|  |
| --- |
| MSC SOFTWARE DEVELOPMMENT |
| Advanced Object Technology |
| Assignment 2 |
|  |
| **Jimmy Collins – R00145569 – Jimmy.Collins@mycit.ie** |
| **5/2/2017** |

|  |
| --- |
|  |

Contents

[Section 1 – Introduction 2](#_Toc480384189)

[Section 2 – Demonstrated Usage of Defined Object Technologies 3](#_Toc480384190)

[Template Pattern 3](#_Toc480384191)

[Proxy Pattern - Remote Proxy 4](#_Toc480384192)

[Usage of Database for Data Persistence 5](#_Toc480384193)

[Façade Pattern 6](#_Toc480384194)

[Section 3 – System UML Diagram 7](#_Toc480384195)

[Section 4 – Screen Captures of Most Impressive Code Snippets 7](#_Toc480384196)

[Section 5 – Evaluation of Work 7](#_Toc480384197)

[Appendix 1 – Database Setup Instructions 7](#_Toc480384198)

[Appendix 2 – Possible Development Environment Certificate Issue 7](#_Toc480384199)

[Appendix 3 – Code Commit Analysis 7](#_Toc480384200)

[Contributions 7](#_Toc480384201)

[Code Frequency 7](#_Toc480384202)

## Section 1 – Introduction

TODO

TODO – Improvements on Assignment 1 code

* UI changes to split out general / advanced UI
* Changes to strategy pattern implementation so it just returns a chart - decoupled it from any particular UI element (should have done this in the first assignment – it was tightly coupled to a ‘tab’ element).
* Refactored some code using the Template pattern – see section 2.

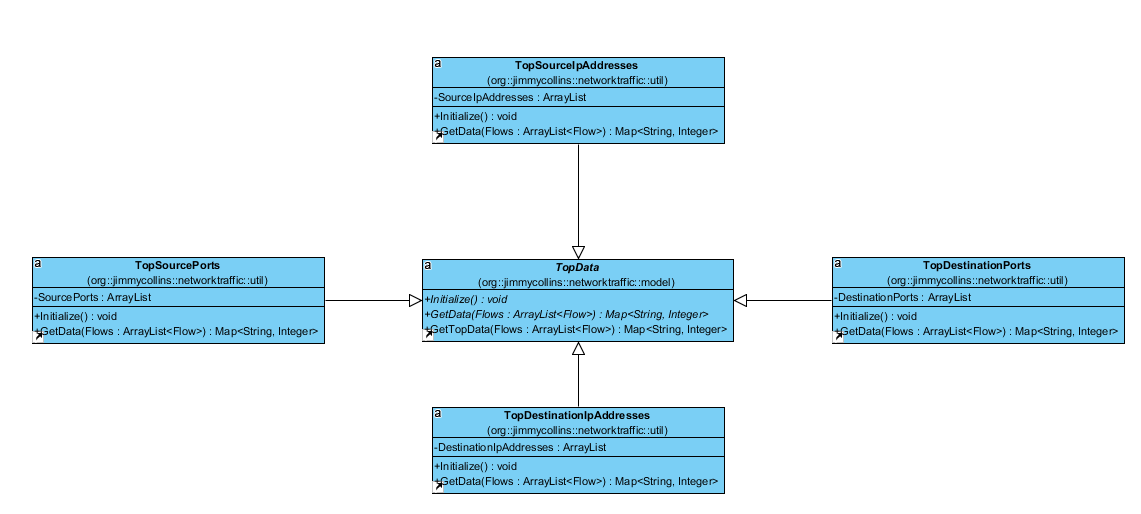
TODO

*Figure 1: The main application user interface.*

## Section 2 – Demonstrated Usage of Defined Object Technologies

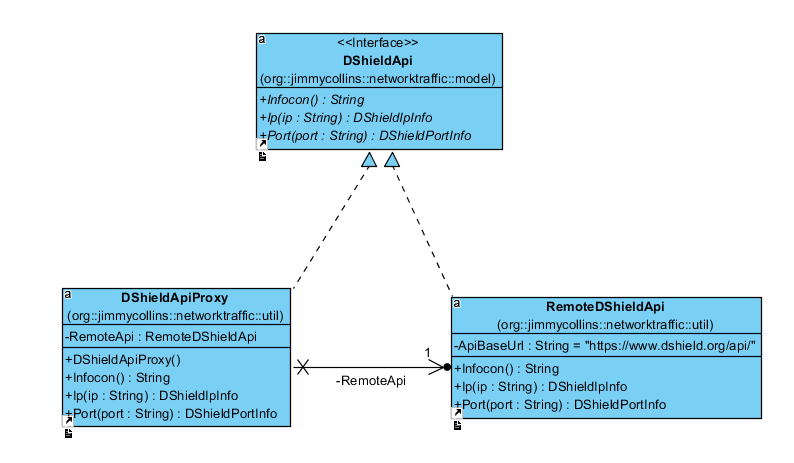
### Template Pattern

I used the template pattern to refactor some code from Assignment 1, which is the logic that is used to generate the top source/destination ports and top source/destination IP address data from the data parsed. Now the TopData class exists, which is an abstract class with the final method GetTopData, and the helper methods Initialize and GetData which are overridden in each of the concrete classes.



### Proxy Pattern - Remote Proxy

The usage of a remote proxy is demonstrated in UML below. In my application I integrate with the DShield[[1]](#footnote-1) API maintained by the Internet Storm Center (ISC). I decided to approach some of the advanced analysis requirements from a security perspective, so I use this API to return some security information on the top IP addresses and ports that are being parsed from the Netflow file.



### Usage of Database for Data Persistence

TODO – Include some information on how I used the Singleton Pattern and how this ties in with my implementation of the Façade Pattern.

<TODO – UML DIAGRAM>

### Façade Pattern

I used the Façade Pattern to refactor my database connection functionality after it had been written the code was quiet complex before this. I ended up creating a new class, DatabaseUtil, which is a wrapper that contains a set of members that are used for database operations that are easily understood and simple to use. These members access the database on behalf of the facade user, hiding the implementation details.

<TODO – UML DIAGRAM>

## Section 3 – System UML Diagram

TODO

## Section 4 – Screen Captures of Most Impressive Code Snippets

TODO

## Section 5 – Evaluation of Work

TODO

## Appendix 1 – Database Setup Instructions

TODO

## Appendix 2 – Possible Development Environment Certificate Issue

TODO

## Appendix 3 – Code Commit Analysis

### Contributions

### Code Frequency

1. <https://www.dshield.org> [↑](#footnote-ref-1)