

Tester le comportament

Fabio RAMPOLDI
iLM

Le Projet I

Gestion RH pour laboratoire multi tutelles

- Application Web (Framework Django)
- Modulaire (Gestion salles, Switch, H&S, Publi...)
- Cahier des charges

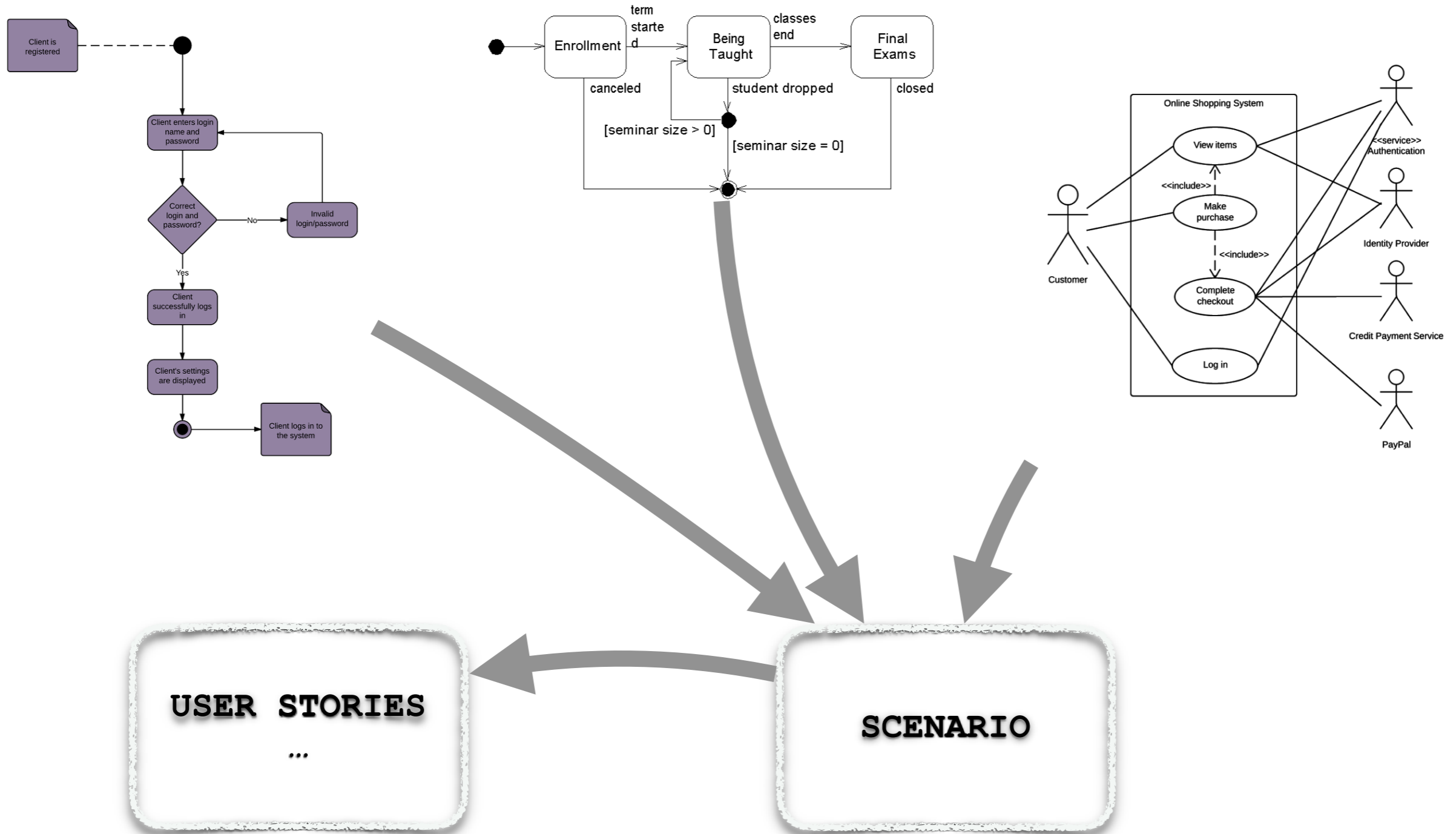
Le Projet II

Être Agile

eXtreme Programming

- privilégier la communication orale directe par rapport à l'échange de documents ;
- **les tests doivent être créés avant chaque écriture du code ;**
- penser en modules et sous-modules ;
- l'intégration, pour éviter une régression;
- **programmation en binôme;**
- définir des règles de codage suivies par l'ensemble de l'équipe ;
- conception et refactoring du code en continue.

RECUEIL DES BESOINS



Scenario I

Déterminer comment le client interagit avec le produit.

Un scénario est un chemin particulier de la description abstraite fournie par un cas d'utilisation.

Scenario II

<i>Nom</i>	Déclarer PARTI un utilisateur
<i>Description</i>	L'utilisateur quitte l'institut
<i>Acteur Principale</i>	Gestionnaire
<i>Acteurs Secondaires</i>	Utilisateur
<i>Préconditions</i>	<ol style="list-style-type: none">1. Le gestionnaire doit avoir accès et être présent dans la base de données
<i>Postconditions</i>	<ol style="list-style-type: none">1. Archiver la fiche de l'utilisateur2. Supprimer les réservations des bureaux de l'utilisateur3. Notifier l'utilisateur à rendre la clé du bureau à la secrétaire
<i>Déroulement normal</i>	<ol style="list-style-type: none">1. Le gestionnaire accède au profil de l'utilisateur2. Le gestionnaire change l'état de l'utilisateur à PARTI3. Le gestionnaire valide le changement et enregistre les modifications
<i>Contraintes non fonctionnelles</i>	—

User Stories

compléter le scénario avec des user stories ou de courtes phrases formatées qui visent à décrire un besoin (validé par le client).

- Étant donné un utilisateur quand quitte l'institut alors son état devient PARTI.

TESTS D'ACCEPTANCE

Ces sont des tests utilisés pour vérifier si le système respecte le besoin du client.

Depuis des *users stories* on peut définir des tests d'acceptations.

Ensuite avec des bibliothèques comme *concordion* ou *behave*, on peut l'écrire pour l'exécuter.

TESTS D'ACCEPTANCE

Le client doit comprendre les tests

Syntaxe Gherkin

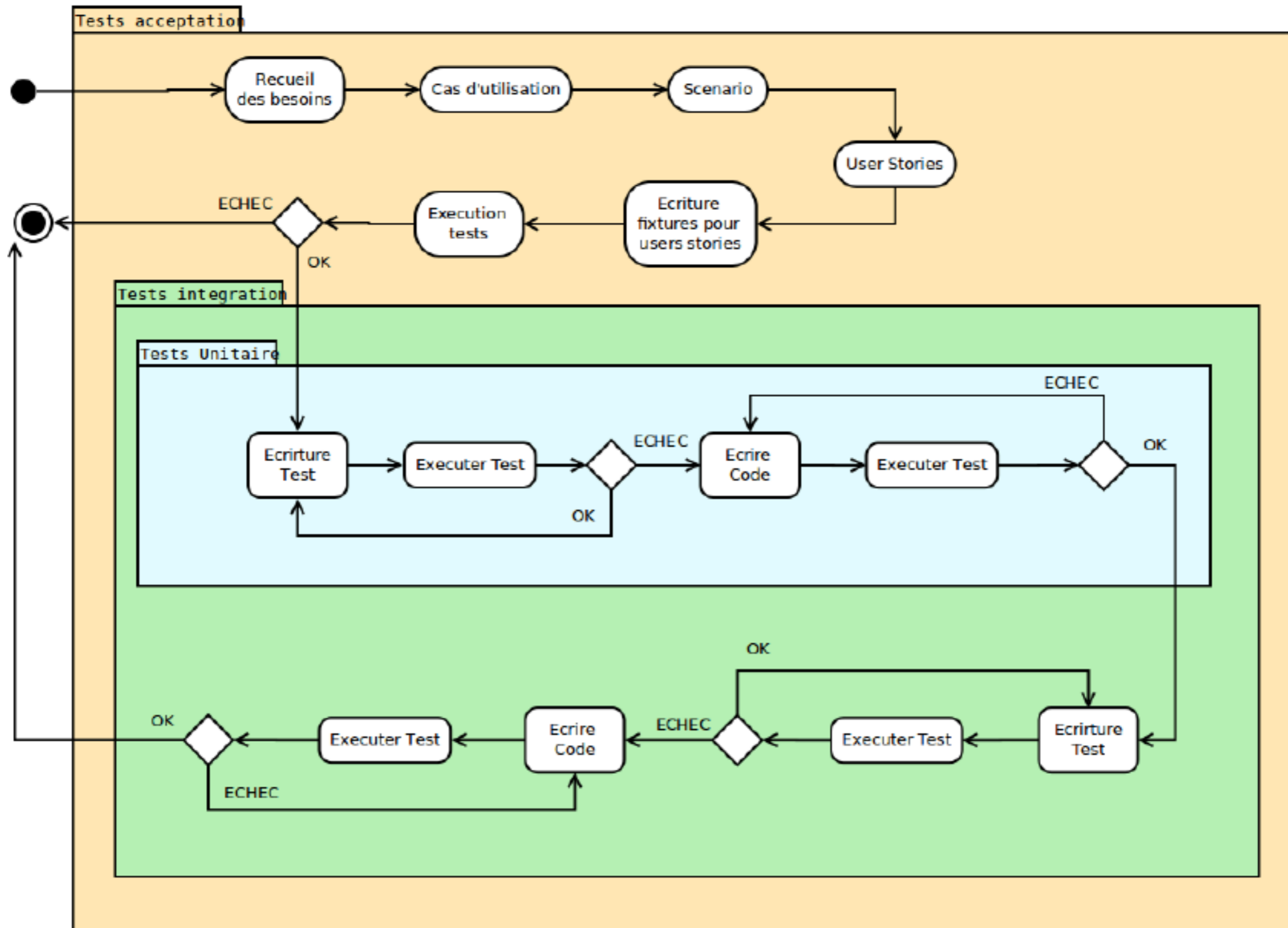
- **Given** [*Étant Donné*]
pour définir un état initial ;
- **When** [*Quand*]
pour annoncer une action ;
- **Then** [*Alors*]
il y a un résultat ;
- **I Want** [*Je Veux*]
désirer quelque chose ;
- **As A** [*En Tant Que*]
pour définir le rôle ;
- **In Order To** [*An De*]
but de la fonctionnalité.

Ecrire les Tests

« En tant qu'utilisateur authentifié je veux créer une fiche nouvel entrant afin de l'intégrer dans mon équipe in fine dans l'institut »

- **Étant donné** que je suis l'utilisateur authentifié **et que je** valide le formulaire avec des données non valides **alors** je reste sur le formulaire afin de le modifier.

Plan de Tests



BDD AVEC DJANGO

```
pip install behave-django
```

```
mkdir -p features/steps
```

```
touch features/administration.feature
```

```
touch features/steps/administration.py
```

```
python manage.py behave
```

Example I

Feature: Administration

Scenario Outline: Admin logs in

Given User go to admin page
When I log in as <username>
Then Show <page>

Example II

```
@given(u'User go to admin page')
def given_admin_login_page(context):
    context.browser.visit('/admin/login')
```

```
@when(u'I log in as "Admin"')
def when_login(context):
    username_field = context.browser.find_by_id('id_username')
    password_field = context.browser.find_by_id('id_password')
    username_field.send_keys('Admin')
    password_field.send_keys('12345')
    submit_button = context.browser.find_by_class('submit-button')
    submit_button.click()
```

```
@then(u'Show {urlforadmin}')
```

```
def then_show_page(context, urlforadmin):
    context.browser.visit(urlforadmin)
```


ET PRO CON

- Historique des besoins
- Documentation itérative
- Simple pour le client
- Ecrire des tests aussi pour des besoins simple
- Modifier les scenarios sans l'accord du client
- Implication du client dans l'écriture des scenario

Merçi