

ArtemisLite Technical Report

*Queen’s university Belfast - Software Development Masters*

*Software Engineering – CSC7053*

*Group 6*

*Authors*

* *Caolán Egan -*
* *Ronan Crossan -*
* *Ryan Bowman -*
* *Órfhlaith Woods – 40134853*

*Date of Submission:*

*Word Count:*

Table

Description automatically generated

Contents

[Abstract 3](#_Toc92916176)

[Keywords 3](#_Toc92916177)

[Introduction 3](#_Toc92916178)

[Requirements Analysis **– 30 MARKS** 3](#_Toc92916179)

[Realisation **– 20 MARKS** 3](#_Toc92916180)

[Design **– 20 MARKS** 3](#_Toc92916181)

[Test **– 5 MARKS** 3](#_Toc92916182)

[Bibliography 4](#_Toc92916183)

[Appendix 1 4](#_Toc92916184)

[Appendix 2 4](#_Toc92916185)

[Appendix 3 4](#_Toc92916186)

**No table of figures entries found.**

# Abstract

100 words on finalised product

## Keywords

Agile, Boundary value analysis (BVA), Eclipse, Java, White box testing, Unified modelling language (UML),

# Introduction

Brief overview of project aims and objectives

# Requirements Analysis **– 30 MARKS**

* One Master UML diagram of whole system
* Use case descriptions (all steps in the use case scenarios must be in the master UML)
* Description of sequences/actions should be focused on in this chapter
* Problem Domain – list problems (two players with same name etc.)and solution in place (alternative flow in UML diagram)
* Layout of virtual board game (text layout only not GUI)

# Realisation **– 20 MARKS**

* Multiple UML diagrams with brief commentary
* HOW it works and WHY we chose these use cases to illustrate
* UML should show HOW software components make method calls to each other and interact with the players
* WHY have you chosen the use cases that you have?

# Design **– 20 MARKS**

* UML diagram that describes system components
* UML should closely illustrate the java code for our game
* Show the classes and methods which support the sequence of method calls
* Benefits of the design we’ve chosen
* How does our design provide a platform for seamless updates to the game/code?
* Flexibility, maintainability and extensibility

# Test **– 5 MARKS**

* Sample from test plan
* Brief description of your approach
* Black/White box
* Micro-Services possibility?

# Bibliography

**There are no sources in the current document.**

# Appendix 1

* Full extent of the test plan
* Acceptance tests
* Results from testing
* Errors found in testing
* BVA – due to resource of time
* Acceptance testing
* JUNIT Testing
* White box testing
* Testing Levels
* Integration testing
* System Testing
* Regression testing

# Appendix 2

* Team minutes

# Appendix 3

* Evidence of GitLab/GitHub

# Appendix 4

* Evidence of day to day project management
* Tasks for team members
* Timelines
* POA
* Meeting schedule
* Screen dumps of sprints
* Notes from meetings