**Project Plan**

**for**

**Gusty.bike**

**Prepared by**

**Jessica Spranger**

**Evan Shipman**

**Jacqueline Coates**

**2/19/2019**

**Table of Contents**

[**1. Introduction**](#_5434tlmaxjci) **3**

[1.1 Purpose](#_nyskskkeregl) 3

[1.2 Scope](#_5xmv2jmytry1) 3

[1.3 References](#_9636ryid5m8c) 3

[1.4 Overview of the remainder of the document](#_fx86ajfybmrk) 3

[**2. Project Description**](#_9hrxqi1wjnx2) **3**

[2.1 System overview](#_24u50nhscfg9) 3

[2.2 Client characteristics](#_fz6mvqfhlf5n) 4

[2.3 User characteristics](#_78eo2gyle8e1) 4

[2.4 Functional requirements](#_z5pl63f7ivt5) 5

[2.4.1 - Blog-posting](#_vvr494srbqxr) 5

[2.4.2 - Blog Search](#_lxz2hajcmkgo) 5

[2.4.3 - Ease of Use & Resources](#_3w2d59hyjld8) 5

[2.4.4 - Admin and User Logins](#_khqgxgedik09) 5

[2.4.5 - Contest Hosting Service](#_cmi33n43k0u7) 6

[2.4.6 - Contest Student Submission](#_lmxxyebj2exf) 6

[2.4.7 - Add Site Content Page](#_xx32a4gfltma) 6

[2.4.8 - Image Slideshow](#_on39qzs9cgz7) 6

[2.5 General constraints](#_u31qpbdrtcod) 6

[2.5.1 - Ease of use](#_rjyd3sgk9v2c) 6

[2.5.2 - Cloud hosted](#_xgmc4d3j0iim) 7

[2.5.3 - Budget limit](#_xtrig6i3ji3e) 7

[2.5.4 - Logo](#_qcukylemuaga) 7

[2.5.5 - User manual](#_hn13tey39dxw) 7

[**3. Project Schedule**](#_2h4i50jv2zqc) **8**

[3.1 Approach](#_ywq35et6pwum) 9

[3.2 Milestones and Deliverables](#_c7tc15ux16s9) 9

[3.2.0 - Set Up Building Environment - Milestone](#_1ddcnbhm62bu) 10

[3.2.1 - Install Wordpress - Milestone](#_xxb5nyddz122) 10

[3.2.2 - Blog Design - Deliverable](#_8nrwa4j48gxp) 10

[3.2.3 - Blog Posting Feature - Deliverable](#_cd5fsr21w3xg) 10

[3.2.4 - Blog Editing Feature - Deliverable](#_gz759lfdtks6) 10

[3.2.5 - Blog Searching - Deliverable](#_se50h1lq0foq) 10

[3.2.6 - Login - Deliverable](#_4mavgqw28gju) 11

[3.2.7 - Add Content Pages - Deliverable](#_85mgxjya8kbe) 11

[3.2.8 - Research Mooshak - Milestone](#_m5opilc8irg) 11

[3.2.9 - Set Up Mooshak - Milestone](#_ko0ntxtabzrn) 11

[3.2.10 - Integrate Mooshak with Wordpress - Milestone](#_4g4cwjcyd4jn) 11

[3.2.11 - Mooshak Logins - Deliverable](#_mruypp4ohazb) 11

[3.2.12 - User Manual - Deliverable](#_5510en3ej8kg) 12

[3.3 Work Breakdown Structure](#_d1h79cwjjfo6) 12

[3.3.0 - Set Up Building Environment](#_b9ssga8ey05z) 12

[3.3.1 - Install Wordpress](#_ewp5pz4dot9w) 12

[3.3.2 - Blog Design](#_kabpla7cgn3o) 12

[3.3.3 - Blog Posting Feature](#_cfm5273ig78m) 13

[3.3.4 - Blog Editing Feature](#_7lbtcxqzde29) 13

[3.3.5 - Blog Searching](#_uzzbg0c0tpk4) 13

[3.3.6 - Login](#_dfg47v98qn5e) 13

[3.3.7 - Add Content Pages](#_hu0nfpab7uh4) 13

[3.3.8 - Research Mooshak](#_n29ayhtiroje) 13

[3.3.9 - Set Up Mooshak](#_oad469j5wv6y) 14

[3.3.10 - Integrate Mooshak with Wordpress](#_a31ukqmc50ez) 14

[3.3.11 - Mooshak Logins](#_g6zw6nhtm0qb) 14

[3.3.12 - User Manual](#_5gegs5dqddoq) 14

[3.4 Gannt chart](#_1brr3sjq13vo) 15

[3.5 Task Dependency Diagram](#_bwbmsolm2dkw) 15

[**4. Appendix**](#_4jihqm9itge7) **16**

[4.1 Glossary of terms related to your project](#_ngpetty4wawo) 16

[4.2 Author information](#_wj1hzbzbvawr) 16

[4.3 Additional documents](#_cwwbxszd4uq) 16

[6.3.1 - The Client’s Logo](#_3qw81avt9abz) 16

[Figure 1 - Jaffy the Giraffe](#_d5jq919d6oex) 17

# 1. Introduction

## 1.1 Purpose

This document outlines our implementation team’s approach to developing a website for the client, Gusty Cooper. It seeks to minimize ambiguity between the client and the developers by providing an overview of the system and describing the plan for it to be developed.

## 1.2 Scope

This project is to develop a website for the client, a professor at the University of Mary Washington, Gusty Cooper. The client’s students will use the site to submit code answers to posted problems, and anyone will be able to find the site and see the client’s blog. The website is maintained by the client following completion.

## 1.3 References

The client’s previous websites may be referenced to further understand his background.

<http://gusty.bike/>

<http://gustycooper.github.io/gustycooper.github.io/>

<http://gustycooper.org/>

## 1.4 Overview of the remainder of the document

The document is organized such that in reading section 2 of the document, an in-depth understanding of the system and its requirements can be gathered, including both functional and non-functional features. Section 3 details the project’s plan, including the approach, milestones and deliverables, work schedule breakdown, as well as the gannt chart and task dependency diagram. In section 4, the appendix, terms used in the document are defined and additional resources are listed for reference.

# 2. Project Description

## 2.1 System overview

The website will be hosted on AWS cloud service. It has two main features: a personal blog written by the client, and Mooshak for the client to host and receive submissions for a programming. Wordpress will be used to host the blog. The client can add, remove, and edit blog posts as well as define and edit the programming competition requirements. Comments will be disabled on the website using Disable Comments Plugin. The client can create and remove login credentials for users as needed. Users may see the client’s blog whether or not they have an account. Users with accounts may login to participate in contests.

## 2.2 Client characteristics

The client, Gusty Cooper, is a professor at the University of Mary Washington. The client needs a website that will aid him in grading code submissions from students, and function as a place to host his personal blog. As an experienced computer scientist, the client desires that the site be built using a modern technology stack. He is searching for a site that is easy to use and maintain in his free time, since he has been disappointed by the complexity and cumbersome nature of past websites he has used.

## 2.3 User characteristics

There are three types of users for the website: the administrator, a student user, and anonymous viewer.

The client is the site administrator, who is interested in maintaining both his personal blog and the programming contest. The client is the only user able to create, edit, and remove blog posts, as well as create and edit programming contests he hosts through Mooshak. With Mooshak, the client is able to create and delete student accounts with usernames and passwords.

Anonymous viewers are any users who may visit the client’s website. These users may be students, professors, fans of Gusty, or anyone who may come upon the client’s website. They will be able to view all of Gusty’s posts on the blog and search the blog for content they may want to find. They will not be able to create an account and will not be able to access any of the programming contest features.

Next, students in the client’s class want to be able to participate in problem submission in the programming contest for their classes with Gusty. These students will be given accounts from the client, allowing them to login to Mooshak and participate in the programming contests by submitting solutions via a file upload. Only students given accounts by the client will be able to login. These students, as a sub-user of a viewer, will also be able to view Gusty’s blog, in order to see what their professor is posting.

## 2.4 Functional requirements

The following outlines the functional requirements of the system that were introduced in section 2.1, documented below.

### 2.4.1 - Blog-posting

Description*:* Client will be able to post both text and picture content to the website, where it will be publicly displayed on the homepage. The blog will support Markdown syntax for special formatting. After a post has been made, the client will be able to edit the text and pictures of the content afterwards, and also delete the post altogether. Commenting by the users/viewers of the site shall be disabled.

### 2.4.2 - Blog Search

Description:Any viewer will be able to search the website for other content within the host site. This does not pertain to the contest-hosting service, if a different one is used than the initial site.

### 

### 2.4.3 - Ease of Use & Resources

Description:Website will be easily updatable and manageable, with the tools available for ease of use and instruction. This will be done with a user manual written by the implementation team describing the structure of their code, and providing further resources on how to use the software they chose and/or coded in.

### 2.4.4 - Admin and User Logins

Description: The client will be provided with an option to login to the website with administrative privileges. They will have the option to manage their own username and password for the website. The client/admin will also be given the option to create separate usernames and passwords for other users. The other users created by the administrator will not have the privilege of changing their username or password, and will not be granted administrative privileges.

### 2.4.5 - Contest Hosting Service

Description: The client will be able to host student contests, where students with login credentials can participate in either within or outside of the website’s framework. The client will be able to post content questions or prompts that can be seen and answered by the student users.

### 2.4.6 - Contest Student Submission

Description: The users will be presented with the question to be answered, provided by the client, and will answer them within the website.

### 2.4.7 - Add Site Content Page

Description: The client will be able to add pages to the blog to separate blog posts according to their functions. For example, a page category could be programming, bikes, or family, or any page the client chooses to add. These pages will create a new page which will host all of the blog posts in that particular category.

### 2.4.8 - Image Slideshow

Description:The client will be able to add images to an image slideshow featured on the homepage, for everyone visiting his site to view.

## 2.5 General constraints

The following non-functional requirements describe other constraints on the system that are important to the client, such as his need for ease of use, a flexible budget, and the site hosting.

### 2.5.1 - Ease of use

The website’s features will be simple enough that the client will be able to figure out how to use it within an hour.   
*Rationale*: The client needs to be able to quickly and easily add content to the site without much hassle. The website will be easily accessible to the client and easy for him to maintain in his free time.

### 2.5.2 - Cloud hosted

The website will be hosted on some cloud based platform. Currently, the client deploys his other websites through AWS, though he is open to other cloud hosting options to best suit his needs.  
*Rationale*: Hosting the website on a cloud based service allows the site to be persistent and allows it to be easily maintained and updated by the client.

### 2.5.3 - Budget limit

The website’s budget will be flexible at around $100 per year. This amount may be negotiated further with discussion with the client.  
*Rationale*: The client is willing to pay for a good quality website that will provide him with the features and ease of use that he needs. Currently, the client pays around $100 per year for his websites on AWS as a reference point, though he is willing to pay more or less for a better alternative.

### 2.5.4 - Logo

The website will include the client’s logo. See appendix 6.3.1 for the client’s logo.  
*Rationale*: The client wants his logo to be displayed somewhere creatively on the website, to be seen by all who view it. His son created the logo and he is proud of it.

### 2.5.5 - User manual

The client would like to receive a user manual that will instruct him on how to use the system and update the system. This will instruct him on how to perform functions related to posting on his blog and setting up programming contests with the system. This manual will also contain details about the implementation and software the team has used for the project, so if updates need to be made to the system, then the client, as someone with knowledge of computer science, will be able to update it in the future.  
*Rationale*: The client needs the website functionality to be easily accessible to him, and would find it helpful for a manual to be included for his convenience. He values having input with the implementation team and is very interested to see the technologies they use to implement the website, which he could expand upon in the future.

# 3. Project Schedule

One of the biggest requirements is to integrate Mooshak with a website. The implementation team has decided to focus on Mooshak rather then creating a website from scratch. Testing will be done in the first 3 weeks in order to fully understand Mooshak and how to integrate it. The last 2 weeks will be focused on teaching the client what is on his website and map out to him where items are stored and implemented. The schedule also has two weeks where is something goes awry, there is still time in the schedule to fix it.

Week 1

*Items Due:* February 19th Project Plan Due

* Research mooshak
  + The team will be researching mooshak installation, figuring out what is the best way to integrate the tool to a wordpress website.
* Setup AWS CodeCommit
  + The team will figure out how to edit the website either on the client's current aws account or a new account.

Week 2

*Items Due:* February 28th Memo Due

* Research and Test Mooshak installation
  + The team will test Mooshak integration to wordpress on a dummy site. Testing to see what is the most secure way to set up mooshak on a website.

Spring Break

* There will be no work being done during this time.

Week 3

*Items Due:* March 14th Usability Study Due

* Test Mooshak
  + A continuation of week 2.

Week 4

* Install Wordpress, set up plugins, and pick a theme
  + The team will set up wordpress on AWS. Once successfully installed, a theme will be picked out based on what the client likes best.

Week 5

*Items Due:* March 28th Testing Plan Due

* Setup Mooshak on official site
  + Setup Mooshak and make sure the team knows where the file is saved and how the client can access it in the future.

Week 6

* Testing and Correcting
  + Test to make sure Mooshak is successfully integrated with the website. Confirm with client that the website design is to his liking.

Week 7

* Testing
  + The team will continue to test the website as well as set up the final project speech.

Week 8

*Items Due:* April 18th Project Due

* Give Gusty a tutorial on his new website and the layout on how the website was created.

## 3.1 Approach

This team will aim to have an agile approach. Each week the team will select what they

would like to focus on for the next 4-5 days. The biggest task in this project is

integrating Mooshak and understanding Wordpress’ setup, so research is important in

this project.

## 3.2 Milestones and Deliverables

The following task list outlines our list of milestones and deliverables for the

development of the website. For this dual purpose site, thee four main parts are: the

initial set up milestone, the blogging deliverable, the programming contest deliverable

and the user manual deliverable. The subtasks for each of the major tasks are

enumerated below each section for reference.

**Set Up Milestone**

### 3.2.0 - Set Up Building Environment - Milestone

First, the team will need to be able to ensure that the site is ready to be hosted on

AWS, and will need to setup the Wordpress server and version control there.

**Blogging Deliverable**

The first major deliverable of the site is the blogging feature, which can be delivered

to the client completely after the following milestones are completed. Most of the

features below are integral to Wordpress and will be setup in tandem.

### 3.2.1 - Install Wordpress - Milestone

The team will become familiar with Wordpress, and will install a Wordpress site so

that all members will have administrative and development access.

### 3.2.2 - Blog Design - Deliverable

The team will work on the creative design of the site, including incorporation of

Gusty’s logo and image sliders. At the end of this task, the team should deliver a

prototype of the design to the client for design feedback.

### 3.2.3 - Blog Posting Feature - Deliverable

The team will implement the blog’s simple posting feature, which will allow the

administrator to create blog posts with embedded text and images, as well as a

tagging system. The team will also ensure that comments on blog posts are

disabled.

### 3.2.4 - Blog Editing Feature - Deliverable

The team will implement the blog editing feature, which will allow the

administrator to edit or remove previously made blog posts.

### 3.2.5 - Blog Searching - Deliverable

The team will implement the blog searching feature, which will allow any viewer of

the blog to search its posts. The search will include both blog content searching

and searching and filtering based on tags.

### 3.2.6 - Login - Deliverable

The team will ensure Gusty is given an account to login to the Wordpress site, so

that he may become an administrator of the blog, to add, edit, and remove blog

posts and pages.

### 3.2.7 - Add Content Pages - Deliverable

The team will implement the feature that allows the administrator to easily add

pages to the blog. In this stage, the team should also ensure, that in the blog

posting feature, the client can easily associated a blog post with a page.

**Programming Contest Deliverable**

The other feature of the website, the programming contest implemented using Mooshak, can be delivered to the client depending on the following subtasks.

### 3.2.8 - Research Mooshak - Milestone

Before implementation, the team will need to spend some time researching

and testing Mooshak, to become familiar with the software, and how it might be

integrated with AWS and Wordpress.

### 3.2.9 - Set Up Mooshak - Milestone

After researching, the team will be able to set up Mooshak so that it may

be used for programming contests.

### 3.2.10 - Integrate Mooshak with Wordpress - Milestone

The team will need to integrate Mooshak with Wordpress after researching this

possibility.

### 3.2.11 - Mooshak Logins - Deliverable

The team will need to ensure that Mooshak will support both the client’s

administrative login as well as student login. Based on Mooshak’s features, the

administrator will be able to post problems and create student accounts once

logged in, and the students will be able to submit files to problems once logged in.

**User Manual Deliverable**

### 3.2.12 - User Manual - Deliverable

The team should compile a user manual that will instruct the client on how to use

the blogging and programming contest features. It will also instruct the client on

how to update the site in the future if needed and give details on its

implementation for the client’s own personal understanding.

## 3.3 Work Breakdown Structure

The following section lists project schedule estimates for each of the milestones and deliverables, as well as a rationale for each one.

### 3.3.0 - Set Up Building Environment

*Estimated* *time*: 4 days

*Justification*: Getting a server running to develop the website on and setting up the version control should not take more than 4 days. This estimate was based on previous experience.

### 3.3.1 - Install Wordpress

*Estimated* *time*: 5 days

*Justification*: We estimate that it should only take about a day to initially install Wordpress and its dependencies on the server. The additional 4 days were added to allow the development team to become more familiar with Wordpress and to give more time for unforeseen challenges.

### 3.3.2 - Blog Design

*Estimated* *time*: 45 minutes

*Justification*: Choosing a blog design will be the simply finding a theme and a way to incorporate Gusty’s logo into it. It is not expected to take long.

### 3.3.3 - Blog Posting Feature

*Estimated* *time*: 45 minutes

*Justification*: Enabling blog posts will simply require tweaking the configuration via the Wordpress GUI.

### 3.3.4 - Blog Editing Feature

*Estimated* *time*: 45 minutes

*Justification*: Adding the ability to edit previous posts will require changing settings in the Wordpress GUI.

### 3.3.5 - Blog Searching

*Estimated* *time*: 45 minutes

*Justification*: Enabling searching of the blog is a simple task, it should be done from the Wordpress GUI.

### 3.3.6 - Login

*Estimated* *time*: 45 minutes

*Justification*: Adding a login for Gusty is simple, and can be done from the Wordpress GUI.

### 3.3.7 - Add Content Pages

*Estimated* *time*: 1 hour

*Justification*: Custom content pages should be easy to dynamically add via the Wordpress GUI.

### 3.3.8 - Research Mooshak

*Estimated* *time*: 10 days

*Justification*: We allotted 10 days to researching Mooshak due to a lack of understanding of the system. This should allow adequate time to become familiar with how the system works and how it can be integrated with Wordpress.

### 3.3.9 - Set Up Mooshak

*Estimated* *time*: 3 days

*Justification*: Once the team sufficiently understands how Mooshak works, setting it up on a server should not take very long. This estimate was made from previous experience with similar systems such as nginx and apache.

### 3.3.10 - Integrate Mooshak with Wordpress

*Estimated* *time*: 2 days

*Justification*: With a working Mooshak system, integration with Wordpress should be relatively simple. We do not expect it to take much longer than simply adding another accessible directory to the web server but added some extra time for unforeseen challenges.

### 3.3.11 - Mooshak Logins

*Estimated* *time*: 2 days

*Justification*: Once Mooshak is working, the administrator should have the ability to create student accounts from the GUI. We added some extra time to account for unexpected problems.

### 3.3.12 - User Manual

*Estimated* *time*: 5 days

*Justification*: Once the team understands how the system works, the process of compiling that understanding into a manual should only take about 5 days, based on previous experience.

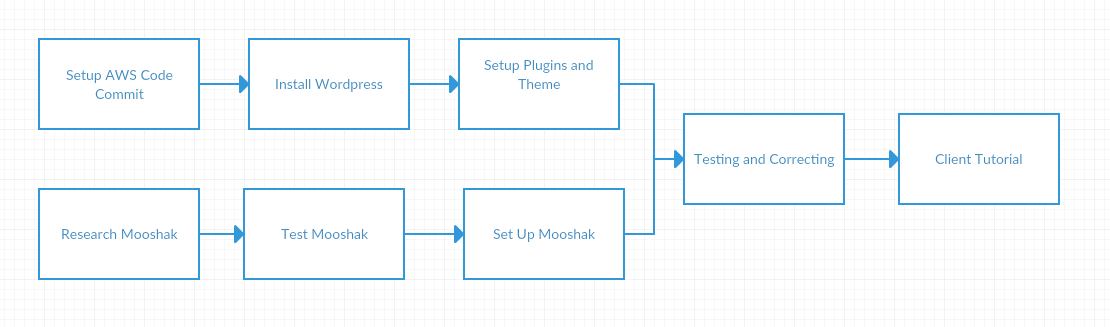
## 3.4 Gannt chart

****

The first 4 weeks will focus on setup and how to install Mooshak. During this time the team will figure out what is the best way to integrate Mooshak on a website. The last 4 weeks will setting up the final product; creating a Wordpress site runned on an AWS server.

## 3.5 Task Dependency Diagram

The following section contains a task dependency diagram to illustrate the dependence relationship between the tasks.



After the server and AWS CodeCommit have been set up, Wordpress can be installed on the server. Once Wordpress is installed, the necessary Wordpress plugins can be installed and the configuration adjusted.

Before any work can be done with Mooshak, the team will need to research Mooshak and learn how it works. When the team has a basic understanding of how the system works, testing can begin. Once the team understands Mooshak and has tested it, Mooshak can be setup on the server.

After Wordpress and Mooshak are running and have been integrated, the website can be tested and any problems or issues may be corrected. The user manual will be created after the website has been completed to ensure accuracy.

# 4. Appendix

## 4.1 Glossary of terms related to your project

* Amazon Web Services (AWS): A popular cloud computing platform hosted by Amazon. The client currently employs this service to host his websites. See <https://aws.amazon.com/> for more details.
* AWS CodeCommit: A service that hosts Git repositories for version control within AWS. See <https://aws.amazon.com/codecommit/> for more details.
* Mooshak: A system that allows for implementation of a programming contest or similar functionality on a web platform. For more details see <https://mooshak.dcc.fc.up.pt/> for Mooshak version 1 and <https://mooshak2.dcc.fc.up.pt/> for Mooshak version 2.
* Wordpress: A free and open source website management tool. This is provided by the University with Domain of One’s Own.

## 4.2 Author information

Jessica Spranger: 1.1, 1.4, 2.2, 2.3, 2.5, 3.2

Evan Shipman: 2.4, 3.3, 3.5

Jacqueline Coates: 2.1, 3 Intro, 3.1, 3.4

## 4.3 Additional documents

### 4.3.1 - The Client’s Logo

The client’s logo features Jaffy the giraffe, a character of his creation that represents his website.

#### *Figure* 1 *- Jaffy the Giraffe*

