|  |  |
| --- | --- |
|  | *Code Inspection Report*  *Anti-Spam Configuration Software Development Project*  BSc in LEI  Academic Year 2017/2018 - 1º Semester  Software Engineering I  Group 22  72973, Ricardo Santos, EIC1  72898, Diana Lopes, EIC1  73529, Kevin Corrales, EIC1  73541, Miguel Rodrigues, EIC1  ISCTE-IUL, Instituto Universitário de Lisboa  1649-026 Lisbon  Portugal  November 25th 2017 |

**Table of Contents**

[Introduction 3](#_Toc498465002)

[Code inspection – Name of the component being inspected 3](#_Toc498465003)

[Code inspection checklist 3](#_Toc498465004)

[Found defects 3](#_Toc498465005)

[Corrective measures 3](#_Toc498465006)

[Conclusions of the inspection process 3](#_Toc498465007)

# Introduction

The Analyser class has the objective of calculate the false negatives and false positives. This class has the methods that count either false positive and negatives and the respective percentage.

The Email class has the goal of representing an Email. An Email can be also considered as spam. Its attributes are the identifier and the list of rules assigned to that Email.

The File Reader class reads the file that contains the spam rules and creates a list with those rules, reads the file that contains the spam rules, using the weights as arguments, read the files that contain the e-mail or spam and creates a list with those e-mails or spams, converts a file into a vector of doubles, checks which of the lines represents the major approximation to a professional e-mail, save the values given through a list of strings and saves them in a file.

The Rule class has the goal of representing an Email rule. Its attributes are the identifier and weight.

The Interface\_Window class represents the interface and its components.

The Message class indicates an error during the programm execution.

# Code inspection – Name of the component being inspected

The Analyser class has the objective of calculate the false negatives and false positives. This class has the methods that count either false positive and negatives and the respective percentage.

|  |  |
| --- | --- |
| *Meeting date:*  *Meeting duration:*  *Moderator:*  *Producer:*  *Inspector:*  *Recorder:* | *??/??/2017*  *?? minutes* |
| *Component name (Package/Class/Method):* |  |
| *Component was compiled:* |  |
| *Component was executed:* |  |
| *Component was tested without errors:* |  |
| *Testing coverage achieved:* |  |

# Code inspection checklist

# ,Found defects

Identify and describe found defects, opinions and suggestions.

|  |  |  |  |
| --- | --- | --- | --- |
| **Found defect Id** | **Package, Class, Method, Line** | **Defect category** | **Description** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| ... | ... | ... | ... |

# Corrective measures

*Found defect Id, how/when/who will correct the identified defect.*

# Conclusions of the inspection process

*Quality assessment of the component inspected for the purpose of integration/delivery the software (does it need no changes, minor/major changes/corrections, build from scratch).*