



LOG3430

Méthodes de test et de validation du logiciel

Automne 2021

TP No. 1

Groupe [1]

[2013658] – [Falicoff]

[2014536] - [Rouleau]

[1995467]-[Labrie]

Soumis à :

Cyrine Zid

Vendredi 1er Octobre 2021

Partie 1

Avant l'écriture des tests:

Name	Stmts	Miss	Branch	BrPart	Cover
crud.py	224	200	100	0	7%
email_analyzer.py	46	38	14	0	13%
main.py	43	43	12	0	0%
renee.py	99	99	40	0	0%
test_crud.py	113	0	0	0	100%
test_email_analyzer.py	35	0	0	0	100%
test_vocabulary_creator.py	18	0	0	0	100%
text_cleaner.py	29	18	6	0	31%
vocabulary_creator.py	83	73	26	0	9%
TOTAL	690	471	198	0	25%

test_crud.py (Tests ajoutés):

Avant l'ajout :

Name	Stmts	Miss	Branch	BrPart	Cover
crud.py	224	64	100	15	68%

Après l'ajout :

Name	Stmts	Miss	Branch	BrPart	Cover
crud.py	224	41	100	17	80%

Les tests ont été ajoutés dans les fonction `update_users` et `update_groups` de sorte de faire la couverture des branches (toutes les possibilités des différents paramètres qu'il est possible de modifier).

1- `test_update_users_Passes_correct_data_to_modify_users_file_Date_of_last_seen_message`

2- `test_update_users_Passes_correct_data_to_modify_users_file_Date_of_first_seen_message`

3- `test_update_users_Passes_correct_data_to_modify_users_Trust`

- 4- `test_update_users_Passes_correct_data_to_modify_users_SpamN_HamN`
- 5- `test_update_users_return_false_on_wrong_Groups`
- 6- `test_update_users_return_false_on_wrong_field`
- 7- `test_update_groups_Passes_correct_data_to_modify_groups_file_Trust`
- 8- `test_update_groups_return_false_on_wrong_List_of_members`
- 9- `test_update_groups_return_false_on_wrong_field`

A la fin, on obtient une couverture finale de:

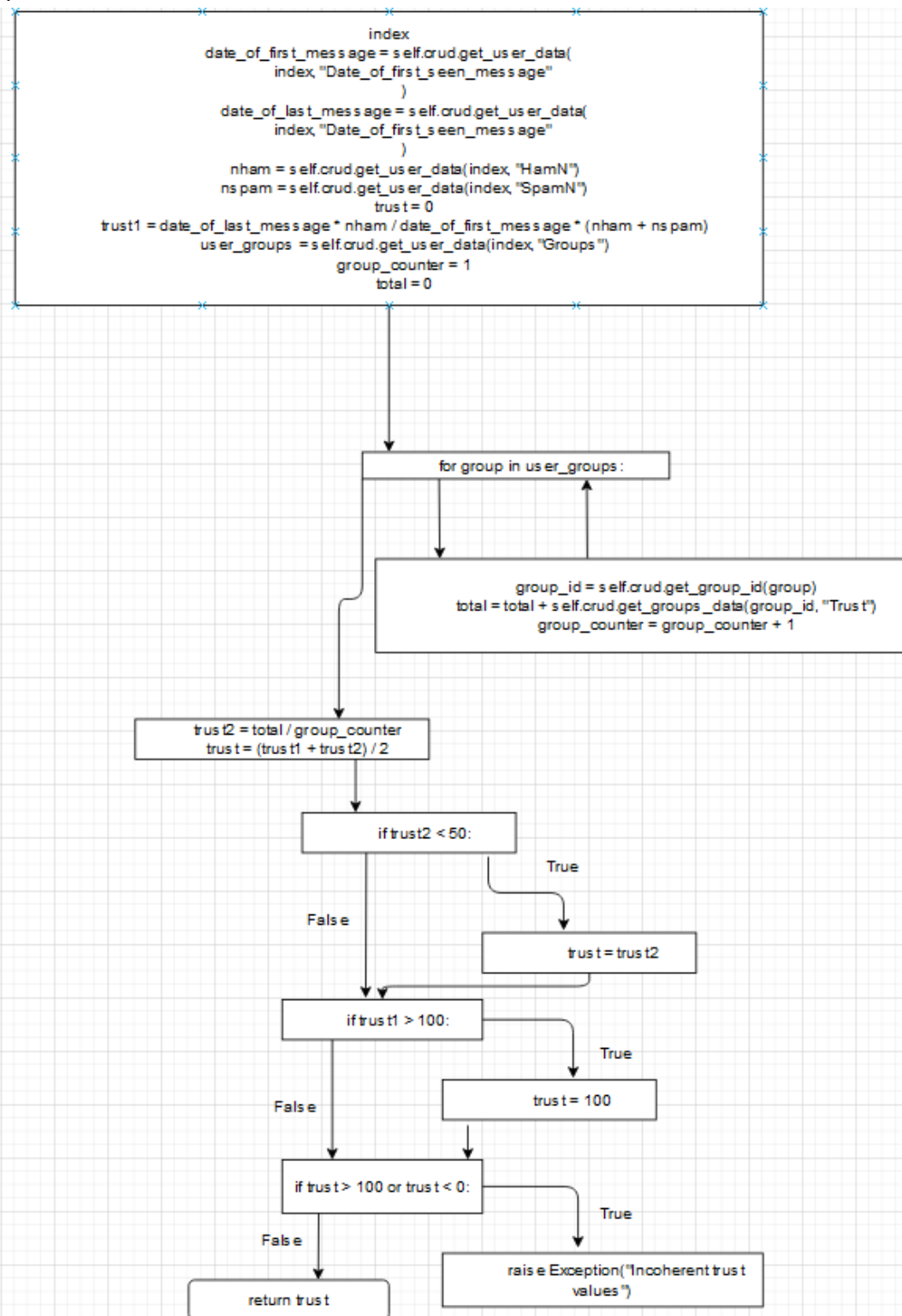
Name	Stmts	Miss	Branch	BrPart	Cover
crud.py	224	41	100	17	80%
email_analyzer.py	46	4	14	4	87%
vocabulary_creator.py	80	12	26	1	88%
TOTAL	350	57	140	22	82%

Partie 2

Implémentation de la fonction calculate Trust:

```
178     # index est l'index de l'utilisateur
179     def calculate_trust(self, index):
180         date_of_first_message = self.crud.get_user_data(index, "Date_of_first_seen_message")
181         date_of_last_message = self.crud.get_user_data(index, "Date_of_last_seen_message")
182         nham = self.crud.get_user_data(index, "HamN")
183         nspam = self.crud.get_user_data(index, "SpamN")
184         trust = 0
185         trust1 = date_of_last_message * nham / date_of_first_message * (nham + nspam)
186         user_groups = self.crud.get_user_data(index, "Groups")
187         group_counter = 1
188         total = 0
189
190         for group in user_groups:
191             group_id = self.crud.get_group_id(group)
192             total = total + self.crud.get_groups_data(group_id, "Trust")
193             group_counter = group_counter + 1
194
195         trust2 = total / group_counter
196         trust = (trust1 + trust2) / 2
197
198         if trust2 < 50:
199             trust = trust2
200         if trust1 > 100:
201             trust = 100
202         if trust > 100 or trust < 0:
203             raise Exception("Incoherent trust values")
204
205         return trust
206
```

Grphe CFG:



Variable	Def	C-Use	P-Use
index	179	180, 181, 182, 183, 186	
Chemins pour variable	Defs: {179} C-Use: {180, 181, 182, 183, 184, 185, 186} P-Use: {} All-Uses: {179}{180, 181, 182, 183, 184, 185, 186}		

Variable	Def	C-Use	P-Use
date_of_first_message	180	185	
Chemins pour variable	All Defs: {180} All C-Use: {185} All P-Use: {} All Use: {180, 185}		

Variable	Def	C-Use	P-Use
date_of_last_message	180	185	
Chemins pour variable	All Defs: {180} All C-Use: {185} All P-Use: {} All Use: {180, 185}		

Variable	Def	C-Use	P-Use
Nham	182	185	
Chemins pour variable	All Defs: {182} All C-Use: {185} All P-Use: {} All Use: {182, 185}		

Variable	Def	C-Use	P-Use
nspam	183	185	
Chemins pour variable	All Defs: {183, 185} All C-Use: {183, 185} All P-Use: {} All Use: {183, 185}		

Variable	Def	C-Use	P-Use
Trust	184, 196, 199, 201	205	202
Chemins pour variable	All Defs: {184, 196, 198, 200, 205}, {199, 200, 202, 205}, {201, 202, 205} All C-Use: {184, 196, 198, 200, 205}, {199, 200, 205}, {201, 202, 205} All P-Use: {184, 196, 198, 200}, {199, 200, 202}, {201, 202} All Use: {184, 196, 198, 200, 205}, {199, 200, 202, 205}, {201, 202, 205}		

Variable	Def	C-Use	P-Use
Trust1	185	196	200
Chemins pour variable	All Defs: {185, 186, 187, 188, 190, 191, 192, 193, 190, 195, 196} All C-Use: {185, 186, 187, 188, 190, 191, 192, 193, 190, 195, 196} All P-Use: {185, 186, 187, 188, 190, 191, 192, 193, 190, 195, 196 200} All Use: {185, 186, 187, 188, 190, 191, 192, 193, 190, 195, 196 200}		

Variable	Def	C-Use	P-Use
User_groups	186		190
Chemins pour variable	All Defs: {186, 187, 188, 190} All C-Use: {} All P-Use: {186, 187, 188, 190} All Use: {186, 187, 188, 190}		

Variable	Def	C-Use	P-Use
group_counter	187, 193	193, 195	
Chemins pour variable	All Defs: {187, 193} All C-Use: {187, 193, 195} {187, 195} All P-Use: {} All Use: {187, 193, 195}		

Variable	Def	C-Use	P-Use
Total	188, 192	195	
Chemins pour variable	All Defs: {188, 192} All C-Use: {188, 192, 195} {188, 195} All P-Use: {} All Use: {188, 192, 195}		

Variable	Def	C-Use	P-Use
Group	190	191	
Chemins pour variable	All Defs: {190, 191} All C-Use: {190, 191} All P-Use: {} All Use: {190, 191}		

Chemins minimaux

Tous le chemins minimaux sont les mêmes pour tous les critères (les trois chemins sont les mêmes pour les critères utilisés ci dessus)

Trust2 > 50 ^ trust1 < 100

{179,180,181,182,183,184,185,186,187,188,190.191,192,193,190,195,196,198,200,202,205}

Trust1 > 100 ^ trust2 > 50

{179,180,181,182,183,184,185,186,187,188,190.191,192,193,190,195,196,198,200,201,202,205}

Trust 1 < 100 ^ Trust < 50

{179,180,181,182,183,184,185,186,187,188,190.191,192,193,190,195,196,198,199,200,202,205}

Jeux de tests

On cherche un jeux de test qui satisfait

	user1	user2	user3
Trust 1 > 100	f	f	v
Trust 2 < 50	v	f	f

Pour user1

On cherche pour que user1 aye trust a 0% ainsi que ses groupes

Trust1: $1 * 0$

Trust2: 0 car ses groupes on 0% Trust

Pour user2

On cherche pour que user2 aye trust a entre 50 et 75% ainsi que ses groupes

Trust1: $1 * 60\%$

Trust2: 60 car ses groupes on 70 et 50% Trust

Pour user3

On cherche pour que user3 aye trust a 100% ainsi que ses groupes

Trust1: $100 * 1$

Trust2: 100 car ses groupes on 100% Trust

On peut donc définir notre input:

```
{
  "1": {
    "name": "user1@doom.com",
    "Trust": 0.0,
    "Spamn": 2,
    "HamN": 0,
    "Date_of_first_seen_message": 968976000.0,
    "Date_of_last_seen_message": 968976000.0,
    "Groups": [
      "badgroup1",
      "badgroup2"
    ]
  },
  "2": {
    "name": "user2@mail.com",
    "Trust": 0.60,
```

```
"SpamN": 2,  
"HamN": 3,  
"Date_of_first_seen_message": 968976000.0,  
"Date_of_last_seen_message": 968976000.0,  
"Groups": [  
  "normalgroup1",  
  "normalgroup2"  
]  
}
```

```
"3": {  
  "name": "user3@mail.com",  
  "Trust": 1.0,  
  "SpamN": 0,  
  "HamN": 2  
  "Date_of_first_seen_message": 1814567890.0,  
  "Date_of_last_seen_message": 1014567890.0,  
  "Groups": [  
    "goodgroup1",  
    "goodgroup2"  
  ]  
}
```

Avec

```
"badgroup1",  
"Badgroup2" ayant un trust de 0
```

```
"normalgroup1",  
"Normalgroup2" ayant un trust de 70 et 50 respectivement
```

```
"goodgroup1",  
"Goodgroup2" ayant un trust de 100
```

On s'attend à ce que le test retourne:

User1: 0

User2: 60

User3: 100

Critère le plus strict

- Le critère le plus strict est le All-Use puisqu' il doit passer par chaque Def et Use.
 - All Defs est moins strict car il faut au moins un chemin DC pour chaque Def
 - C-use est moins strict car il ne passe pas par les P-uses
 - P-use est moins strict car il ne passe pas par les C-uses