Software Test Plan

COSC4426A Project

H R Mandiv

Instructor: Anthony Pagnotta

Algoma University

Project Test Plan (COSC4426A - Software Testing, Quality Assurance, and Release Management)

Sault Ste. Marie, Ontario, Canada

Date 5/29/2019

TABLE OF CONTENTS

1. INTRODUCTION

The Software Test Plan is designed to prescribe the scope, approach and resources. The plan identifies the items to be tested, the features to be tested and the types of testing to be performed.

1.1 Objective

A Test driven development process using Red, Green, Refactor and repeat cycles. Build a Automated Selenium test framework with selenium webdriver and check if the input box components of the form in the home page of the web application works the way it is supposed to. And the integration test checks that when the form is filled correctly and the submit button is pressed, the web site redirects the user to the correct destination. For this tests project, the system test checks if the user can bypass the form fill up process or not to reach the Success page. As the function of the web application is to have a user fill out a form and only upon successful completion of the form the user can get redirected to a success page. The scope of this project is very limited, the only web browser it will be tested on will be Chrome version 74.0.3729.169 (official build)(64-bit).(Therefore the chrome webdriver used in selenium will be chrome driver version 74.0.3729.6). All the tests are run against the production website for this project, as my development, staging and production sites will basically be the same.

1.2 Testing Strategy

Test plan components include:

- Items to be tested,
- Features to be tested,
- Features not to be tested,
- Pass / Fail criteria

1.3 Scope

In scope: -

The only job of the web application is to have a user fill out a form and only upon successful completion of the form the user can get redirected to a success page. The scope of this project is very limited, the only web browser it will be tested on will be Chrome version 74.0.3729.169 (official build)(64-bit).(Therefore the chrome webdriver used in selenium will be chrome driver version 74.0.3729.6). All the tests are run against the production website as for this project, my development, staging and production sites are basically the same. unit tests, integration test and system test are very closely related in this project and the code of the test framework will not be seperated to individually perform unit, integration or system test. They are more conceptual than having a visual representation in the testing framework code. The projects size and with the time available to work on this test plan is not enough to perform a full coverage testing.

Out of scope: -

As the web application will use Asp.net mvc 5 framework, the framework forces quality assurance to a satisfactory level, so the frameworks vulnerabilities will not be tested. No authorization token checks are performed or any network vulnerabilities tests will be performed. The frameworks guidelines will be used to pass data from module to module with the recommended viewModel state validations in the index view page and in the home controller the http post tagged index action method will perform a modelstate validation before redirecting the user to the Success page. The framework is designed to allow the actions mentioned above to pass, if and only if the model binding is valid or else the user is asked to fill the form out correctly again, therefore for the purposes of this project further unit testing on the home controller and the view pages are skipped. The fact that the proposed tests on the production website passes means that all the underlying asp.net mvc framework is functioning the way it needs to for this web application at its current stage.

2. TEST ITEMS

- First name input box
- last name input box
- Submit button
- No access to Success page unless the form has been submitted correctly.

2.1 Program Modules* (Check the link below for further understanding of the mvc pattern that is used for the web app development)

- lifecycletest/Views/Home/Index.cshtml**
- lifecycletest/Views/Home/Success.cshtml***

*https://dotnet.microsoft.com/apps/aspnet/mvc

***https://dev.azure.com/hmandiv/_git/MVCLifecycle?path=%2Flifecycletest%2FViews%2FHome%2FIndex.cshtml&version=GBmaster*

****https://dev.azure.com/hmandiv/_git/MVCLifecycle?path=%2Flifecycletest%2F*

Views%2FHome%2FSuccess.cshtml&version=GBmaster

3. FEATURES TO BE TESTED

- Input form that expects a valid user first name and last name
- The proper redirection to the Success page upon successful completion of the the form by the user.
- Success page will not be accessible unless the user fills out the form in the home page.

4. FEATURES NOT TO BE TESTED

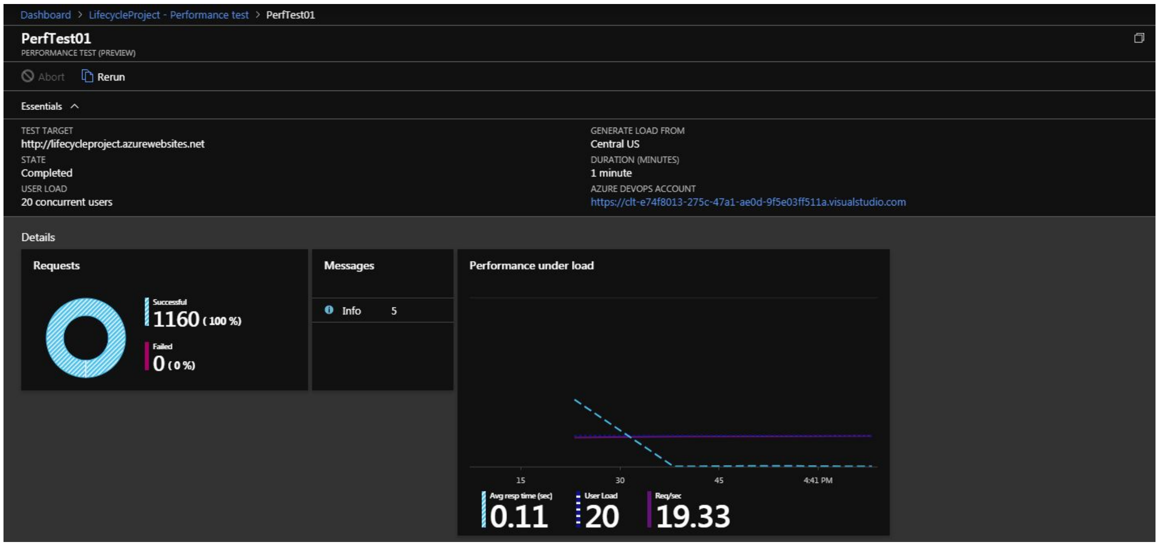
- Asp.net mvc 5 framework and its vulnerabilities
- The MVC pattern, as its assumed that the quality assurance of microsoft's asp.net framework will enforce the quality and proper functionality of the individual components.

5. APPROACH

5.1 Unit/Integration/Interface Testing

1. Go to home page
2. Check if the home was opened correctly or not
3. Fill out the form and submit
4. Check if the success page is visible or not.
5. Go to Success page without filling out the form and the Success page should not be visible.

5.2 Performance Testing



Dashboard > LifecycleProject - Performance test > PerfTest01

PerfTest01
PERFORMANCE TEST (PREVIEW)

TEST TARGET
http://lifecycleproject.azurewebsites.net
STATE
Completed
USER LOAD
20 concurrent users

GENERATE LOAD FROM
Central US
DURATION (MINUTES)
1 minute
AZURE DEVOPS ACCOUNT
https://clt-e74f8013-275c-47a1-ae0d-9f5e03ff511a.visualstudio.com

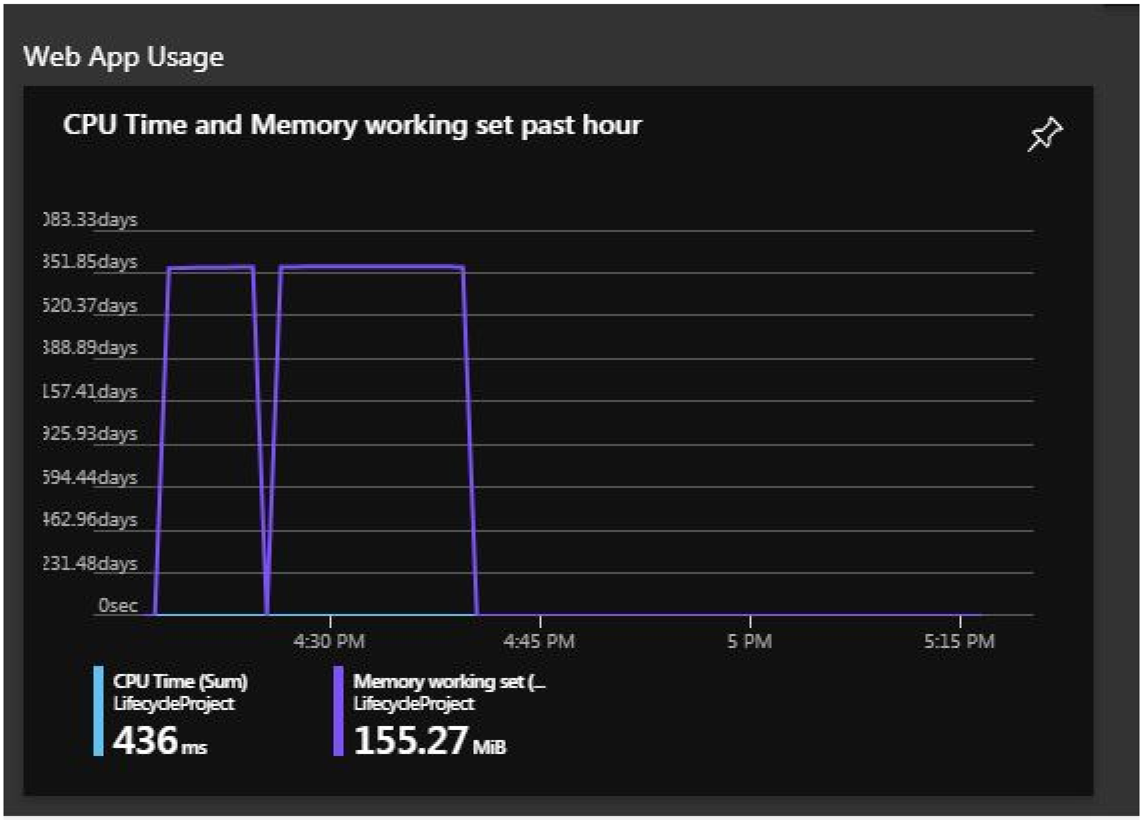
Details

Requests
Successful 1160 (100 %)
Failed 0 (0 %)

Messages
Info 5

Performance under load

Avg resp time (sec) 0.11
User Load 20
Req/sec 19.33



Web App Usage

CPU Time and Memory working set past hour

CPU Time (Sum)
LifecycleProject
436 ms

Memory working set (...
LifecycleProject
155.27 MiB

5.3 PASS / FAIL CRITERIA

- pass if the the assertions returns the expected value and no errors are thrown.
- Fails if assertions returns any unexpected value and an error is thrown.

5.3.1 Approval Criteria

- *All tests must pass*

6. TESTING PROCESS

Test Home Page For Form Fill Out And Submit: -

- Fill out the form in home page and submit
- Check if Success page was shown or not

Test For No Access To Success Page: -

- Go to Success page from home page
- And check if Success page was visible or not.

6.1 Testing Tasks

1. Set Up the selenium Framework in Visual studio 2017
2. Start Red phase
3. Start Green phase
4. Start refactor phase
5. Repeat till the current iteration has reached its goals
6. Start next iteration

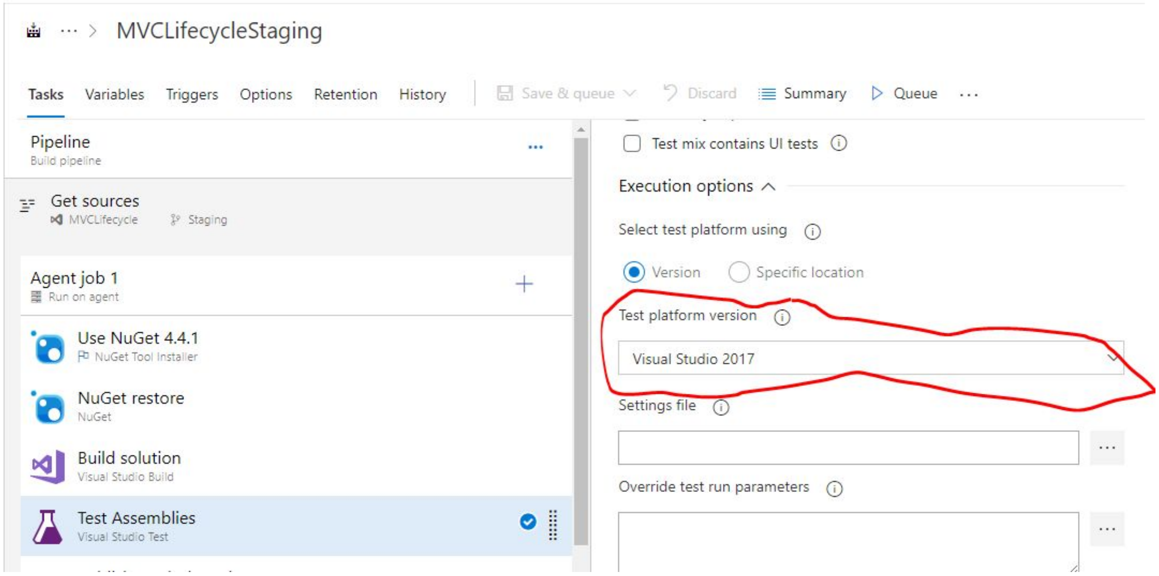
6.2 Test Execution: -

In my laptop: -

- In visual studio 2017 run Unit tests with test explorer.

In Azure devOps Build Pipeline: -

- Use Test assemblies (Visual studio Test)



Pipeline
Build pipeline

Get sources
MVCLifecycle   Staging

Agent job 1
Run on agent

Use NuGet 4.4.1
NuGet Tool Installer

NuGet restore
NuGet

Build solution
Visual Studio Build

Test Assemblies
Visual Studio Test

☐ Test mix contains UI tests

Execution options

Select test platform using

● Version ○ Specific location

Test platform version

Visual Studio 2017

Settings file

Override test run parameters

6.3 **Resources***(includes pipeline integration)

1. Visual studio 2017
2. Unit test (.net framework)
3. Selenium support
4. Selenium Webdriver
5. Chrome driver
6. Windows Web App service

*https://docs.microsoft.com/en-us/azure/devops/pipelines/test/continuous-test-selenium?view=azure-devops

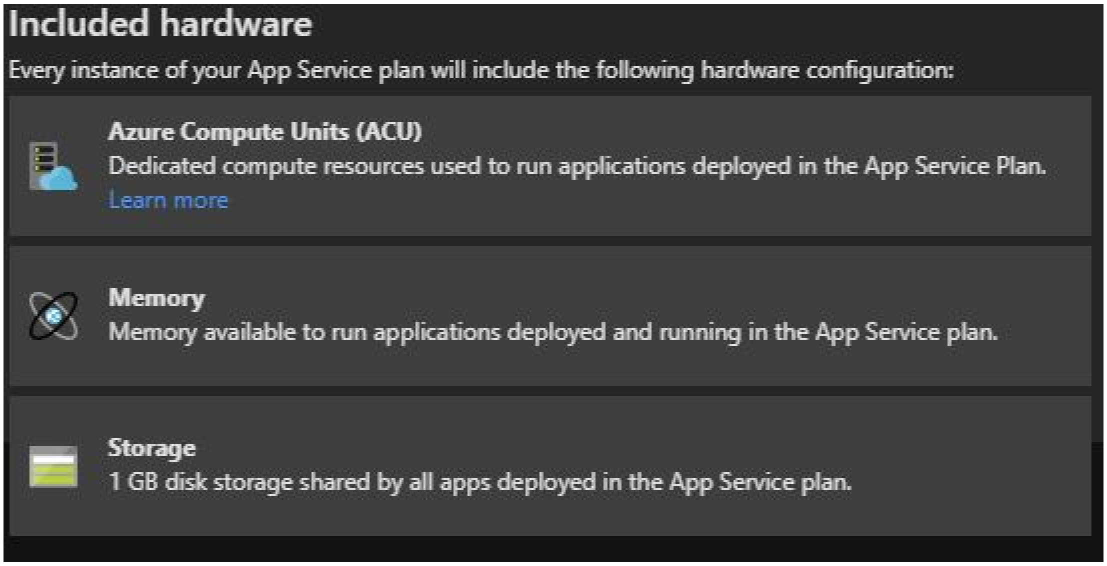
6.4 Software

- Visual studio 2017 unit test project (.net framework) (install the latest versions available for visual studio 2017 from nuget package manager)
- Azure DevOps and Azure portal*
- Operating System will be latest windows version

- App service (Dev/Test)

*https://docs.microsoft.com/en-us/azure/app-service/app-service-web-get-started-dotnet-framework

6.5 Hardware:*



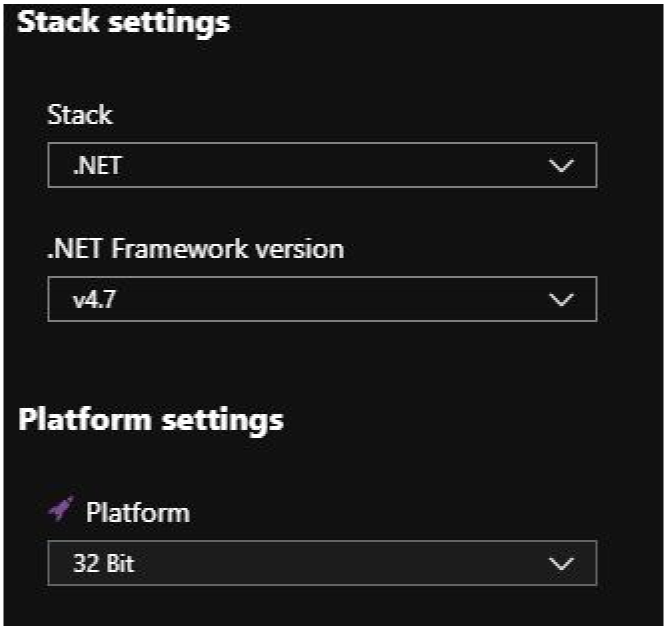
Included hardware

Every instance of your App Service plan will include the following hardware configuration:

Azure Compute Units (ACU)
Dedicated compute resources used to run applications deployed in the App Service Plan.
Learn more

Memory
Memory available to run applications deployed and running in the App Service plan.

Storage
1 GB disk storage shared by all apps deployed in the App Service plan.



Stack settings

Stack
.NET

.NET Framework version
v4.7

Platform settings

Platform
32 Bit

**https://docs.microsoft.com/en-us/azure/virtual-machines/windows/*