TD Gestion de projet - 01

1. Analyse et conception du projet

Introduction

Une entreprise souhaite mettre en place une application Web pour pouvoir gérer la réservation de ses salles de réunions, l'ancienne méthode étant de réserver les salles en renseignant un encart papier sur la porte.

Les collaborateurs font face aux problématiques suivantes:

- Difficulté à trouver de salle disponible
- Des salles réservées régulièrement sont parfois vides
- Les collaborateurs en déplacement ne peuvent pas libérer les salles réservées sans passer par des personnes présentes sur place.

Objectif du projet

L'objectif du projet est de développer une application Web qui permettra de gérer efficacement la réservation des salles de réunions, elle devra répondre aux besoins suivants:

- Faciliter la recherche de salles disponibles
- Optimiser l'utilisation des salles
- Permettre la gestion à distance

Les utilisateurs de l'application seront les collaborateurs ainsi que les office manager.
Les collaborateurs pourront lister les salles, voir leur statut de réservation, faire une réservation et les annuler.
Les salles seront réservées suivant des périodes précises.

Les Office Managers pourront effectuer les mêmes actions que les collaborateurs, ils pourront également ajouter ou supprimer des salles, temporairement empêcher la réservation de celles-ci ou annuler des réservations existantes.

Chaque salle a une capacité maximale qui lui est assignée à la création. Après blocage d'une salle, les Office Managers peuvent également changer la capacité de la salle.

Contraintes

L'application s'intégrera dans le SI actuel de l'entreprise. Il sera donc déployé par l'équipe infrastructure interne de l'entreprise. De plus, la maintenance sera assurée par les équipes

de développement de l'entreprise, par conséquent il faudra que l'application puisse être déployée selon les standards de l'entreprise.

Ainsi, l'application devra être développée en Python, utilisant le Framework Django, pouvant être déployée dans un container Docker et que l'on puisse configurer plusieurs instances pour chaque immeuble. De plus, chaque fonctionnalité devra être couverte par un test unitaire.

Livrable

Les livrables attendus pour ce projet sont:

- Un cahier des charges.
- Un planning prévisionnel estimant le nombre d'itérations nécessaires à développer cette application.
- Un repository git avec le code source de l'application.
- Un Dockerfile permettant de containeriser l'application.
- Un fichier d'intégration continue Gitlab CI permettant d'exécuter les tests du projet et de créer et déployer le container applicatif.

Contact

Pour toute question ou clarification, veuillez contacter les responsables de TD.

2. Planification, initialisation et structure du projet

Avant Projet et rédaction du cahier des charges

1. En groupes de 2 ou 3 préparer la rédaction du cahier des charges.
2. Lire la présentation du projet.
3. Lister les uses cases de l'application afin d'en agrémenter le cahier des charges.
4. Quels sont les risques liés à cette application ? En agrémenter également le cahier des charges. Ceux-ci seront regroupés dans 3 catégories:
    a. Risques liés à la conception et au développement
    b. Risques de planification
    c. Risques liés à la maintenance
5. Modéliser la base de données et les entités associées.
6. Écrire les user stories SCRUM et chiffrer la complexité des user stories.

7. On estimera que l'on peut effectuer 20 points par sprint et qu'un sprint coûte environ 10000€ produire une estimation de la durée du projet
8. Identifier les dépendances entre les Story et construire un diagramme de dépendances entre les stories.
9. Faire un diagramme de Gantt pour avoir une idée de ce qui peut être parallélisé ainsi que de pouvoir estimer comment la charge sera répartie entre les sprints

Initialisation du projet

Objectifs

1. Initialiser un projet Django avec Poetry.
2. Configurer les dépendances et les tests.
3. Initialiser le projet Django.
4. Configurer les tests pour Django.
5. Créer un super utilisateur pour Django.
6. Démarrer le serveur Django.
7. Initialiser un dépôt Git pour le projet.

Étapes d'initialisation du projet

1. Installer poetry

```
pip install poetry
```

2. Initialiser un projet à l'aide de poetry.

```
poetry new reservation-salle
```

3. Ajouter les dépendances pour créer un projet Django ainsi que pour les tests

```
poetry add django
poetry add --group dev pytest pytest-django pytest-html
```

4. Supprimer le dossier reservation_salle créé par Poetry et lancer l'initialisation du projet par Django:

```
rm -rf reservation_salle
poetry run django-admin startproject reservation_salle
```

5. Initialiser les applications suivies par django en ajoutant la ligne suivantes dans le fichier settings.py dans le tableau INSTALLED_APPS:

```
'reservation_salle'
```

6. Appliquer les migrations initiales (sur la base de données):

```
poetry run reservation_salle/manage.py migrate
```

7. Configurer les framework de tests pour qu'il utilise Django en ajoutant la configuration suivante dans le fichier pyproject.toml:

```
[tool.pytest.ini_options]
DJANGO_SETTINGS_MODULE = "reservation_salle.settings"
python_files = ["test_*.py", "_test.py", "testing/.python/*.py"]
pythonpath = [".", "reservation_salle"]
```

8. Exécuter les tests pour vérifier que tout fonctionne:

```
2. poetry run pytest
```

9. Créer un super utilisateur pour Django:

```
poetry run reservation_salle/manage.py createsuperuser
```

10. Démarrer le serveur Django:

```
poetry run reservation_salle/manage.py runserver
```

Vous pouvez vérifier que le serveur est correctement démarré en vous rendant à l'adresse http://localhost:8000

Initialisation du dépôt Git

1. Initialiser le dépôt Git:

```
git init
```

2. Créer le fichier .gitignore et ajouter les fichiers qui pourraient être pertinents à l'utilisation:

```
.idea
db.sqlite3
.pytest_cache
.DS_Store
```

3. Créer le commit initial du projet:

```
git commit -m "Initialisation du projet Django avec Poetry et
configuration des tests"
```

4. Créer la branche de développement pour suivre le principe de git flow:

```
git checkout -b develop
```