KTF UI'S MASTER TEST PLAN

Created by [BigBang]
Issued March 29, 2016

Version History

V. No.	Description	Author	Date
1.0	Modified template as per group decision on the sections to include / exclude.	Anissa Lintang, Kega Kurniawan	2 April 2016
1.1	Modified template on tests targeted	Marvin Mitchell	3 April 2016
1.2	Modified template for scenario case per test	Anissa Lintang, Dion Edo	9 April 2016

Table of Contents

1. Introduction

The primary goal of this project is to develop a website for Komunitas Tari Fisip, Universitas Indonesia (KTF UI). (KTF UI) wishes to have a professional website that look like a company profile. The website shows the documentation on their show such as text, photos and videos linked to youtube. The site have approximately 10 pages and all of it have a responsive website template. This test plan contains a comprehensive list of tests that will be performed along with a workflow of how the tests will be executed.

1.1 Purpose

The purpose of the Iteration Test Plan is to gather all of the information necessary to plan and control the test effort for this phase.
This Test Plan for the KTF UI's Website supports the following objectives:
- Identify the requirements that are to be tested.
- Outline the testing approach that will be used.
- Describe the workflow of the testing process that must be executed.
- Provide a timeline with milestones for the testing phase.

1.2 Scope

This document is intended to provide a test plan to test the KTF UI's Website, which Bigbang developed. The test plan will consist of unit, integration, function, user interface, load, and configuration. Testing techniques that will be performed include white box and black box testing, and basis path testing. A test plan workflow will also be included along with milestones that have been set for this phase.

1.3 Document Terminology and Acronyms

Term	Definition
KTF UI	Komunitas Tari Fisip, Universitas Indonesia
UI	User Interface
BVA	Boundary Value Analysis
GUI	Graphical User Interface
API	Application Programming Interface

1.4 References

Ini di isi segala macem referensi buat projek ini

1.5 Document Structure

The remainder of this document is divided into following major parts: evaluation mission and test motivation, target test items, outline of planned tests, test approach and testing workflow, iteration milestones. The evaluation mission and test motivation contains a brief background on this project, its objectives and motivators for testing. The target test items and outline of planned tests include what will be tested and what tests will not be performed. The test approach contains the actual tests that were performed and how the tests were carried out. The testing workflow contains the workflow that BigBang Team followed in this phase. The last two sections contain the milestones of this phase.

2. Evaluation Mission and Test Motivation

The goal of this test plan is to ensure that the KTF UI's Website meets the specifications and design criteria of the two previous phases. Moreover, the test plan will provide a methodology on what the implementation team should test and the types of tests they will perform. Finally, the test plan will enable BigBang Team to release a stable KTF UI's Website.

2.1 Background

This project involves creating KTF UI's Website based on the requirements and design documents of the two previous phases. The website will be developed by the implementation using Wordpress CMS with template design and developed by BigBang Team. A comprehensive test plan has been developed to ensure that the website conforms to the specifications, design and to perform quality assurance on the final product. This will enable Bigbang Team release a complete and bug free KTF UI's Website and minimize the risk of website failure. The requirements document outlines the website's specifications and high-level requirements along with an analysis model with use cases diagrams of the website. The design document contains architectural and website interface, which is a foundation that the implementation team can create the website. The test plan will allow BigBang Team to verify if the final product successfully meets these specifications with a variety of testing techniques. The plan will also help in fault detection with the test cases that have been designed. The requirements and design documents are available at

https://github.com/gunadarma-academy/asde-big-bang.

2.2 Evaluation Mission

The three main objectives of this plan are:
- Ensuring that the specifications of the requirements document have been achieved.
- Ensuring that the specifications of the design document have been achieved.
- Ensuring that the risk of website failure is reduced to a minimum.

To achieve these objectives, BigBang Team has developed a test plan to verify that these objectives have been met. Meeting these objectives will enable BigBang Team to release a stable version KTF UI's Website.

2.3 Test Motivators

The targeted test items listed below will be the motivation for testing in this phase.

Unit Testing	A select number of methods will be tested in a couple of classes with black and white box testing to ensure that they function correctly

Function Testing	Will ensure that the use cases have been met
User Interface Testing	Will verify if the requirements of the GUI have been implemented as specified
Load and Stress Testing	See how the website performs when being access at its limits
Configuration Testing	Ensure that the website works correctly under different environment configurations
Security and Access Control Testing	Ensure and verify the website only can be accessed by user authorized

2.4 Test Milestone

Milestone of the master test plan will start from the 4th week due to week 1-3 is a phase of collecting user requirements, design and development of KTF UI's website.

Week	To-do Tests	Performer
4	Unit Testing	Marvin Mitchell
4	Function Testing	Anissa Lintang Dion Edo
4	Security and Access Control Testing	Kega Kurniawan
5	Load and Stress Testing	Marvin Mithcell Kega Kurniawan
5	Configuration Testing	Anissa Lintang
6	User Interface Testing	Dion Edo

3. Target Test Items

In this section, we will list the target test items. These are the items that should be tested. Due to time restrictions, we were not able to document and generate test cases for all the target test items; therefore, although we list all the target test items, we only provide a detailed test plan for a few of the major test items. For ease of reference, we have categorized the test items by motivation.

Unit Testing
Unit testing consists of testing all the different units of the system, in isolation. In essence, we must therefore test each class in isolation, and each method in isolation using white box and black box techniques. The list of test items for unit testing consists of all the classes and all their methods, as per the design document. Below is a list of the test items for which test cases have been generated and included in this document:
- Function pureal_init_style
- Function pureal_init_wp_setup
- Function pureal_sanitize_image
- Function pureal_init_widget
- Function pureal_set_excerpt_length
- Function pureal_init_theme_customization
- Function pureal_update_custom_css
- Function pureal_preview_js

Function Testing
Function testing consists of testing all the requirements and specifications from user, as per the requirements and specifications document. In essence, the list of functions to test corresponds to the list of use cases and requirements in the requirements document. Due to the importance of function testing, we have included detailed test cases for all the product functions. Below is the list of functions that were tested:
- Request profile
- Request information
- Request multimedia
- Insert content
- Edit and update content

User Interface Testing
User interface testing is concerned with making sure that each functionality concerning the user interface is works as per the requirements defined in the design document. Below is a list of the User Interface items that were tested:
- Page Home
- Page What we offer

- Page Achievements
- Page Gallery
- Page Project
- Page Contacts
- Page About

Load and Stress Testing

Load and Stress Testing is a part of website performance testing, concerned with testing the system beyond the limits it was designed for. In this type of test, we have focused mainly on testing the website when accessed beyond the limit. The testing will using Apache Jmeter software. Below are the test items that were identified:

- Throughput
- Deviation
- Latency
- Sample time

Configuration Testing:

Configuration testing is concerned with testing the website under different environment configurations. In this type of test, we have focused on testing the website under different website browser. Below is a list of the website browser will be tested:

- Mozilla Firefox
- Google Chrome
- Safari
- Opera
- Internet Explorer

Security and Access Control Testing

Security and Access Control Testing will be performed as the website contain any data that can't be accessed if the user is not authorized. Below is a list that will be tested:

- Admin Login

4. Outline of Planned Tests

BigBang Team will perform the following test: unit testing, integration testing, function testing, user interface testing, load and stress testing, configuration testing and security and access control testing. The following tests will not be performed: data and database integrity testing, business cycle testing,and integrity testing.

4.1 Outline of Test Inclusions
The following tests will be performed to test the KTF UI's Website.

No	Test Inclusions	Description
1	*Unit Testing*	Unit testing will be performed with black box and white box testing. Black box testing will include boundary value analysis and equivalence partitioning. White box testing will include basis path testing.
2	*Function Testing*	Function testing will ensure that the use cases have been implemented correctly by verifying if they are present in the website.
3	*User Interface Testing*	The pages of website will be tested by comparing the requirements in the design document and with the actual implementation of the website.
4	*Load and Stress Testing*	Load testing will see how the website performs when being accessed at its limits. This will be achieved by testing the website with the maximum visitor at peak hour.
5	*Configuration Testing*	Configuration testing is concerned with testing the application under different website browser the users may have.
6	*Security and Access Control Testing*	Security test of the website will be conducted to test the log-in system in administrator page to distinguish between guest and administrator. Because in the website environment there are 2 different user, guest and administrator, which have different roles. The guest only can see and request information and in the other hand administrator have many role such as input, delete, and

		update every information in the website, and adjust the configuration of the website.

4.2 Outline of Test Exclusions

Due to the nature of KTF UI's Website implementation, certain tests will be excluded, which are listed below.

No	Test Exclusions	Description
1	*Database and Database Integrity Testing*	We do not perform database and database integrity testing because the website made by using Wordpress CMS (Content Management System) and the database setting and it's integrity has well developed.
2	*Business Cycle Testing*	Business cycle testing is not applicable to KTF UI's Website because although the website showing it's company profile, it does not include any commercial and business things.
3	*Integrity Testing*	We do not perform integrity testing because the website made by using Wordpress CMS (Content Management System) and all the content's integrity has well developed.

5. Test Approach

The Test Approach describes the recommended strategy for designing and implementing the required tests. In this section, we will be describing the details of the tests that need to be performed for each target test item that was identified. These tests will be organized into the following sub-sections:
1. Unit Testing
2. Function Testing
3. User Interface Testing
4. Load and Stress Testing
5. Configuration Testing
6. Security and Access Control Testing

Moreover, for each of these test motivators, test cases will be described in detail. For each test case, we will provide a description of the test case, the inputs (or steps to reproduce) of the test case, and the outputs (the expected results) of the test case.

5.1 Unit Testing
Unit testing will test individual components along with their functions in isolation. This low level form of testing will include black box testing and white box testing. In black box testing, the function's boundaries will be tested to see if any errors occur there. White box testing will verify that all the paths in the function are correct through basis path testing.

5.1.1 Function pureal_init_style
(penjelasan)

5.1.1.1 Black Box Testing
(isinya test scenario)

5.1.1.2 White Box Testing
Basis Path Testing

5.1.2 Function pureal_init_wp_setup
(penjelasan)

5.1.2.1 Black Box Testing
(isinya test scenario)

5.1.2.2 White Box Testing
Basis Path Testing

5.1.3 Function pureal_sanitize_image
(penjelasan)

5.1.3.1 Black Box Testing
(isinya test scenario)

5.1.3.2 White Box Testing
Basis Path Testing

5.1.4 Function pureal_init_widget
(penjelasan)

5.1.4.1 Black Box Testing
(isinya test scenario)

5.1.4.2 White Box Testing
Basis Path Testing

5.1.5 Function pureal_set_excerpt_length
(penjelasan)

5.1.5.1 Black Box Testing
(isinya test scenario)

5.1.5.2 White Box Testing
Basis Path Testing

5.1.6 Function pureal_init_theme_customization
(penjelasan)

5.1.6.1 Black Box Testing
(isinya test scenario)

5.1.6.2 White Box Testing
Basis Path Testing

5.1.7 Function pureal_update_custom_css
(penjelasan)

5.1.7.1 Black Box Testing
(isinya test scenario)

5.1.7.2 White Box Testing
Basis Path Testing

5.1.8 Function pureal_preview_js
(penjelasan)

5.1.8.1 Black Box Testing
(isinya test scenario)

5.1.8.2 White Box Testing
Basis Path Testing

5.2 Function Testing

This section is concerned with testing the functions (or requirements) of the software. This is a critical aspect of the testing effort, as it ensures that the software meets the requirements, and thus ensures acceptance by the users. For completeness, each requirement should be associated with a set of test cases, some with valid data, and some with invalid data. Despite time restrictions, we have included all the major product functions as well as test cases for each one. In fact, one of the benefits of having this section as complete as possible is that the implementation team can consult this list of test cases to ensure that they have properly implemented the functions, and that the software works both in the normal cases and exceptional cases. The following sections are devoted to the major functions that were selected as testing targets. Each section lists and describes the different test cases that are important to check.

5.3 User Interface Testing

To test the User Interface, each functionality described in the design document will be verified to see if it has been implemented correctly, if it responds normally and also if no errors occur during the process between the user and the game.
A schema will be used to test (unit), what is the purpose of the test (what is tested), what are the inputs (from the user for instance), what is the expected result and also what is the effective (real result).

5.4 Load and Stress Testing

Load testing is normally concerned with testing the system beyond the limits it was designed for. However, due to restrictions we have placed on the number of visitor (unknown number) that can access in one time, we cannot, for instance, test to see if the website works with maximum visitor (undefinetely yet). Therefore, we will be testing the website as close as possible to the limits it was designed for. In fact, we will attempt to simulate a fully loaded board, where all the function are displayed. In this scenario, we will re-evaluate the website's functionalities and response times.

5.5 Configuration Testing

Configuration testing is concerned with testing the application under different environment configurations the users may have. For the KTF UI Website, we will be focusing on testing the website under different browser. As per the requirements document, this includes Mozilla Firefox, Opera, Chrome, IE and Safari. In order to simulate these different client environments, we will be using well-known different browser directly. In order to test the KTF UI Website under several different browser.

5.7 Security and Access Control Testing

Security and access control testing have to be conducted to ensure security system especially security for the admin system, it is necessary because this testing will be test the website how well can handle and prevent unauthorized user to access the admin system. In this testing we will conduct security system test for admin login, and see how well security system from admin login to prevent unauthorized user to enter the admin system.

6. Testing Workflow

In this section, we will describe, in detail, the procedures and guidelines that are to be followed during the testing effort. This will outline the flow of the testing activities implicated, and allow us to easily manage the bugs that are found, resulting in a smooth testing phase.

6.1 Workflow Overview
Some of the goals of this testing phase are to test for: correctness of implementation, good GUI and proper performance level.

6.1.1 Test Plan & Software Engineering process
First, let us make clear the relationship between the test plan and the software engineering process of our project. As shown in the next diagram, the relationships are: I We use the detailed design document to produce the unit-testing plan. I We use the detailed design document to produce the integration-testing plan. I After the system is integrated, we test the system's features by using the requirements document. I Finally, we use the systems specifications to ensure that the implemented system follows them.
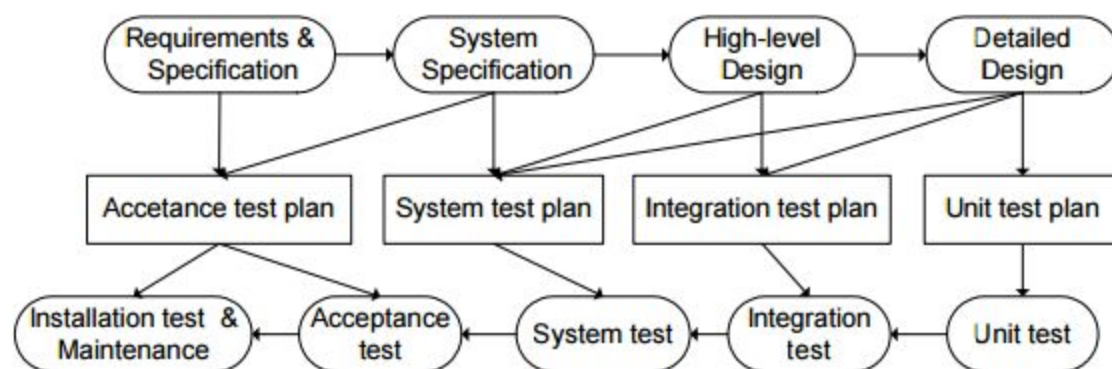
6.1.1 Black Box Testing Template

Unit testing is the activity that verifies each module in isolation. For each test items of unit testing, both black box and white box testing was performed. Several templates for black box testing were made including some for boundary value analysis and equivalence partitioning.

No	Tester Name	Function Name	Expected Output	Actual Output	Bug Find

6.1.2 White Box Testing Template

No	Tester Name	Function Name	Expected Output	Actual Output	Bug Find