Testing Learning

Workgroup number: E8.01
Repository: https://github.com/javiervaz01/Acme-Toolkits
Date: 2022/05/26

Name	Corporate e-mail
Carrasco Núñez, Alejandro	alecarnun@alum.us.es
Durán Terrero, Andrés	anddurter@alum.us.es
López Benítez, Pablo Delfín	pablopben@alum.us.es
Núñez Moreno, Pablo	pabnunmor@alum.us.es
Robledo Campa, Pablo José	pabrobcam@alum.us.es
Vázquez Monge, Francisco Javier	fravazmon@alum.us.es

Table of Contents

Executive summary 3

Revision table 4

Introduction 5

Contents 6

Conclusions 7

Bibliography 8

Executive summary

The following document consists of an explanation of what we know about testing a Web Information System after taking the subject of Design and Testing II, from this degree of Software Engineering.

Revision table

Revision number	Date	Description
v1	2022/05/26	Initial version

Introduction

It has already been taught to us in this course the importance of testing a Web Information System, not only because of making sure that our code is right, but for keeping it simple, eliminating defects or guaranteeing quality.
We have learned that we can envision testing from 2 approaches: a functional and performance way.
The **functional tests** detect failures by comparing the actual results of a request to its expected results.
On the other hand, with the **performance testing**, we can check if performance data deviates from the expectations. This approach requires performing statistical analysis at a given confidence level.
Throughout the contents, we will focus on **functional testing**.

Contents

When testing a Web Information System we had to face different concepts:
- **SUT** (System under test). In the general theory of testing, the SUT ranges from a method in a class to a full system. Our application, **Acme Toolkits** is the SUT, and we test it feature by feature.
- **Fixture**. It consists of a web of artefacts that will be needed to set up for the SUT to be tested. The fixture ranges from an object to a full system. Our **database** is the fixture.
- **Mocks**. It is a fake object in the fixture, and it is implemented as a table in which some selected inputs are mapped onto their corresponding outputs. However, we have not used this in our project.
- **Positive test cases**. It is a script that checks that the SUT works well in a context in which it is expected to work well. Working well means that the SUT does not issue any errors or exceptions and produces the expected results.
For instance, we implemented a positive test case from the feature any.chirp, in which any user (no need to be logged in), could create a Chirp.
- **Negative test cases**. It is a script that checks that the SUT works well in a context in which it is expected to report an error.
For instance, we implemented a negative test case from the feature any.chirp, in which any user (no need to be logged in), created a Chirp with wrong input values.
A negative test case can also be a script that checks that the SUT works well in a context in which it is expected to panic. For instance, we implemented a hacking test from the feature administrator.systemconfiguration, in which an anonymous user, an inventor and a patron tried to show the system configuration, which can only be accessible by the administrator.
- **Test class**. It is a collection of related test cases that is encapsulated in a single class. Typically, a test class encapsulates a positive and possibly a negative test case regarding a particular feature.
- **Test suite**. It is a collection of test cases for a particular object.

Our approach to functional testing is called E2E (end-to-end). It attempts to test software “naturally” since the goal is to simulate a user interacting with the web interface, while checking that the system performs as expected.
The test cases consist of scripts that use the Gecko driver to simulate a user interacting with Firefox (clicking on options, buttons, and submits) and reading the results returned by your application to check that they are the expected ones.

Conclusions

After having taken this subject we believe that we have acquired knowledge about testing a Web Information System, that we did not have before. Most of us had never implemented a negative or a hacking test case, and now we consider them a very useful approach. We hope to have achieved the expected goals regarding testing, however we seek learning more. Therefore, we hope to become real experts in the field of testing and learn everything about it, not only the basics, in our future.

Bibliography

Intentionally blank.