Projet n° 13

Projet final : prêt pour le feu d'artifices ?

LEKISHVILI Rati (Parcours - DA Python)

Introduction

- Ce document fait un bilan de mon projet final sur le parcours de développeur d'application python.

- L'objectif de ce document est de parcourir une grande partie des étapes menant à l'aboutissement de ce projet, en détaillant les difficultés rencontrées ainsi que les solutions trouvées.

- Dans ce document seront passés en revue les points ci-dessous :

    1. Choisir un sujet.
    2. Méthodologie de travail.
    3. Dossier de spécifications fonctionnelles.
    4. Dossier de spécifications techniques.
    5. Développement de l'Application Web.
    6. Tests.
    7. Déploiement sur Digital Ocean.

1. Choisir un sujet

- Pour le projet final, j'ai choisi la première option de l'énoncé, à savoir « **mener un projet numérique libre répondant à un besoin** ». Je me suis fixé comme objectif de créer un outil facilitant la communication entre le personnel soignant des EHPAD et les proches des résidents. Mon projet a bien évidement une portée sociale et ne vise en aucun cas la réalisation de profits financiers.

- Ci-dessous mes critères de sélection :
    - Le thème choisi doit être un sujet auquel j'accorde une importance particulière.
    - Je dois être en mesure d'entrer en contact avec un professionnel du secteur, pour avoir son point de vu technique. Cela m'aidera à réaliser une application de qualité.
    - Essayer d'aider un maximum de personnes.
    - Privilégier un problème récent, car la probabilité de ne pas avoir suffisamment de solutions efficaces est plus importante.

2. Méthodologie de travail

- Pour ce projet, il était demandé d'implémenter, autant que possible, les 12 bonnes pratiques de l'Xtreme programming.

- Ci-dessous, les principales pratiques que j'ai pu implémenter dans le cadre du développement de mon application :
    - Un **rythme soutenable** d'environ 40 heures par semaine.
    - **Jeu du planning** à l'aide d'un tableau Trello, avec les délais de livraisons pour chaque objectif.
    - **Petites livraisons**. Dès qu'une fonctionnalité était terminée, j'actualisais mon dépôt sur Github.
    - **Client sur site**. Se mettre dans la peau du client pour développer les fonctionnalités.
    - **Les tests**. Tests fonctionnels et tests unitaires.
    - **Refactoring**. Travailler par itération pour améliorer la qualité du code.
    - **Convention de nommage**. Respect des normes PEP8.

2. Méthodologie de travail

- Difficulté rencontrée : Le **refactoring**. Il était à mon niveau parfois compliqué de remarquer des améliorations à apporter sur certaines parties de mon code.
  - Solution : Mon mentor a su me conseiller sur les changements à apporter.

- Difficulté rencontrée : **Client sur site**. Je n'ai pas d'expériences personnelles pour me rendre compte du quotidien d'un EHPAD, ni du point de vu du personnel soignant, ni pour celui des résidents.
  - Solution : Une personne de ma famille, travaillant dans un EHPAD en qualité d'infirmière m'a aidé à mieux comprendre les habitudes et le quotidien qui rythme leurs journées.

3. Dossier de spécifications fonctionnelles

- Les spécifications fonctionnelles ont pour objectif d'expliquer dans les détails les fonctionnalités dont le client aura besoin à travers l'utilisation de l'application.

- Ci-dessous, la liste d'outils utilisé pour identifier les fonctionnalités à développer :
  - **Diagramme de contexte** pour repérer les acteurs principaux.
  - **Les Personas** afin d'identifier les attributs et caractéristiques d'un groupe cible.
  - **Diagramme d'Impact**, permet de lister les fonctionnalités à développer en fonction des utilisateurs.
  - **Diagramme de package,** schématise les domaines fonctionnels qui composent le système.
  - **Diagrammes de cas d'utilisations** et scénarios pour décrire en détail chaque fonctionnalité.
  - **Diagramme de séquence**, afin de schématiser comment et dans quel ordre plusieurs objets interagissent.
  - **Diagramme d'activité** pour représenter un flux d'évènements dans un processus.

3. Dossier de spécifications fonctionnelles

- Difficulté rencontrée : Imaginer les fonctionnalités à développer, réellement utiles pour un professionnel, une personne exerçant le rôle de soignant dans un EHPAD.
  - Solution : Une personne de ma famille exerçant en qualité d'infirmière dans un EHPAD m'a apporté son aide.

- Difficulté rencontrée : Respect des règles de rédaction des différents diagrammes.
  - Solution : Diagramme de cas d'utilisation.

    -> https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-cas-utilisation
  - Solution : Persona

    -> https://www.easybear.fr/blog/comment-creer-persona
  - Solution : Diagramme de séquence

    -> https://www.lucidchart.com/pages/fr/diagramme-de-sequence-uml
  - Solution : Diagramme d'activité

    -> https://creately.com/blog/fr/uncategorized-fr/tutorial-diagrammes-de-activite-uml/

4. Dossier de spécifications techniques

- Les spécifications techniques détaillent comment l'application développée va fonctionner, avec quelles technologies, quelle architecture et quel matériel.

- Ci-dessous les parties composants mon document de spécifications techniques :
    - **Modèle physique de données**, représentant les objets de données relationnelles et leurs relations.
    - **Diagramme de classes**, décrivant le domaine fonctionnelle et illustrant les attributs des classes.
    - **Diagramme de composants**, identifiant les relations entre les différents composants du système.
    - **Diagramme de déploiement**, schématisant la disposition physique des ressources matérielles qui compose le système.

4. Dossier de spécifications techniques

- Difficulté rencontrée : Respect des règles de créations des diagrammes.
  - Solution : Diagramme de classes.

    -> https://laurent-audibert.developpez.com/Cours-UML/?page=diagramme-classes
  - Solution : Diagramme de composants.

    -> https://www.lucidchart.com/pages/fr/diagramme-de-composants-uml
  - Solution : Diagramme de déploiement.

    -> https://www.lucidchart.com/pages/fr/diagramme-de-deploiement-uml

- Difficulté rencontrée : Choix du serveur web.
  - Solution : J'ai choisi Apache 2 (et non de NGINX + Gunicorn) car je n'avais jamais déployé d'application Django à l'aide de ce serveur web auparavant.

5. Développement de l'Application Web

- Afin d'être en mesure de déployer l'application sur un serveur web, il est nécessaire de la préparer au mieux en local.

- Ci-dessous les étapes de création de mon application web en local :
  - **Préparation d'un environnement de développement** : environnement virtuel, installation de django …
  - **Développement du projet** : écriture des classes, méthodes, fonctions, routes, vues …
  - **Refactoring et Itérations** : des corrections apportées tout au long du projet, pour procéder à des ajustements quand cela était utile.
  - **Utilisation d'un système de gestion de version** (Git et Github) : une actualisation fréquente du dépôt GitHub, pour progresser pas à pas, et pouvoir revenir à tout moment sur une version précédente si on a créé un un dysfonctionnement qu'on ignore comment corriger par exemple.
  - **Créations des tests** : tests unitaires, tests fonctionnels, respect des normes PEP8 et vérification du taux de couverture.

5. Développement de l'Application Web

- Difficulté rencontrée : Procéder aux réajustements, en revenant sur le travail déjà réalisé, et qui de plus fonctionne correctement n'est pas simple. Cela demande des efforts d'itérations constants, et parfois beaucoup de modifications sont à apporter au projet en raison du changement de point de vu sur une fonctionnalité.

- Difficulté rencontrée : Comprendre le fonctionnement et les possibilités qu'offre le framework Django. Par exemple l'utilisation des vues génériques de Django, développées pour une conception plus rapide et optimisée des applications web.
  - Solution : Documentation de Django.

    -> https://docs.djangoproject.com/fr/4.0/topics/class-based-views/generic-display/

    -> https://docs.djangoproject.com/fr/4.0/

6. Tests

- Les fonctionnalités étant toutes terminées et les tests écrits pour ces dernières, les commandes ci-dessous permettent de vérifier la validation des tests ainsi que de connaitre le taux de couverture.

python3 manage.py test

```
(env) ratilekishvili@macbook-pro-de-rati daily_diary % python3 manage.py test
Found 24 test(s).
Creating test database for alias 'default'...
System check identified no issues (0 silenced).
........................
----------------------------------------------------------------------
Ran 24 tests in 4.213s

OK
Destroying test database for alias 'default'...
(env) ratilekishvili@macbook-pro-de-rati daily_diary %
```

coverage run manage.py test -v 2 && coverage report

```
users/migrations/0028_alter_profile_status.py                                                  4      0   100%
users/migrations/0029_alter_profile_status.py                                                  4      0   100%
users/migrations/0030_alter_profile_status.py                                                  4      0   100%
users/migrations/0031_resident_relatives.py                                                    4      0   100%
users/migrations/0032_resident_activities_resident_floor_resident_material_and_more.py         4      0   100%
users/migrations/__init__.py                                                                   0      0   100%
users/models.py                                                                               49      8    84%
users/signals.py                                                                              11      0   100%
users/tests.py                                                                                 1      0   100%
users/views.py                                                                                57      7    88%
-----------------------------------------------------------------------------------------------------------
TOTAL                                                                                        617     27    96%
(env) ratilekishvili@macbook-pro-de-rati daily_diary %
```

6. Tests

- Difficulté rencontrée : Utiliser « TestCase ».
  - Solution : Documentation de Django sur l'écriture et lancement des tests.

    -> https://docs.djangoproject.com/fr/3.2/topics/testing/overview/

- Difficulté rencontrée : Taux de couverture des tests sur le projet.
  - Solution : Documentation de coverage.py

    -> https://coverage.readthedocs.io/en/6.3.2/

7. Déploiement sur Digital Ocean

- Une fois l'application terminée en local et vérifiée à l'aide des tests, j'ai décidé de la déployer sur une « Infrastructure As A Service », et plus précisément sur la plateforme de Digital Ocean, car je possédais déjà un compte.

- Ci-dessous les étapes du déploiement :
    - Création d'un espace serveur : **dailydiary.**
    - Création d'un nouvel utilisateur avec l'autorisation de la commande **sudo.**
    - Installation et modifications d'un pare-feu autorisant les connexions en **http** et **ssh**.
    - Téléchargement de l'application depuis mon ordinateur vers l'espace serveur.
    - Installation des dépendances et environnement virtuel sur Digital Ocean.

7. Déploiement sur Digital Ocean

- Difficulté rencontrée : Problème survenu lors des chargements des fichiers statiques. La feuille de style main.css n'étant pas accessible, l'interface utilisateur de mon application était compromise.
  - Solution : Digital Ocean fournit des tutoriels pour collecter les fichiers statiques efficacement.

    -> https://www.digitalocean.com/community/tutorials/working-with-django-templates-static-files
- Difficulté rencontrée : Première installation et configuration d'Apache 2.
  - Solution : Documentation d'Apache 2.

    -> https://doc.ubuntu-fr.org/apache2#installation
  - Solution : Article sur le déploiement des applications web Django à l'aide du serveur web Apache2 sur Digital Ocean.

    -> https://www.digitalocean.com/community/tutorials/how-to-serve-django-applications-with-apache-and-mod_wsgi-on-ubuntu-16-04

Les liens du projet

- Tableau de bord Trello :

  -> https://trello.com/invite/b/TPC4w88f/3ae701df9e31ff858ff48c4ac9776d8e/p13-projet-final

- Dépôt sur Git Hub :

  -> https://github.com/LekishviliRati/P13_daily_diary_application.git

- Site Web :

  -> http://165.227.235.193


The End