## Report for Software design & development Year 2 - C.A 2

Student Number: **N00181859**

Name: **Jonathan Berkeley**

# Class diagram

The class diagram is included in the docs folder. It’s very slightly outdated, I cannot update it as I do not have visio and online editors I found do not accept .vsdx files to edit.

The change, however, is minor. The Display interface in the class diagram was broken up into two different interfaces, GetSets and Display. The Get and Set methods were moved into the GetSets interface, and the displayAll method stayed in Display interface.

# Checklist

**Working functionality:**

Create bicycle (on loop) – (Create sub-class)

Create bicycle accessory (on loop) – (Create sub-class)

Read stored bicycle object – (Read sub-class)

Read stored bicycle accessory object – (Read sub-class)

Read product(s) by store ID - (Read by ID for the 1 - many)

Check existance of product by ID

Check existance of store by ID

Create store (on loop) – (Create superclass)

Comprehensive debug mode for detailed error messages

RangedRandom util – Working ranged pseudorandom number generator which is used by the DatabasePopulator

**Non-working functionality:**

DatabasePopulator util – Worked at the start for a java object representation but I added more things and didn’t have time to update it for the 1 – many database.

**Other:**

TesType class is not relevant to the project but I left it in anyway. Used for testing generics and type references.

This project has a Github page! (it was private until late today):

<https://github.com/JonathanBerkeley/sdd_CA2>

# Known bugs

Minor bug when using looped sub-class creation: sometimes when entering ‘y’ to continue entering more data to the database, the program will skip waiting for the user’s response and restart the form. This is due to a bug with the Scanner object taking in newlines as a response. I added some newline catching statements to prevent this, but it may still occur.

Minor bug when entering in a very long string or number into the database, it will cause a caught SQLException when attempting to insert it into the database table. Not a big concern since this couldn’t really happen accidentally, the user must purposely do it.

No bugs required documentable usage of the debugger – therefore I do not have any journal entries for this. There is thorough exception catching throughout the program with an option to enable/disable the exception message printing in the program which proved useful multiple times during development.