CMS Application

Test Plan

**Version**: 1.0

**Created**: 03/10/2017

**Last Updated**: 03/10/2017

Change Log

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Description** |
| 03/10/2017 | 1.0 | Piotr Maj | Document created |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Related documents

|  |  |  |
| --- | --- | --- |
| **#** | **Title** | **Version** |
| [1] | CMS\_CheckList\_AdminPanel\_GUI | 1.0 |
| [2] | CMS\_CheckList\_Blog\_initialGUI | 1.0 |
| [3] | CMS\_FunctionalTestCaseList | 1.0 |

Table of Content

1. Introduction 1
   1. Purpose 1
   2. Project Overview 1
   3. Audience 1
2. Test Strategy 2
   1. Test Object 2
   2. Assumptions and Scope 2
   3. Types of Tests 2
      1. Exploratory Testing 2
      2. GUI Testing 2
      3. Functional Testing 2
      4. Test Automation 2
   4. Test Case’s and Defect’s Classification 3
      1. Test Case’s Classification 3
      2. Defect’s Classification 3
   5. Acceptance criteria 3
3. Execution Strategy 4
   1. Test environment 4
   2. Continous Integration 4
   3. Feature testing 4
   4. Regression testing 5

# Introduction

## Purpose

This test plan describes the testing approach of the CMS application. The document describes testing strategy, execution strategy and exit criteria of this project.

## Project Overview

CMS application is our customized Content Management System based on Ghost publishing platform. Ghost is a fully open source, hackable platform for building and running a modern online publication like blogs, magazines and journalists. We customize it to meet various expectations of our core clients.

## Audience

This document is designed for fallowing audience:

* + Project team members perform tasks specified in this document, and provide input and recommendations on this document.
  + Project Architect and Project Chef Architect to ensures that the test plan and deliverables are in line with the design.
  + Project Managers for the testing activities in the overall project schedule, reviews the document, tracks the performance of the test according to the task herein specified, approves the document and is accountable for the results.
  + Technical Team to provides the environment for testing.
  + Business analysts will provide their inputs on functional changes.

# Test Strategy

## Test Object

The CMS Applications consists of two parts that need to be tested accordingly.

* + Blog – accessible to all public - is generated automatically basing on the settings and content provided in AdminPanel.
  + AdminPanel – accessible only for authorized users - is used for the Blog administration.

## Assumptions and Scope

The fallowing assumptions have been made.

* + In AdminPanel the GUI of setting tabs: General, Navigation, Tags, Code Injection, Apps and Labs is not likely to change, and consequently GUI checklist for this tabs will be described only briefly. If this assumption change, this document MUST be updated accordingly.
  + In AdminPanel the settings tabs Code Injection, Apps, Labs and General > Theme are provided by Ghost framework, are tested my Ghost Foundation Engineers and will not be object of customization, and consequently this options will be tested only briefly. If this assumption change, this document MUST be updated accordingly.
  + In AdminPanel help button (?) and Markdown Help are provided by Ghost framework, are tested my Ghost Foundation Engineers and will not be object of customization, and consequently this options will be tested only briefly. If this assumption change, this document MUST be updated accordingly.
  + Metadata menu will not be tested as it required webpage to be published online.

## Types of Tests

### Exploratory Testing

The purpose of this kind of testing is to find all obvious defects immediately after introduced as well as to find defect in areas not considered in this test plan. Exploratory testing is carried out in the application without any test scripts and documentation. After each defect found, new test case should be created basing on executed exploratory scenario.

### GUI Testing

The purpose of this kind of testing is to find all GUI related defects. GUI Testing is carried out accordingly to provided check lists:

* + CMS\_CheckList\_AdminPanel\_GUI.xlsx
  + CMS\_CheckList\_Blog\_initialGUI.xlsx

### Functional Testing

The purpose of this kind of testing is to cover all important functionalities of the application. Functional Testing is carried out accordingly to provided test case design document:

* + CMS\_FunctionalTestCaseList.xlsx

### Test Automation

The purpose is to automate the most important test cases. The most important functionality of the administration panel is to create and edit new posts. The information which test cases should be automated and which are already automated can be found in test case design document:

* + CMS\_FunctionalTestCaseList.xlsx

## Test Case’s and Issue’s Classification

### Test Case’s Classification

Priority:

* + **1** – test case must be executed.
  + **2** – test case should be executed.
  + **3** – it is up to Test Engineer to decide if execute test case or not.

### Issues’s Classification

Types:

* + **Error** – the functionality that works different than as expected and/or cause termination of the application (synonym of defect, incident or bug).
  + **Observation** – this type is used when Test Engineer is not sure if what he observed is an error or feature and it is not described how this particular functionality should work.
  + **Change Request** – the functionality works correctly but there are some factors cause that this functionality should be changed.

Priority:

* + **1** – issue should be resolved immediately.
  + **2** – issue should be resolved in the normal course of development activities.
  + **3** – issue should be repaired, but repair can be deferred until after more serious defect have been fixed.

Severity:

* + **Critical** – issue that results in the termination of the application or one or more component, causes extensive corruption of the data or the failed function is unusable.
  + **High** – issue that not necessary results in the termination, but causes the application to produce incorrect, incomplete or inconsistent results in important areas.
  + **Low** – issue that does not result in the termination, does not damage the usability of the application, is related to application appearance or the desired results can be easily obtained by working around.

## Acceptance criteria

The criteria that MUST be fulfilled in before each release.

* + 100% automated test cases executed.
  + 100% automated test cases pass.
  + 100% priority 1 test cases executed.
  + 100% priority 1 test cases pass.
  + 100% priority 2 test cases executed.
  + 90% priority 2 test cases pass.
  + No open errors with priority 1 or severity Critical.
  + No open errors with priority 2 or severity High.
  + All remaining errors are either rejected or documented as Change Requests for a future release.

# Execution Strategy

## Test environment

Test will be executed only on desktop devices. No mobile devices like tablets or mobile phones are used for testing. Test will be executed against different operating systems and different browsers.

* + Test planed to be executed on OS X operating system will require real MacBook device.
  + Test planed to be executed on Windows and Linux operating systems will require virtual environment.

List of test targets:

* + OS X: Firefox (latest) - exploratory tests only on this target
  + OS X: Chrome (latest)
  + OS X: Safari (10)
  + Windows: Firefox (latest)
  + Windows: Chrome (latest)
  + Windows: IE (11)
  + Linux: Firefox (latest)
  + Linux: Chrome (latest)

## Continuous Integration

Tests run after each check in into source control repository.

Types of tests that should be run:

* + only automated tests

Test targets:

* + Mac: Firefox (latest)

## Feature testing

Test run after each feature is completed. New created (feature related) functional test cases, most important functional test cases, exploratory testing and automated test shall be run.

Types of tests that should be run:

* + exploratory tests
  + new created, feature related functional test cases
  + most important functional test cases
  + automated tests

Test targets:

* + OS X: Firefox (latest)

## Regression testing

Tests MUST be run before each release.

All types of tests should be run i.e.:

* + exploratory tests
  + GUI tests
  + functional tests
  + automated tests

Test targets:

* + OS X: Firefox (latest) - exploratory tests only on this target
  + OS X: Chrome (latest)
  + OS X: Safari (10)
  + Windows: Firefox (latest)
  + Windows: Chrome (latest)
  + Windows: IE (11)
  + Linux: Firefox (latest)
  + Linux: Chrome (latest)