Sidharth Sharma
ID 40048687

Deliverable 3

SE 6011
August 9, 2019

ProblemFive
***ReviewedFunction* − *F*8**
***Reviewer*(*Self*)*GithubID* :** −`https://github.com/Siddy234567/SOEN6011SidharthSharma.git`
***F*8(*shared*)*GithubID* :** −`https://github.com/samanehshirdel/processes`

1 Code Reviewing

1.1 Overview

Code review is basically a technique to check the quality of the code, keeping in mind the certain aspects[1]. In this document, F5 function is being reviewed through certain guidelines. Reviewed code is checked both manually and automatically as well. For programming style conventions, Checkstyle tool is taken into consideration where as other guidelines are checked manually. Mentioned below are the highlights of the code analysis done[2].

1.2 Programming style

1. When checked manually, some programming style conventions were followed.
2. On running through Checkstyle tool, almost nine hundred and thirteen code violations were observed ranging from incorrect indentation, wrongly processed whitespaces, inclusion of tab characters and the list continues.
3. It was also observed that the class had more than seven hundred lines of code, thereby exceeding the genuine limit of two hundred LOC[3].
4. Numerous constant declarations were also seen.

1.3 Documentation

1. Some meaningful documentation could be seen.
2. No java docs available as agreed upon by the team.
3. Even though there is some documentation, but hardly makes proper description.

1.4 Quality Attributes

1. Except the main function, the other methods fail to provide meaningful explanation, thereby affecting **readability**.
2. Exception handling has been used, there by making it **reliable**.

3. It has been observed that the **usability** offered adds to the quality aspect.

4. Separate testing functions make the code easy to test, adding a plus point for **testability**.

5. Since the code is bundled up in a single class file, henceforth it is tough to analyse, thereby affecting **maintainability**.

1.5 Design Principles and Architecture

1. Would have been good to see some design pattern implementation, since the code was big.

2. Fails to implement any SOLID principles.

3. Methods can be split further, thereby achieving segregation.