

|  |  |
| --- | --- |
|  | Fundamentals of Software Testing Assignment |
|  |  |
|  | Elise Callus – xxxx96(M) | Jonathan Borg – 63396(M) | Karl Farrugia – 59796(M)  Cps3230 – Fundamentals of Software Testing  B.Sc(Hons) Computing Science |

Table of Contents

[Introduction 2](#_Toc471298787)

[Overview of the Problem 2](#_Toc471298788)

[Task 1. Unit testing and Test Driven Development 3](#_Toc471298789)

[Task 2. Cucumber and Automated Web Testing 3](#_Toc471298790)

[Task 3. Model Based Testing 3](#_Toc471298791)

[Task 4. Performance Testing 3](#_Toc471298792)

[Conclusion 4](#_Toc471298793)

# Introduction

The assignment was tackled on one computer with one person coding and one person overviewing the code, and at a later point switching roles. A git repository was created; this repository was used to store the current state of the program. Apart from this, the repository was used to revert changes and keep track of all the progress done on the project (<https://github.com/jonathanborg/Cps2002Assignment>), the completion of task 1 was tagged in github at build 99 and this is accessible in the following link (https://github.com/jonathanborg/Cps2002Assignment/releases/tag/Part\_1\_Completion). This same git repository was synched with Jenkins, for continuous integration of the code (https://jenkins-ict.research.um.edu.mt/job/KarlJonathanAssignment/). In Jenkins, an Emma Code Coverage was used to follow the amount of testing done on the entire project. The coverage is expected to fall slightly short of 100%, due to some methods are set internally and thus the user has no access to them. The second part of the assignment was tackled by adjusting the code using different design pattern technique to adhere to the new requests as indicated by the user and afterwards getting the coverage as close to 100% as possible. Overall the assignment was managed well between the two members, with each member having his own strong suit to offer for the overall completion of the task.

## Overview of the Problem

The task at hand was to create a library which allows user to borrow books. The system implementation consists of five classes: Library, Catalogue, User, Book, and Genre. These five classes shall be accompanied by a Junit test class, in which all the classes shall be tested for. This Testing shall see that all the necessary requirements specified by the needs of the client are seen to be met, and tested for soundness.

The library class shall have instances of catalogue and User within it, the library will thus have access to the book class via catalogue. This is due to the fact that the catalogue class has the instances of books within it and provides all the necessary methods regarding books to the library class. Finally, Book class will have instances of genre and user within it. The Book, Genre and User classes shall contain all the information of the respective objects, while Library and Catalogue shall contain all the methods to edit the formerly mentioned objects, and thus meeting the requirements of the clients in the process.

For testing purposes the majority of the methods are set with a return type of Boolean instead of void. This does not affect the method in any way, but eases testing procedures in the Junit testing.

The entire system abides by the first rule “There is no need to persist any of the system’s data to permanent storage.”. This is due to the fact that it does not save any data to persistent storage but stores the current data in memory.

# Task 1. Unit testing and Test Driven Development

The main objective of this task was to build and unit-test the class diagram provided. This diagram was modified and altered to provide better functionality of the classes and variables, while ensuring maximum test coverage.

Figure one shows the new modified class diagram describing the system implemented in our project.

# Task 2. Cucumber and Automated Web Testing

# Task 3. Model Based Testing

# Task 4. Performance Testing

# Conclusion

Both parts of the assignment are seen to be completed successfully to the best of our ability, meeting all the requirements specified. Continuous integration was utilised and this may be seen from the builds in Jenkins and from the Jenkin’s Emma Code Coverage. Well though planning has been put into both parts of the assignment and these could be noted within the documentation itself. All deliverables have been mentioned and given in the assignment but can also be found listed hereunder:

1. Details of your environment setup (Github repository URL and Jenkins project name)

* Github repository URL: <https://github.com/jonathanborg/Cps2002Assignment>
* Jenkins project name: KarlJonathanAssignment
* Jenkins project URL: <https://jenkins-ict.research.um.edu.mt/job/KarlJonathanAssignment/>

1. The Github tag and Jenkins build number at the point where the initial setup was finished.

* Github tag URL: <https://github.com/jonathanborg/Cps2002Assignment/releases/tag/Part_1_Completion>
* Jenkins Build Number: 99

1. The Github tag and Jenkins build number at the point where the system was completed.

* Github tag URL: <https://github.com/jonathanborg/Cps2002Assignment/releases/tag/Assignment_Completion>
* Jenkins Build Number: 123