

Test Plan

for

Gusty.bike

Prepared by

**Jessica Spranger
Evan Shipman
Jacqueline Coates**

3/28/2019

	2
1. Introduction	3
1.1 Purpose	3
1.2 Scope	3
1.3 References	3
1.4 Overview of the remainder of the document	3
2. Project Description	4
2.1 System overview	4
2.2 Client characteristics	4
2.3 User characteristics	4
2.4 Functional requirements	5
2.4.1 - Blog-posting	5
2.4.2 - Blog Search	6
2.4.3 - Admin and User Logins	6
2.4.4 - Contest Hosting Service	6
2.4.5 - Contest Student Submission	6
2.4.6 - Add Site Content Page	7
2.4.7 - Image Slideshow	7
2.5 General constraints	7
2.5.1 - Ease of use	7
2.5.2 - Cloud hosted	7
2.5.3 - Budget limit	8
2.5.4 - Logo	8
2.5.5 - User manual	8
3. Test Plan	9
3.1 Testing strategy	9
3.2 Testing resources and staffing	10
3.3 Test work products	10
3.4 Test record keeping	11
3.5 Test schedule	12
4. Test Procedure	14
4. Appendix	16
4.1 Glossary of terms related to your project	16
4.2 Author information	16
4.3 Additional documents	16
4.3.1 - The Client's Logo	16
Figure 1 - Jaffy the Giraffe	17

1. Introduction

1.1 Purpose

This document outlines our implementation team's approach to testing the website for the client, Gusty Cooper. It seeks to minimize ambiguity between implementation team and the testing team by providing an overview of the system and describing the plan for its testing. Additionally, this allows documentation to be provided to the client showing what testing was performed to ensure that the requirements of the website are met.

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://gustycoper.github.io/gustycoper.github.io/>

<http://gustycoper.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. Next, section 3 describes the testing strategy, the testing resources, the testing products and record keeping, as well as the testing schedule for the project. After this, in section 4,

the testing procedure is outlined in extensive detail, so that in reading it, the implementation team and the testing team will have a good idea of the test cases to perform to ensure the website's requirements are met. Finally, in section 5, 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. These users could be of varying technical background, and it is assumed that they will need little to no technical background in order to interact with Gusty's blog and view his posts.

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, as trusted users of the site, and it is assumed that they have basic or introductory technical background and will be able to understand and learn how to work the Mooshak software. 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.

2.4.3 - 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.

2.4.4 - 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.

2.4.5 - 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.

2.4.6 - 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.

2.4.7 - 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. 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.

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. Test Plan

Concentration for testing the system will be on the user manual and testing the installation procedure and the system as a whole, rather than the individual components that make up the system, e.g. WordPress and Mooshak. The justification for this is due to the large percentage of development being component-based.

3.1 Testing strategy

The primary strategies for testing the system will be through installing a clone of the system by following the user manual, analyzing the system as a whole, the manual's usability, the system's usability, and the security of the system. The testing team will install a copy of the system on a newly created AWS instance, using login credentials given to them and a draft of the user manual that will be given to the client upon completion. It will detail the necessary steps for setting up the website as well as how to maintain it.

Usability Testing

- The user manual should be easy to follow during installation and setup.

- The Wordpress component of the system will be tested from the client's perspective to ensure that the system is easily updatable and maintainable.
- Mooshak will be tested from the client's perspective in terms of its ease of use.¹

Security Testing²

- Both the Wordpress and Mooshak components will be tested to verify that an unauthorized user may not access administrative capabilities.
- Further testing will be done on Mooshak to confirm that an unauthorized user may not enter without a username and password.

System Testing

- The website should remain functional across multiple web browsers.
- Both Wordpress and Mooshak should be accessible and responsive.

3.2 Testing resources and staffing

In order to complete the test, the testing team will need access the user manual and the shell script. The script will install Mooshak and Wordpress on the provided AWS instance. Testers will be given login credentials for the AWS account, given to the team the day of. There will also be a manual given to the testing team, in which they will need to follow a series of steps on how to install Mooshak and Wordpress and how to login.

¹ The implementation team will perform ease-of-use testing on Mooshak

² Security Testing will be performed by the implementation team, not the test team

3.3 Test work products

By testing the usability, security, and system of the system, we can confirm the effectiveness of the user manual and any error that may occur if the system is updated. The main portion of testing will be designed around the user manual. This manual will eventually be given to the client as a reference on how to install Mooshak and Wordpress on the AWS instance. The manual will have space for comments to be written down. For the security testing, the implementation team will attempt to login to Mooshak and see if they can break into the system; if this occurs then it will be documented the steps taken to login.

3.4 Test record keeping

First, the testing team will be able to provide feedback on their work in testing the installation with the user manual. When they move through the installation of both Mooshak and Wordpress, they will produce the website that is viewable from the AWS instance. This website is the first record that they will produce after the installation process that can be evaluated by the implementation team.

Additionally, in order to test the keep track of their progress in the user manual, the testing team will use the following questions to provide feedback on the process:

1. Were you able to complete the tasks given by the user manual?
 - a. Are you able to access Mooshak after installing it?
 - b. Are you able to access the Wordpress blog after installing it?
2. If the answer to the previous is 'No', what step were you halted by and why?
3. Was the manual clear in describing the steps needed to perform the tasks?
4. Was anything confusing about the user manual, in the way it was worded, or the steps that were needed?

5. Is there any other functionality or step not described in the user manual, that you would have liked to see explained by it?
6. Provide any other comments you have about the user manual or the website.

Secondly, for all other testing that is done by the implementation team and not the testing team, the implementation team will check off each of our security tests. We will also check off each of the requirements as we deem them completed, to ensure that all test cases and requirements pass before the product can be delivered to the client.

3.5 Test schedule

The testing schedule of this product will be conducted in two phases. First, by the testing team, and second, by the implementation team. Firstly, the implementation team will have access to the testing team, whose testing feedback will be vital in testing the product. The testing team to work through the draft of the user manual as if they were Gusty Cooper, ensuring that they can install both Mooshak and Wordpress based on the guidance given to them by the manual.

In the following week, the implementation team will have received this feedback and make the necessary changes to correct the user manual and the installation process. Once they can verify that the installer script and instructions provided in the user manual are clear and correct, the team can continue to test the rest of the site's functionalities. This will ensure that all of the requirements of the project have been met. This will include testing the security of the site as well as system testing to ensure that the site is response. We will then be ready to give Gusty an in-person tutorial of how the site works and deliver to him the user manual and the installer script for the project.

Week 1

- Write the test plan document
 - Thursday 3/28: Test Plan Due
- Deliver website designs to Gusty
 - Deliver potential website designs to the client, to make sure he likes the blog's appearance and design.
- Write the user manual
 - The draft of this is to be finished by Tuesday the following week.

Week 2

- Tuesday 4/2: Testing Team is working during class
 - Deliver to the testing team the user manual to guide them through the installation of Mooshak and Wordpress.
- Thursday 4/4: Testing Team is working during class
 - The testing team finishes working through the user manual.
 - The testing team provides feedback on the manual and the website's functionality.

Week 3

- Revise the user manual
 - Based on the testing team's feedback, revise the user manual and make any necessary changes to the setup of Mooshak and Wordpress.
- Implementation Team Testing
 - Testing of the website's requirements will be performed by the implementation team, so each requirement can be checked off.
 - Additional Security and System testing will be done at this time.

Week 4

- Give Gusty a user tutorial
 - Deliver to Gusty the user manual, and walk him through the functionality and structure of the site.

- Last minute changes can be made to the website based on his feedback on the client's usability testing of the site.
- Thursday 4/18: Project Due

Week 5

- Project Presentations
 - Demonstrate the project in class with the client.

4. Test Procedure

The testing team will focus on the usability and system testing for this procedure. The team will be given a username and password for an AWS account as well as an ssh key for the instance. The instance used for this testing will already be installed and set up on the account. The team will also be given a shell script, which it will install Mooshak and Wordpress, and a user manual, which will go over the steps of installation.

Usability Test

1. Install Mooshak

Run the shell script as directed. The beginning of the user manual will instruct you how to set each package as it is downloading. After the script is finished go to section 3.4 and answer the questions. There will be space in the user manual for testers to write down extra questions or comments about the script.

2. Install Wordpress

The shell script will first download Mooshak and then Wordpress. Same instructions as task 1.

Test Number	Related Requirements	Subsystem	Purpose	Test Case Data	Expected Results
1	2.4.5 2.4.4	Apache Java	Install mooshak	Shell script	Mooshak successfully installed
2	2.4.3	Apache SQL	Install wordpress	Shell script	Wordpress successfully

		PHP			installed
--	--	-----	--	--	-----------

System Test

1. Linking Wordpress and Mooshak together

After Wordpress and Mooshak have been successfully installed, the user manual describes how to set up part of the website. This is to ensure that the directions in the user manual are clear and concise. Some of the instructions will be how to link Mooshak to the blog site and establishing user restrictions.

Test Number	Related Requirements	Subsystem	Purpose	Test Case Data	Expected Results
1	2.4.5 2.5.5 2.4.6	Wordpress	Create a link for mooshak	Wordpress Post Mooshak url	Create a private policy page

Security Test

Security testing of the system will be performed by the implementation team. The rationale for this is that due to the system's component-based nature and the overall security of the system being largely dependent on the developers of the individual components. The implementation will perform the following tests:

1. Verify that an unauthorized user may not access administrative capabilities in Mooshak
2. Verify that an unauthorized user may not login to Wordpress
3. Verify that a user may not enter Mooshak without a username and password

Test Number	Related Requirements	Subsystem	Purpose	Expected Results
1	2.4.3 2.4.4	Mooshak	Only the professor has admin capabilities	An unauthorized user may not access admin capabilities

2	2.4.3	Wordpress	Check login restrictions	An unauthorized user is unable to login
3	2.4.3 2.4.4	Mooshak	Check login restrictions	A non-student cannot access Mooshak

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.
- 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.

4.2 Author information

Jessica Spranger: 1.1, 3.4, 3.5

Evan Shipman: 3 intro, 3.1, 4 security test

Jacqueline Coates: 3.2, 3.3, 4 intro, 4 usability test, 4 system test

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

