

Changes in our UML:

-We add the class UserAccount.

-We add in the class System attributes : VatPercentage , CreditCards ,country code , spamWords ,welcomePageMsg.

-We change the attribute father in category to optional and the navegability.

-We add a relationship beetwen Note and Report.

-We change our navegability in some clases because bidireccional relationships didn't work. These relationships were between Aplication and Task and Task and Customer.

QUERIES:

C QUERIES:

-Query C/1:

Especification:

The average, the minimum, the maximum, and the standard deviation of the number of fix-up tasks per user.

Code:

```
select avg(c.tasks.size), min(c.tasks.size), max(c.tasks.size), stddev(c.tasks.size) from Customer c;
```

Result:

```
1 object selected  
[1.0, 0, 2, 1.0]
```

-Query C/2:

Especification:

The average, the minimum, the maximum, and the standard deviation of the number of applications per fix-up task.

Code:

```
select avg(t.applications.size), min(t.applications.size), max(t.applications.size),  
stddev(t.applications.size) from Task t;
```

Result:

1 object selected
[1.0, 1, 1, 0.0]

-Query C/3:

Especification:

The average, the minimum, the maximum, and the standard deviation of the maximum price of the fix-up tasks.

Code:

```
select avg(t.maximumPrice), max(t.maximumPrice), min(t.maximumPrice),  
stddev(t.maximumPrice) from Task t;
```

Result:

1 object selected
[1400.8, 1400.8, 1400.8, 0.0]

-Query C/4:

Especification:

The average, the minimum, the maximum, and the standard deviation of the price offered in the applications.

Code:

```
select avg(a.maximumPrice), max(a.maximumPrice), min(a.maximumPrice),  
stddev(a.maximumPrice) from Application a;
```

Result:

1 object selected
[200.0, 300.0, 100.0, 81.64965809277261]

-Query C/5:

Especification:

The ratio of pending applications.

Code:

```
select count(ap1)*1./(select count(ap2) from Application ap2) from Application ap1 where  
ap1.status = 'PENDING';
```

Result:

1 object selected
0.3333

-Query C/6:

Especification:
The ratio of accepted applications.

Code:

```
select count(ap1)*1./(select count(ap2) from Application ap2) from Application ap1 where  
ap1.status = 'ACCEPTED';
```

Result:

1 object selected
0.3333

-Query C/7:

Especification:
The ratio of rejected applications.

Code:

```
select count(ap1)*1./(select count(ap2) from Application ap2) from Application ap1 where  
ap1.status = 'REJECTED';
```

Result:

1 object selected
0.3333

-Query C/8:

Especification:
The ratio of pending applications that cannot change its status because their time period's elapsed.

Code:

```
select count(ap)*1. from Task t join t.applications ap where ap.status = 'REJECTED' and  
t.endDate < ap.datetime;
```

Result:

1 object selected
1.0

-Query C/9:

Especificación:

The listing of customers who have published at least 10% more fix-up tasks than the average, ordered by number of applications.

Code:

```
select c from Customer c where (c.tasks.size)>= 1.1*(select avg(c.tasks.size) from Customer c);
```

Result:

1 object selected

domain.Customer{id=249, version=0}

domain.DomainEntity::id: int = 249

domain.DomainEntity::version: int = 0

domain.Actor::name: java.lang.String = "Marta"

domain.Actor::middleName: java.lang.String = "Marta"

domain.Actor::surname: java.lang.String = "Dorante"

domain.Actor::photo: java.lang.String =
"http://as00.epimg.net/img/comunes/fotos/fichas/equipos/large/171.png"

domain.Actor::email: java.lang.String = "anmarta@gmail.com"

domain.Actor::phoneNumber: java.lang.String = "+34638677432"

domain.Actor::address: java.lang.String = "Calle Betis"

domain.Actor::socialProfiles: java.util.Collection = [domain.SocialProfile{id=263,
version=0}]

domain.Actor::boxes: java.util.Collection = [domain.Box{id=256, version=0}]

domain.Actor::messagesReceived: java.util.Collection = []

domain.Actor::messagesSended: java.util.Collection = []

domain.Actor::userAccount: security.UserAccount = security.UserAccount{id=189,
version=0}

domain.Customer::complaints: java.util.Collection = []

domain.Customer::notes: java.util.Collection = []

domain.Customer::endorseReceiver: java.util.Collection = []

domain.Customer::endorseSender: java.util.Collection = []

domain.Customer::tasks: java.util.Collection = [domain.Task{id=246, version=0},
domain.Task{id=247, version=0}]

-Query C/10:

Especification:

The listing of handy workers who have got accepted at least 10% more applications than the average, ordered by number of applications.

Code:

```
select h from HandyWorker h join h.application as a where h.application.size > (select  
avg(h.application.size)*1.1 from HandyWorker h) and a.status='ACCEPTED' order by  
h.application.size;
```

Result:

0 objects selected

B QUERIES:

-Query B/1:

Especification:

The minimum, the maximum, the average, and the standard deviation of the number of complaints per fix-up task.

Code:

```
select avg(t.complaints.size), min(t.complaints.size), max(t.complaints.size),  
stddev(t.complaints.size) from Task t;
```

Result:

1 object selected
[0.5, 0, 1, 0.5]

-Query B/2:

Especification:

The minimum, the maximum, the average, and the standard deviation of the number of notes per referee report.

Code:

```
select avg(r.notes.size), min(r.notes.size), max(r.notes.size), stddev(r.notes.size) from Report r;
```

Result:

1 object selected
[1.0, 1, 1, 0.0]

-Query B/3:

Especification:

The ratio of fix-up tasks with a complaint.

Code:

```
select count(t1)*1./(select count(t2) from Task t2) from Task t1 where t1.complaints.size >= 1;
```

Result:

1 object selected
0.5

-Query B/4:

Especification:

The top-three customers in terms of complaints.

Code:

```
select c from Customer c order by c.complaints.size;
```

Result:

2 objects selected

domain.Customer{id=248, version=0}

domain.DomainEntity::id: int = 248

domain.DomainEntity::version: int = 0

domain.Actor::name: java.lang.String = "Marta"

domain.Actor::middeName: java.lang.String = "Marta"

domain.Actor::surname: java.lang.String = "Dorante"

domain.Actor::photo: java.lang.String =
"http://as00.epimg.net/img/comunes/fotos/fichas/equipos/large/171.png"

domain.Actor::email: java.lang.String = "andrea@gmail.com"

domain.Actor::phoneNumber: java.lang.String = "+34638677432"

domain.Actor::address: java.lang.String = "Calle Betis"

domain.Actor::socialProfiles: java.util.Collection = []

domain.Actor::boxes: java.util.Collection = [domain.Box{id=245, version=0}]

domain.Actor::messagesReceived: java.util.Collection = []

domain.Actor::messagesSended: java.util.Collection = []

domain.Actor::userAccount: security.UserAccount = security.UserAccount{id=189,
version=0}

domain.Customer::complaints: java.util.Collection = []

domain.Customer::notes: java.util.Collection = []

domain.Customer::endorseReceiver: java.util.Collection = [domain.Endorse{id=251,
version=0}, domain.Endorse{id=250, ver
sion=0}]

domain.Customer::endorseSender: java.util.Collection = [domain.Endorse{id=251,
version=0}, domain.Endorse{id=250, versi
on=0}]

domain.Customer::tasks: java.util.Collection = []

domain.Customer{id=249, version=0}

domain.DomainEntity::id: int = 249

```

domain.DomainEntity::version: int = 0

domain.Actor::name: java.lang.String = "Marta"

domain.Actor::middeName: java.lang.String = "Marta"

domain.Actor::surname: java.lang.String = "Dorante"

domain.Actor::photo: java.lang.String =
"http://as00.epimg.net/img/comunes/fotos/fichas/equipos/large/171.png"

domain.Actor::email: java.lang.String = "anmarta@gmail.com"

domain.Actor::phoneNumber: java.lang.String = "+34638677432"

domain.Actor::address: java.lang.String = "Calle Betis"

domain.Actor::socialProfiles: java.util.Collection = [domain.SocialProfile{id=263,
version=0}]

domain.Actor::boxes: java.util.Collection = [domain.Box{id=256, version=0}]

domain.Actor::messagesReceived: java.util.Collection = []

domain.Actor::messagesSended: java.util.Collection = []

domain.Actor::userAccount: security.UserAccount = security.UserAccount{id=189,
version=0}

domain.Customer::complaints: java.util.Collection = []

domain.Customer::notes: java.util.Collection = []

domain.Customer::endorseReceiver: java.util.Collection = []

domain.Customer::endorseSender: java.util.Collection = []

domain.Customer::tasks: java.util.Collection = [domain.Task{id=246, version=0},
domain.Task{id=247, version=0}]

```

-Query B/5:

Especificación:

The top-three handy workers in terms of complaints.

Code:

```
select h from HandyWorker h join h.finder f join f.task t order by t.complaints.size;
```

Result:

2 objects selected

domain.HandyWorker{id=242, version=0}

domain.DomainEntity::id: int = 242

domain.DomainEntity::version: int = 0

domain.Actor::name: java.lang.String = "Maria"

domain.Actor::middeName: java.lang.String = "Maria"

domain.Actor::surname: java.lang.String = "Fernandez"

domain.Actor::photo: java.lang.String = "http://maria.png"

domain.Actor::email: java.lang.String = "maria@gmail.com"

domain.Actor::phoneNumber: java.lang.String = "+34638097432"

domain.Actor::address: java.lang.String = "Av Reina Mercedes"

domain.Actor::socialProfiles: java.util.Collection = [domain.SocialProfile{id=244, version=0}]

domain.Actor::boxes: java.util.Collection = [domain.Box{id=252, version=0}]

domain.Actor::messagesReceived: java.util.Collection = []

domain.Actor::messagesSended: java.util.Collection = []

domain.Actor::userAccount: security.UserAccount = security.UserAccount{id=188, version=0}

domain.HandyWorker::make: java.lang.String = "Task 2"

domain.HandyWorker::finder: domain.Finder = domain.Finder{id=241, version=0}

domain.HandyWorker::application: java.util.Collection = [domain.Application{id=237, version=0}]

domain.HandyWorker::notes: java.util.Collection = [domain.Note{id=199, version=0}]

domain.HandyWorker::endorseReceiver: java.util.Collection = [domain.Endorse{id=251, version=0}]

domain.HandyWorker::endorseSender: java.util.Collection = [domain.Endorse{id=250, version=0}]

domain.HandyWorker{id=243, version=0}

domain.DomainEntity::id: int = 243

```
domain.DomainEntity::version: int = 0

domain.Actor::name: java.lang.String = "Julian"

domain.Actor::middeName: java.lang.String = "Ostras"

domain.Actor::surname: java.lang.String = "LOL"

domain.Actor::photo: java.lang.String =
"http://as00.epimg.net/img/comunes/fotos/fichas/equipos/large/171.png"

domain.Actor::email: java.lang.String = "julian@gmail.com"

domain.Actor::phoneNumber: java.lang.String = "+34638677432"

domain.Actor::address: java.lang.String = "Calle Betis"

domain.Actor::socialProfiles: java.util.Collection = []

domain.Actor::boxes: java.util.Collection = [domain.Box{id=255, version=0}]

domain.Actor::messagesReceived: java.util.Collection = []

domain.Actor::messagesSended: java.util.Collection = []

domain.Actor::userAccount: security.UserAccount = security.UserAccount{id=191,
version=0}

domain.HandyWorker::make: java.lang.String = "Task 3"

domain.HandyWorker::finder: domain.Finder = domain.Finder{id=241, version=0}

domain.HandyWorker::application: java.util.Collection = [domain.Application{id=238,
version=0}]

domain.HandyWorker::notes: java.util.Collection = [domain.Note{id=202, version=0}]

domain.HandyWorker::endorseReceiver: java.util.Collection =
[domain.Endorse{id=250, version=0}]

domain.HandyWorker::endorseSender: java.util.Collection = [domain.Endorse{id=251,
version=0}]
```