to make posting blog posts and course material significantly easier for the client. The client is Gusty Cooper, an adjunct professor at the University of Mary Washington not to mention an avid bicyclist too. Currently, he has three separate websites performing all these functionalities which have become onerous for him to continue to do so. This website will not only be able to host all the information needed for his courses for students to access but will also have a separate space for his personal blog postings. At the end of this project, we hope to have a provided a website that smooths the transition of operating between three separate sites to solely the one that we've built for him.

1.3 References

1. 430 Requirements Documentation
2. 430 Writing Guide

1.4 Overview

The rest of this document is broken down into five main parts:

Project Description - This section gives a summary of the client's motivation, user and client interactions with the system and the necessary background information on the system itself.

Requirements - This section gives a list of desired behaviors for the system and will be used to determine whether the project was a success.

Non-Requirements - This section gives a list of technical requirements or a limiting factor on the overall behaviors of the system.

Assumptions - This section will list the majority of the assumptions that can be made about the system.

Appendices - This section contains a glossary that defines the terms specific to this project. The sections and their page numbers can be found in the Index on page 1 of this document.

2. Project Description

2.1 System Overview

The website maintains information on current and past course information, The website will only support the user privilege for our client to log in and add new posts or new course information. The system will be available to any potential student and user on the internet. Gusty will be able to access restricted areas and functions of the site through an administrator username and password. The system will be a merger of content from the client's current web pages. They will be replaced by this new system, and feature a significant design and logical layout change.

2.2 Client Characteristics

The client, Gusty Cooper, is an adjunct professor at UMW where he regularly teaches courses 110, 220, 240, 284,305. For each of these classes, Gusty has lecture notes, relevant course materials, and videos to provide a constructive learning environment. All this information is available on his site gustycooper.github.io. Gusty is also a devoted cyclist and blogger. His other two sites, gusty.bike, and gustycooper.org both feature information about his races, bikes and other adventures. The client asked us to create a new website because he has lost the functionality of accepting assignments on his courses site. Also, his other two sites are poorly organized and not easily navigable.

2.3 User Characteristics

Users of this site should be able to access course information on specific classes that Gusty has taught or teaches. The user will only have access to the information posted by our client on the site. They will also be able to submit an assignment through a portal. They should also be able to view blog posts uploaded by the client. The client would like to the commenting abilities turned off so the user will be able to comment or "like" a post or have the ability to interact with the post in any way. Expected users of this site will students, fellow cyclists, followers of the client's bike collection and tournaments as well as Gusty enthusiasts.

2.4 Product Functions

The primary functions of the new system will be to provide users with direct access to current course information. It will also allow for differential access to certain web pages if the user is an administrator. The system will allow users viewing permissions to the site's personal posts and media. The system will have Mooshak integrated into it to allow students and users the ability to submit assignments. Mooshak will automatically grade these assignments and make the grades accessible to the users and clients. Other desired features of the new website will be a consistent design layout through the website. There should also be a full-text search for web pages as well as a search bar to allow for easier page navigation.

3. Requirements:

1. Basic:
 - a. Website must have a public hosting and accessible to end-users
 - b. Website must use a modern web-stack technology (example: React)
2. Client Log-in:
 - a. Client must be able to log-in
 - b. Client must be able to make changes to the website
3. Student Log-in:
 - a. In order for
4. Functionality:
 - a. Website must have a dedicated section for students to upload files
 - b. Implement Mooshak:
 - c. Mooshak form will allow submissions to include their name
 1. Name passed on to client for grading purposes
 - d. Mooshak will grade the results automatically
 - e. Website must incorporate a "slide show" style display of images.

4. Non-Requirements

In the case of this project, the non-requirements are perhaps just as important as the requirements themselves. We've had the opportunity to dissect the previous websites that have been created for our client and discuss the pros and cons of each one. With this information, we are able to weed out any functional and non-functional requirements that are not needed in our final product. These requirements are, but not limited to:

NR 1: The students should not be able to comment on nor append any posting that the client creates within the confines of the websites. The client has found that the Wordpress style of website creation clashes with their vision.

NR 2: The client does not want the system that will be used to be overly complicated. The client has expressed their interest in being able to upkeep the system on their own whenever an update is needed.

NR 3: The client wants to use their already in use domain, Gusty.bike which is being hosted with AWS, to place their new website on. That being said, the client has explained that he is not tied to AWS for the hosting.

These non-requirements paint a better picture of how the website will feel and look, which hopefully will be used in development ahead to more correctly represent the idea that our client has in mind.

5. Assumptions

The major ambiguity that is to be faced in the creation of this product is the choice of programming language to use in the front-end and back-end development. Upon researching the current modern options for back-end languages, a few assumptions can be made about each one.

The choice that seemed to appear the most often was node.js, which boasts its use of asynchronous programming makes it very memory efficient. Node.js is a javascript runtime, which many interactive websites use for their front-end, making it a very modern and convenient choice to build this project on. The cons of using node.js would be making efficient use of time; javascript happens to be the more uncommon language used in the computer science department, and some of the time would have to be spent researching the functionalities of the language. This would mean that less time can be used on implementing features, and the final product may reflect that.

Django was a back-end that was discussed; a python based web server that allows for simple website creation at a quick pace. Since python is such a user friendly language, it would be no problem to pick and hack out a website, which seems to have been the case in previous iterations of this project. The main problem faced with using Django is the integration of the Mooshak application that is an imperative requirement for the website. This may prove problematic in the late stage of development if Mooshak is unable to be fully integrated within the confines of python, which will be something to seriously consider when deciding on a language choice.

Another option to consider would be using PHP as the back-end to an HTML based front-end. PHP is often seen as dated, but its solid structure and simplicity makes it fast and simple to use and update. HTML is also a common yet efficient language that can create appealing looking websites and with a bit of creativity, can be made to look very professional. The cons of using PHP is that it is a preprocessing language, meaning all calculations are done before the webpage is loaded. This makes events like altering information and animations a bit clunky.

SQL or mySQL is imperative in the creation of databases for holding information such as grades and security. SQL can be used to create a simple username and password system. This way, the client can have every student listed and delegated to a specified class, if multiple classes exist. The client would also be able to login as an administrator to alter the site such as adding posts, uploading files, etc. It can be said that javascript also has methods of creating a username and login system, which may prove to be a far simpler solution if that route is to be taken.

Currently, it is undecided which method would be considered the best choice given the project, but with theses assumptions the pros and cons can be weighed by the client on which choices would best suit their vision of the final product.

6. Appendices

6.1 Glossary

-Front-End: The part of a website that a user is able to see. This can include anything from text, pictures, design layout, color, etc. the more common languages used in front end development include HTML, PHP, CSS, and Javascript.

-Back-End: The server side of website development. This includes the inner workings of the website like databases, security, and overall structure of the platform. Popular languages used in Back-End development are PHP, Python, SQL, mySQL, and Java (node.js).

-Asynchronous Programming: Refers to the method in which a program handles the jobs assigned to it. In this case, instead of waiting for a function to close out completely, functions can be run in between each other, making a more efficient program.

-PHP - Hypertext Preprocessor

-HTML: Hypertext Markup Language

-CSS: Custom Style Sheet

-SQL: Structured Query Language, used in the creation of databases and querying information from said databases.

-UMW: University of Mary Washington

6.2 Author Information

Michael Hudson: Handled the Non-Requirements, the Assumptions, and all of the Appendices. Also created the index and outline of the documentation.

Ebony Clarke: Created the Introduction and the Project Description of the documents.

David Velasquez: Handled the Requirements portion of the document.

6.3 Additional Documents

Front-end and back-end development

<https://www.pluralsight.com/blog/film-games/whats-difference-front-end-back-end>

<https://www.invensis.net/blog/it/difference-between-front-end-and-back-end-development/>

Node.js

<https://nodejs.org/en/about/>

https://www.w3schools.com/nodejs/nodejs_intro.asp

Django

<https://www.djangoproject.com/>

SQL

<https://www.mysql.com/why-mysql/>