Text

Description automatically generated with medium confidence

LOG3430 -Méthodes de test et de validation du logiciel

TP1 – Tests unitaires

Groupe 1

Équipe 7

Hugo Lachieze-Rey (1934177)

Dimitry Kamga (1898357)

Remis à :

Hanane Ikhelef

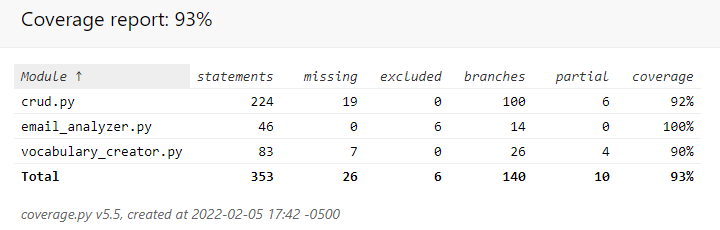
Hiver 2022

1. Tests unitaires
   1. Couverture avant les tests

Table

Description automatically generated with medium confidence

* 1. Couvertures après les tests



* 1. Ajout de tests custom

Nous avons ajouté des tests custom afin d’augmenter notre *coverage* pour chacun des fichiers demandés :

**Email Analyzer**

def test\_spam\_ham\_body\_prob\_Returns\_expected\_probability\_ham(self, mock\_load\_dict):

def test\_subject\_spam\_ham\_prob\_Returns\_expected\_probability\_ham(self, mock\_load\_dict):

**Vocabulary Creator**

def test\_write\_data\_to\_vocab\_file\_returns\_false(self):

def test\_write\_data\_to\_vocab\_file\_return\_true(self):

def test\_clean\_text\_should\_return\_text (self, mock\_clean\_text):

**Crud**

def test\_add\_new\_user\_return\_false\_on\_email\_unicity\_and\_format(  
 self,mock\_read\_groups\_file,mock\_read\_users\_file  
):

def test\_get\_new\_user\_id\_when\_adding\_correct\_user(  
 self, mock\_read\_users\_file):

def test\_get\_new\_group\_id\_when\_adding\_correct\_group(  
 self, mock\_read\_groups\_file):

def test\_add\_new\_group\_fail\_name\_in\_groups\_lookup(  
 self, mock\_read\_groups\_file, mock\_read\_users\_file  
):

def test\_add\_new\_group\_Passes\_correct\_values(  
 self, mock\_get\_user\_id, mock\_read\_groups\_file, mock\_read\_users\_file  
):

def test\_update\_users\_Passes\_correct\_and\_invalid\_is\_Date\_of\_last\_seen\_message\_data\_to\_modify\_users\_file\_when\_field(  
 self, mock\_read\_users\_file, mock\_modify\_users\_file  
):

def test\_update\_users\_Passes\_correct\_and\_invalid\_is\_Date\_of\_first\_seen\_message\_data\_to\_modify\_users\_file\_when\_field(  
 self, mock\_read\_users\_file, mock\_modify\_users\_file  
):

def test\_update\_users\_Passes\_correct\_and\_invalid\_Trust\_data\_to\_modify\_users(  
 self, mock\_read\_users\_file, mock\_modify\_users\_file  
):

def test\_update\_users\_Passes\_correct\_and\_invalid\_SpamN\_data\_to\_modify\_users(  
 self, mock\_read\_users\_file, mock\_modify\_users\_file  
):

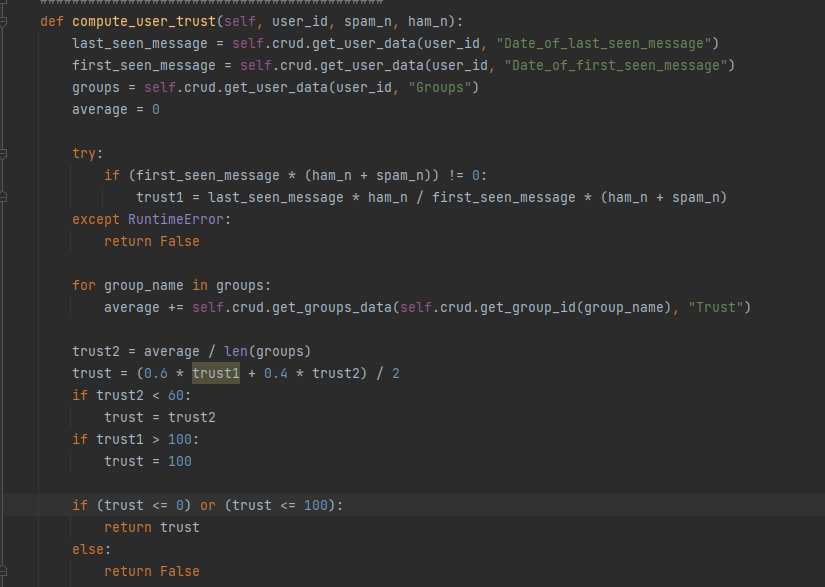
def test\_update\_users\_Passes\_correct\_Groups\_modify\_user\_file(  
 self, mock\_read\_users\_file, mock\_modify\_users\_file, mock\_read\_groups\_file  
):

def test\_update\_users\_return\_False\_on\_invalid\_user\_Groups(  
 self, mock\_read\_users\_file, mock\_modify\_users\_file, mock\_read\_groups\_file  
):

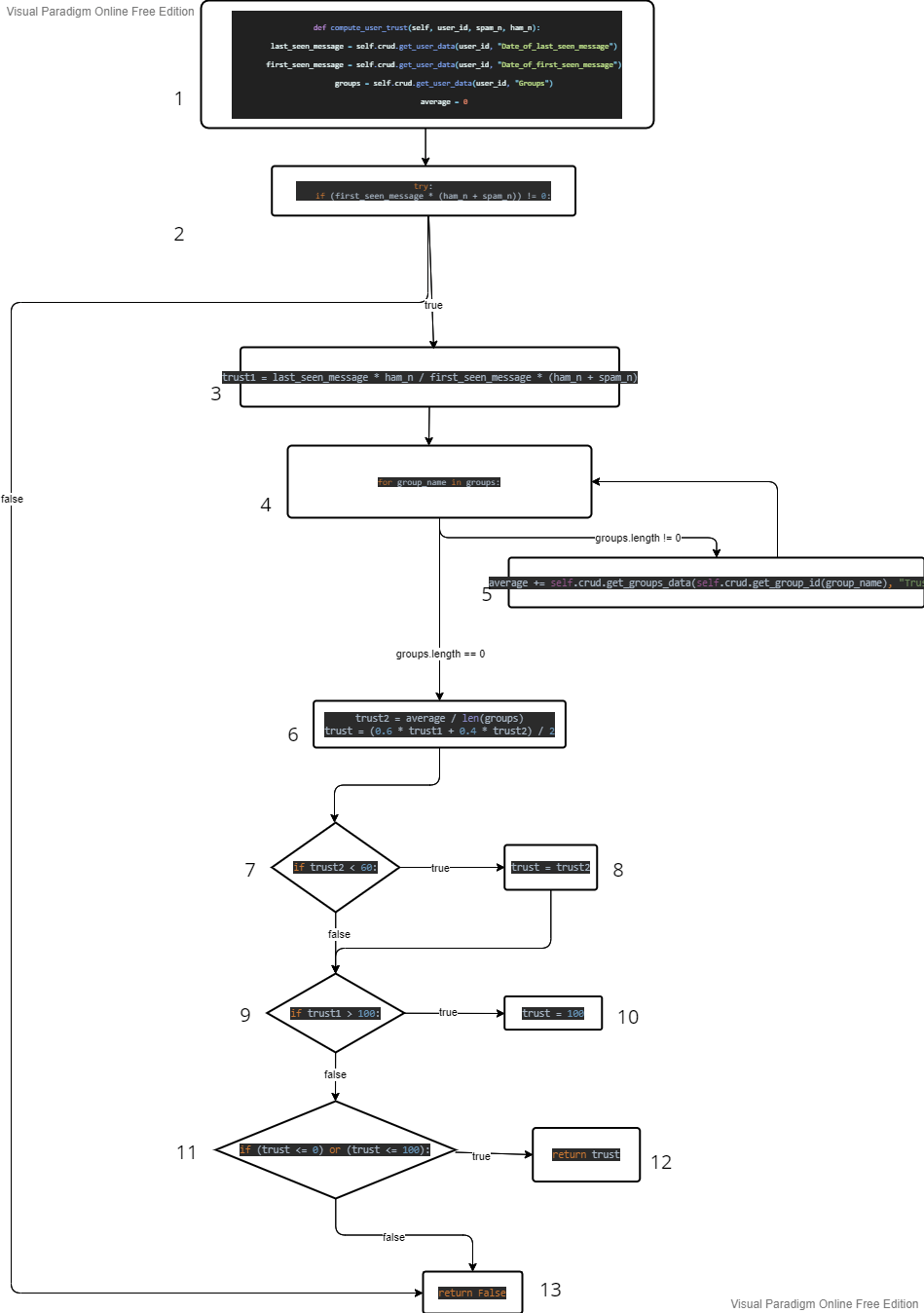
def test\_update\_groups\_Passes\_correct\_and\_invalid\_field\_Trust\_data\_to\_modify\_groups\_file(  
 self, mock\_read\_groups\_file, mock\_modify\_groups\_file  
):

def test\_update\_groups\_Passes\_correct\_on\_invalid\_List\_of\_members(  
 self,  
 mock\_read\_users\_file,  
 mock\_read\_groups\_file,  
 mock\_modify\_groups\_file,  
):

1. Tests de flots de données
   1. Ajout de la fonction compute\_user\_trust



* 1. Graphe CFG



* 1. Tableaux des nœuds

|  |  |  |  |
| --- | --- | --- | --- |
| Nœud | DEF | C-USE | P-USE |
| user\_id | 1 | 1 |  |
| spam\_n | 1 | 2, 3 | 2 |
| ham\_n | 1 | 2, 3 | 2 |
| last\_seen\_message | 1 | 3 |  |
| first\_seen\_message | 1 | 2, 3 |  |
| trust1 | 3 | 6 | 9 |
| average | 1, 5 | 6 |  |
| groups | 1 | 6 | 4 |
| group\_name | 4 | 5 | 4 |
| trust2 | 6 | 6 | 7 |
| trust | 6, 8, 10 | 12 | 11 |

* 1. Cas de tests

**ALL-DEFinition coverage :** au moins un chemin *definition-clear* pour chaque nœud de définition.

DC-PATH (user\_id, 1, 1) = {1}

DC-PATH (spam\_n, 1, 2) = {1, 2}

DC-PATH (ham\_n, 1, 2) = {1, 2}

DC-PATH (last\_seen\_message, 1, 3) = {1, 2, 3}

DC-PATH (first\_seen\_message, 1, 2) = {1, 2}

DC-PATH (first\_seen\_message, 2, 3) = {2, 3}

DC-PATH (trust1, 3, 6) = {3, 4, 6}

DC-PATH (trust1, 3, 6) = {3, 4, 5, 4,6}

DC-PATH (trust1, 6, 9) = {6, 7, 9}

DC-PATH (average, 1, 6) = {1, 2, 3, 4, 6}

DC-PATH (average, 5, 6) = {5, 4, 6}

DC-PATH (groups, 1, 4) = {1, 2, 3, 4}

DC-PATH (group\_name, 4, 5) = {4, 5}

DC-PATH (trust2, 6, 7) = {6, 7}

DC-PATH (trust, 6, 11) = {6, 7,9,11}

**Cas de test C-USE**

|  |  |  |
| --- | --- | --- |
|  | Chemins | Cas de test |
| A | {1,2,3,4,5,4,6,7,9,11,13} |  |
| B | {1,2,3,4,5,4,6,7,8,9,11,13} |  |
| C | {1,2,3,4,5,4,6,7,9,10,11,13} |  |
| D | {1,2,3,4,5,4,6,7, 9,11,12} |  |
| E | {} |  |

**Cas de test P-USE**

|  |  |  |
| --- | --- | --- |
|  | Chemins | Cas de test |
| A | {1,2,3,4,5,4,6,7,9,11,1} |  |
| B | {1,2,3,4,5,4,6,7,8,9,11,13} |  |
| C | {1,2,3,4,5,4,6,7,9,10,11,13} |  |
| D | {1,2,3,4,5,4,6,7, 9,11,12} |  |
| E | {} |  |

**Cas de test All USE**

|  |  |  |
| --- | --- | --- |
|  | Chemins | Cas de test |
| A | {1,2,3,4,5,4,6,7,9,11,13} |  |
| B | {1,2,3,4,5,4,6,7,8,9,11,13} |  |
| C | {1,2,3,4,5,4,6,7,9,10,11,13} |  |
| D | {1,2,3,4,5,4,6,7, 9,11,12} |  |
| E | {} |  |