|  |
| --- |
| Testing Plan |
| MyNewMedia Site Project |
| Zachary Merrill;Stephanie Koesters;John De Los Reyes |

Table of Contents

[1 Introduction 3](#_Toc353560674)

[1.1 Testing purpose 3](#_Toc353560675)

[1.2 Scope 3](#_Toc353560676)

[1.3 Technologies 3](#_Toc353560677)

[1.4 Overview 3](#_Toc353560678)

[2 Core Site Tests 4](#_Toc353560679)

[2.1 Selenium Tests 4](#_Toc353560680)

[2.1.1 Login Test 4](#_Toc353560681)

[2.1.2 User Home Redirect Test 4](#_Toc353560682)

[2.1.3 Search Test 4](#_Toc353560683)

[2.1.4 Subscription Test 4](#_Toc353560684)

[2.1.5 Channel Manager Test 4](#_Toc353560685)

[2.1.6 Channel Home Test 4](#_Toc353560686)

[2.1.7 RSS Feed Pull Test 4](#_Toc353560687)

[2.1.8 Logout Test 4](#_Toc353560688)

[2.1.9 Splash Page Redirect Test 4](#_Toc353560689)

[2.2 3rd-Party Package Test 5](#_Toc353560690)

[2.2.1 Messages 5](#_Toc353560691)

[2.2.2 Tags 5](#_Toc353560692)

[2.2.3 Registration 5](#_Toc353560693)

[3 Channel App Tests 6](#_Toc353560694)

[3.1 Model Tests 6](#_Toc353560695)

[3.1.1 Channel 6](#_Toc353560696)

[3.1.2 Link 6](#_Toc353560697)

[3.1.3 Feed Item 6](#_Toc353560698)

[3.1.4 Feed Tracker 6](#_Toc353560699)

[3.2 View Function Tests 6](#_Toc353560700)

[3.2.1 Model Tests 6](#_Toc353560701)

[3.2.2 View Tests 6](#_Toc353560702)

[4 Profile App Tests 6](#_Toc353560703)

[4.1 Model Tests 6](#_Toc353560704)

[4.1.1 Profile 6](#_Toc353560705)

[4.1.2 Preferences 6](#_Toc353560706)

[4.2 View Function Tests 6](#_Toc353560707)

[4.2.1 Model Tests 6](#_Toc353560708)

[4.2.2 View Tests 6](#_Toc353560709)

[5 Recommendation App Tests 6](#_Toc353560710)

[5.1 Model Tests 6](#_Toc353560711)

[5.2 View Function Tests 6](#_Toc353560712)

[5.2.1 Model Tests 6](#_Toc353560713)

[5.2.2 View Tests 6](#_Toc353560714)

[6 Search App Tests 6](#_Toc353560715)

[6.1 Model Tests 6](#_Toc353560716)

[6.2 View Function Tests 7](#_Toc353560717)

[6.2.1 Model Tests 7](#_Toc353560718)

[6.2.2 View Tests 7](#_Toc353560719)

[7 Subscription App Tests 7](#_Toc353560720)

[7.1 Model Tests 7](#_Toc353560721)

[7.1.1 Subscription 7](#_Toc353560722)

[7.2 View Function Tests 7](#_Toc353560723)

[7.2.1 Model Tests 7](#_Toc353560724)

[7.2.2 View Tests 7](#_Toc353560725)

# 1 Introduction

## Testing purpose

As a potentially high-traffic internet destination the MyNewMedia site will need to be subjected to a rigorous testing plan. The site will need to be completely bug free before it can be deployed to a general audience. All tests shall be designed with regression testing and automation in mind, so that future developments can be tested against old requirements with ease.

## Scope

There are three main types of tests the MyNewMedia site will use, each corresponding to the Model-View-Template architecture that the Django framework enforces. The model tests will create database objects and query for them, to ensure the database both accepts and returns good data and rejects data that does not conform to the constraints. View level tests will make sure the underlying site functions are behaving as expected. Template level tests will be browser based scripts that ensure HTML and CSS components are being rendered correctly.

## Technologies

All model and view level tests will be programmed as Python unit tests that can be run from the server directly. Template layer tests will be scripted using the Selenium IDE – a plug in for the Firefox web browser. Selenium allows developers to mimic browser interaction in code. These tests can be exported to Python and packaged with the other unit tests, but require a Firefox browser to be installed with the Selenium IDE to be run.

## Overview

The following is an outline of the tests to be performed:

* Core Site Tests – includes Selenium tests and tests of 3rd-party packages
* Channel App Tests
* Profile App Tests
* Recommendation App Tests
* Search App Tests
* Subscription App Tests

# 2 Core Site Tests

## 2.1 Selenium Tests

The following tests shall be scripted using the Selenium IDE and shall require a Firefox web browser to run. Each test references a specific web page behavior in the site.

### 2.1.1 Login Test

The login test verifies the login behavior by logging into the site with a test user that shall be created during the test set up.

### 2.1.2 User Home Redirect Test

The user home redirect test verifies that the test user will be redirected to the User Home page after logging in.

### 2.1.3 Search Test

The script will enter a search for a unique test channel that shall be created during test set up. The search will be performed by looking up a user, channel title, channel description and channel tag. A successful test will return the correct channel in all cases.

### 2.1.4 Subscription Test

This test will subscribe the test user to the test channel. A successful test will update the web page components to reflect the new data.

### 2.1.5 Channel Manager Test

The script will create a channel using the channel form then verify the newly created channel appears on the test user’s Channel Manager page. The new channel will then be deleted. This script will test both the channel form and the Channel Manager functionality.

### 2.1.6 Channel Home Test

The browser will be navigated to the test channel page using a specified URL. This test will verify the dynamic channel URLs are working correctly.

### 2.1.7 RSS Feed Pull Test

On the test channel page, the script will verify the correct feed has been pulled, and the correct elements are present. A test RSS feed XML document has been created for this test.

### 2.1.8 Logout Test

The script will verify the logout link on the upper right menu works correctly.

### 2.1.9 Splash Page Redirect Test

The script will verify the browser is redirected to the splash page on logout.

## 2.2 3rd-Party Package Test

The following tests are designed to verify the 3rd-party Django packages used in the site are working correctly.

### 2.2.1 Messages

This will test the Messages database tables for the Django-Messages packages. A message will be ‘sent’ from one user to another and verified that it ends up in another user’s inbox. The message will then be deleted from the senders outbox and the receiver’s inbox.

### 2.2.2 Tags

This test will verify that tags are correctly assigned to channels and duplicate tags are not created.

### 2.2.3 Registration

This test will create a test user and ensure the information is added to the database correctly and all information is properly encrypted.

# 3 Channel App Tests

## 3.1 Model Tests

### 3.1.1 Channel

The Channel test will verify that added channels are correctly entered into the channel database tables and entries with incorrect data are rejected without saving. It will also verify that the channel pictures are correctly saved in the Media folder where all dynamic content is saved.

### 3.1.2 Link

The Link test will verify that added manual links are correctly entered into the channel database tables and entries with incorrect data are rejected without saving.

### 3.1.3 Feed Item

The Feed Item test will perform a feed pull and insert the returned items in the database. The test will verify that all feed items returned are entered correctly.

### 3.1.4 Feed Tracker

The Feed Tracker test will verify that data is entered correctly into the database and duplicate entries are not allowed.

## 3.2 View Function Tests

Note that the forms used to add channels, edit channels, delete channels and add links will be tested in the browser using Selenium.

### 3.2.1 ‘channelhome’ function test

This test will verify the return the correct channel in the resulting dictionary object when passed a test URL extension.

### 3.2.2 ‘markasread’ function test

Verifies that the correct feed tracker object is inserted into the database when this function is passed a test channel id and feed item id.

### 3.2.3 ‘pullfeed’ function test

Verifies the returned list contains the expected feed items when this function is passed a test feed URL and number of posts to limit the return list.

### 3.2.4 ‘browsechannels’ function test

Verifies the returned dictionary contains all channel types.

### 3.2.5 ‘artisthome’ function test

Verifies the returned dictionary contains all channel objects belonging to the passed user.

# 4 Profile App Tests

## 4.1 Model Tests

### 4.1.1 Profile

The profile tests will verify all profile information is entered correctly. It will also verify that the profile pictures are correctly saved in the Media folder where all dynamic content is saved.

### 4.1.2 Preferences

The preferences test will verify that user preferences are correctly saved in the database.

## 4.2 View Function Tests

### 4.2.1 ‘profilequery’ function test

Verifies that the returned profile will always belong to the user currently logged in.

# 5 Recommendation App Tests

## 5.1 Model Tests

The Recommendation App does not use the model layer.

## 5.2 View Function Tests

### 5.2.1 ‘loadRecommendations’ function test

### 5.2.2 ‘loadOtherRecommendations’ function test

# 6 Search App Tests

## 6.1 Model Tests

The Search App does not use the model layer.

## 6.2 View Function Tests

### 6.2.1 ‘searchresults’ function test

Verifies that when passed a specific search term, the function will return three separate sets of objects in the dictionary. They are the search results based on channel title or description, based on all existing users and based on all existing tags.

# 7 Subscription App Tests

## 7.1 Model Tests

### 7.1.1 Subscription

The subscription tests will verify that a subscription object created will be entered into the database correctly. It will also verify that the deletion of a subscription object will remove that object from all related tables.

## 7.2 View Function Tests

### 7.2.1 ‘subscribe’ function test

Verifies the current user has a subscription created for the passed channel.

### 7.2.2 ‘unsubscribe’ function test

Verifies the current user has an existing subscription, then verifies the function removed that subscription.