**Acceptance Testing**

**UAT Execution & Report Submission**

|  |  |
| --- | --- |
| Date | 20 November 2022 |
| Team ID | PNT2022TMID39178 |
| Project Name | Project – Web Phishing Detection |
| Maximum Marks | 4 Marks |

# Purpose of Document

The purpose of this document is to briefly explain the test coverage and open issues of the Web Phishing Detection project at the time of the release to User Acceptance Testing (UAT).Defect Analysis

Thisreportshowsthenumberofresolvedor closed bugs at each severity level, and how they were resolved

User Acceptance Testing (UAT) is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. UAT is done in the final phase of testing after functional, integration and system testing is done. The main Purpose of UAT is to validate end to end business flow. It does not focus on cosmetic errors, spelling mistakes or system testing. User Acceptance Testing is carried out in a separate testing environment with production-like data setup. It is kind of black box testing where two or more end-users will be involved.

# Test Case Analysis

This reports how the number of test cases that have passed, failed, and untested

**UAT is performed by** –

∙ Client

∙ End users

Phishing detection is done using 16 different heuristic rules. In the system, 11 main classes were defined, and 1 class was defined with 5 sub-classes. This covers all 16 heuristic rules. To test the system, 15 test cases were designed using assertion methods. Ten test cases were designed to test the 10 main classes and 5 test cases were designed to test the class with five sub-classes. The getter-setter method was used to test the class with five sub-classes. The getter method is used to obtain or retrieve a variable value from the class, and the setter method is used to store the variables. The class with five sub-classes checks the 5 different heuristic rules, length of the URL, number of dots and slashes in the URL, presence of @ symbols in the URL, IP address mentioned in the URL, and the presence of special character such as ',', '\_', ';' in the URL. Initially, only a single test case was created for the class with five sub-classes, but it was failing as this class has five methods. After applying the getter setter method, all the test cases passed without any issues. The test results are shown in assert Not Null() is used to check if the input URL is not empty, and assert Array Equals() is used to compare the result from the detection method with the expected result.