Application Testing Methods

Benjamin Roberts

Level 4 Software Development

Contents

[Introduction 4](#_Toc174202491)

[Testing – Theory Review 5](#_Toc174202492)

[Types of Testing 5](#_Toc174202493)

[Frameworks 5](#_Toc174202494)

[Methodologies 5](#_Toc174202495)

[Workplace Findings 5](#_Toc174202496)

[Testing – Applied 6](#_Toc174202497)

[Scenarios 6](#_Toc174202498)

[Test Code created 6](#_Toc174202499)

[Conclusion 7](#_Toc174202500)

[References 8](#_Toc174202501)

**No table of figures entries found.**

# Introduction

From reading the brief, the idea that comes to mind for me is:

* Do a theory review of the various Testing Frameworks and Methodologies (with references)
* Write details on which Frameworks and methodologies are used within my team
* Expand on this via reaching out to our dedicated IT Testing teams, who I know engage in more testing methodologies then my team

Then, put together a Test Plan for testing a deployed version of an implementation of our Credit Decisioning system. I can cover:

* System Testing via the Postman software, which makes use of JavaScript test cases via an implementaion of the Chai libary for running test cases on the HTTP response recived from a web application
* Integration testing with integrated 3rd party API's in the Credit Decisioning system
* Performance Testing via testing peak volumes the system can handle
* Possibly secuirty testing (not something we do in my team today, but I'm hopeful I could apply something with the support of our dedicated IT Testing teams)

(Note: would need to censor alot of sensitive information with this idea, possibly even specific parts of the written test code, but this would be a great idea to apply my learnings I feel.)

# Testing – Theory Review

## Methodologies

### Functional

* Unit Testing
* Integration Testing
* System Testing
* Acceptance Testing (UAT)

### Non-Functional

* Performance Testing
* Security Testing (Penetration)
* Usability Testing
* Compatibility Testing

## Frameworks

* JUnit (Java)
* NUnit (C#)
* Chai (JavaScript – Node)
* Selenium (Web Apps)
* Pytest (python)

## Workplace Findings

## Theory Reflections

# Testing – Applied

## Pre-Deployment – Unit Tests

* Example of WR Unit Test, not much needed, just demonstrate get the concept

## Scenarios

* System Testing (Postman)
* Integration Testing (test Delphi and TAC Integrations work)
* Performance (peak volunes and response times)
* Security (support from IT)

## Test Code created

# Conclusion

# References

<https://smartbear.com/learn/automated-testing/test-automation-frameworks/>

<https://smartbear.com/learn/automated-testing/software-testing-methodologies/>