Oskari Vuori, Riku Välikangas

Test plan for an E-commerce store

Group assignment

COMP.SE.200 Software Testing

10/2022

Table of contents

[1. Introduction 1](#_Toc117074352)

[2. Scenarios 2](#_Toc117074353)

[3. Tools 3](#_Toc117074354)

[4. Tests 4](#_Toc117074355)

[5. Virtual Machine 5](#_Toc117074356)

[References 6](#_Toc117074357)

Definitions, acronyms and abbrevations

* List of any definitions, acronyms and abbrevations used in the document.

.

# Introduction

* Short introduction to the contents of the document.
* What is in the document?
* What is the purpose of the document?

# Scenarios

* The descriptions of your end-to-end scenarios.
* What components of the E-commerce store will be tested, how and why?
* What tools will be used for the testing (if you can think of any based on the provided documentation)?
* A (UML) diagram of your scenario. The diagram can be, for example, an activity or sequence diagram, whatever feels most natural. The diagram can be high-level (abstract). If you know which tools are should to be used, describe how they are used in relation to your diagram.
* List the source code files you selected for your unit tests from the provided utility [library](https://github.com/otula/COMP.SE.200-2022-2023-1). Use a prioritization method to validate your file selections. Remember your end-to-end scenarios.

# Tools

* Description of the chosen tools, packages, libraries, etc. including possible links to web sources (tutorials, homepages, etc). Why did you select these tools and how did you test them?
* How would the tools be used to test your end-to-end scenarios?

# Tests

* What kind of tests are to be performed (unit, integration, …), and what is their importance for the end-to-end scenarios you described?
* What components of the E-commerce store will be tested and what will *not* be tested, and why?
* How will the test results be documented?
* If bugs or issues are found, how are they classified or categorized? Add a template of you bug/issue report if needed.
* When the tests are considered "passed"?
* How to guarantee adequate code coverage?

# Virtual Machine

* Optionally, the description of your Virtual Machine setup with screenshots (e.g. of a tool test you are running), if you want to try your luck in getting the bonus point.
* Setting up a virtual machine is optional, but using one can award you 1 bonus point, and also save you a lot of trouble if something goes irreparably wrong with your testing environment, or you do not want to mess up your working "every day setup" with extra software. You can freely choose your virtual machine platform. [Oracle's VirtualBox](https://www.virtualbox.org/) is a popular free choise, but you can also use, for example, VMWare or Amazon AWS. Any recent Linux distribution (e.g. Ubuntu, Debian/testing) should work with the newest testing tools. The provided library is tested with NodeJS 14, so picking that version is a safe bet, though newer versions will probably also work.
* To get the bonus point you should add a brief description of your environment with a screenshot of your running setup into your testing report.
* Note: you can use other virtualization tools (e.g. Docker) in you tests if you like, but this is not required, and will not give you the extra point.

# References