

**ASSIGNMENT COVER SHEET**

**Student’s Name: Ken Kilmartin & Catharine Phoenix**

**ID Number: B00091252 & B00091705**

**Course: Bachelor of Science (Honours) in Computing**

**Year: 3rd Year**

**Lecturer: Orla McMahon**

**Title of Assignment: Assignment 2**

**Due Date: 9th April**

**Date Submitted: 9th April**

The material contained in this assignment is the authors original work, except where work quoted is duly acknowledged in the text. No aspect of this assignment has been previously submitted for assessment in any other unit or course.

Signed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Date: \_\_\_\_\_\_/\_\_\_\_\_\_/\_\_\_\_\_\_

Signed: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Contents

[**Introduction** 1](#_Toc510280750)

[**Methodology** 2](#_Toc510280751)

[**UML** 3](file:///C:\Users\Ken\Desktop\Object%20Orientatated%20design%20paterns\Assignment%202\Report%20(3).docx#_Toc510280752)

[**Bibliography** 4](#_Toc510280753)

# **Introduction**

The Four design patterns we decided to implement within our project are as follows:

* The Singleton design pattern.
* The Abstract Factory design pattern.
* The Adapter design pattern.
* The Command design pattern.

We made our choice based on the design patterns we had thought would suit our purpose best for this assignment. We had also considered the Builder design pattern instead of the Command pattern due to the command creating a few more classes but we had both wanted to look at the Command in more detail. The group also decided to make use of GitHub for this assignment due to the time period we had to complete this assignment was over the Easter break. Our repository can be seen at:

<https://github.com/KenKilmartin/oodpAssignment2>

# **Methodology**

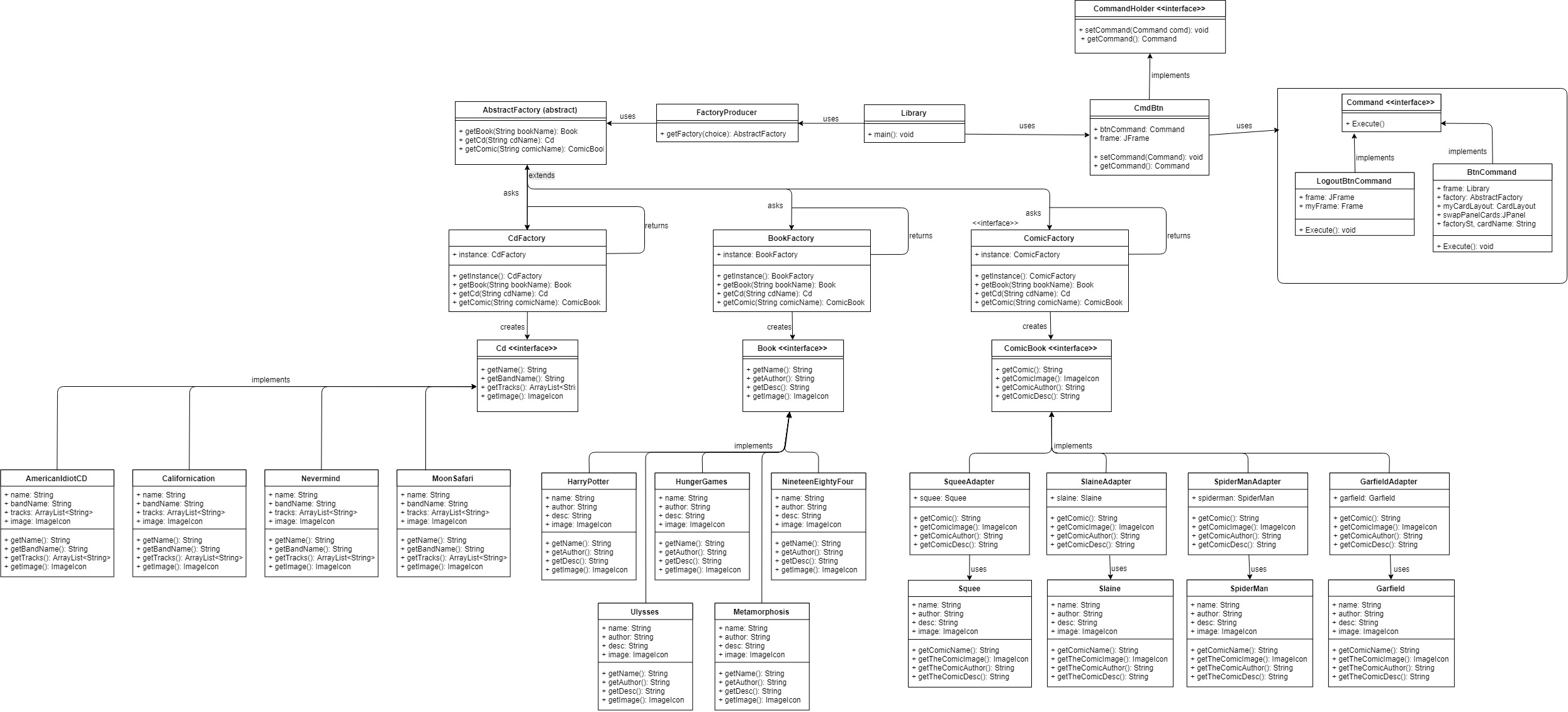
We had decided to build a program for a library to use. We choose to build a library program as we had thought the Abstract factory and Singleton design patterns would be a good fit to try use these design patterns. We then had thought if the library was donated a bunch of comics we could try making an adapter for the comics to fit in.

We used the Abstract factory to encapsulate a group of individual factories such as books, CDs and comic books. We choose these as they all share a common theme. We choose to do this design pattern as we both felt that in the first assignment we had it mixed up a little. This time we made our abstract factory and then had each factory look’s after a single class compared to the first assignment we had all classes in the one factory.

Each of the comics then uses an adapter design pattern to make it fit into the library program. The thought behind this was a comic book shop donated a bunch of old comics to the library and gave the database so then the library had to make adapters for each of the comics to make it fit into their program. We can see that for example on the Garfield comic it does not implement anything. An adapter was made to make it fit around Garfield called Garfield adapter and then it could go into the library system.

We had originally thought about not trying to use the Command design pattern as we had thought that it would just add too many classes to the project. But then had realised we could just apply this pattern to just one of the interface. We use this pattern in the library when you are deciding on which product you want to rent be it Book, CD, Comic or Logout.  
  
The Singleton design pattern was used on each of the factories (Book, CD, Comic). We did this first by making it a private static object ensuring that there was a single object of the factory being made, then we made a constructor private so that this class cannot be instantiated and then we get the only object that is available.

It was a challenging to try think of a project to try fit the design patterns into but as soon as we had figured out what we would fit them into it was easier to program it. We both feel in real world situations it may be easier to apply patterns into problems.



# **UML**

# **Bibliography**

*Design Pattern - Abstract Factory Pattern*. (2018). Retrieved 3 12, 2018, from Source Making: https://sourcemaking.com/design\_patterns/abstract\_factory

*Design Pattern - Adapter Pattern*. (2018). Retrieved 3 12, 2018, from Source Making: https://sourcemaking.com/design\_patterns/adapter

*Design Pattern - Command Patern*. (2018). Retrieved 3 12, 2018, from Source Making: https://sourcemaking.com/design\_patterns/command

*Design Pattern - Singleton Pattern*. (2018). Retrieved 3 12, 2018, from Source Making: https://sourcemaking.com/design\_patterns/singleton