
Cahier de reprise

Ce document a pour objectif de présenter le travail réalisé sur ce projet et de permettre à quiconque d'avoir les clés en main afin de reprendre ce projet.

I) Contexte du projet

Tout d'abord, ce projet a pour but d'aider notre cliente, Marie-Gabrielle Pujol qui est Ingénieure organisation au sein de la Direction de la Stratégie Cellule organisation et méthode au CHRU de Tours. Une de ses missions est en lien avec la Maison de Médecine de Garde de Trousseau qui permet des consultations assurées par un médecin généraliste libéral sur des créneaux spécifiques avec des médecins inscrits sur la base du volontariat pour des vacations d'une durée de 4 heures.

Les créneaux sont ouverts de 20h à minuit du lundi au vendredi et de 12h à minuit le samedi, dimanche et les jours fériés. Ces créneaux sont mis à disposition sur Doctolib et sont réservables par le SAU (Service d'Accueil Urgences), le SAS Ambulatoire (Service d'Accès aux Soins), les médecins de la MMG et les Médecins Régulateurs Libéraux. De plus, les rendez-vous pris ont une durée de 15 minutes.

Les données de ces rendez-vous peuvent être extraites de Doctolib dans un format .csv. Dans une démarche de confidentialité et de sécurité, ces données sont anonymisées. Nous avons les informations suivantes : la date de début du rendez-vous, l'heure de début du rendez-vous, le motif du RDV, des Notes, la Date de saisie, la Date de dernière mise à jour, qui a créé le rendez-vous (SAU, MRL...), le Statut (Vu, à venir Absent), Date de naissance, Heure d'arrivée, Heure de départ. A partir de ces données Mme Pujol devait calculer sur excel différentes statistiques comme l'âge des patients afin de les trier par tranche d'âge, mais aussi le nombre de rendez vous honorés, le nombre de vacations potentielles, couvertes, le nombre de rendez vous potentiels, honorés, couvert, pris, mais aussi la répartition de la prise de rendez-vous sur Doctolib en fonction des sources (ex : SAU, 15).

La répartition des motifs, mais aussi si les médecins quittent leur poste ou prennent une pause car il le signifie à la place d'un rendez-vous.

L'objectif de toutes ces statistiques est de pouvoir suivre et présenter la progression de l'activité de la MMG dans sa globalité que ce soit du côté des patients ou des médecins.

II) Description des données

Les données manipulées par l'application proviennent d'une extraction de Doctolib au format csv. Le fichier doit contenir les données qui correspondent à l'entête suivante:

"Date de début;Début;Motif du RDV;Notes;Date de saisie;Date de dernière mise à jour;Créé par; Statut;Date de naissance;Heure d'arrivée;Heure de prise en charge;Heure de départ".

Comme l'exemple suivant :

05/01/2023;20h00;Céphalées;;2023-01-05T14:25:55.988+01:00;2023-01-05T20:45:13.477+01:00;Urgence1MMG;Vu;16/04/2007;2023-01-05T20:45:13.161+01:00;;2023-01-05T20:45:13.464+01:00

III) Fonctionnalités de l'application

1) Import

L'application permet d'ajouter les données d'un fichier CSV issu de Doctolib. Un clic sur le bouton "Importer" ouvre une fenêtre permettant de sélectionner le fichier à importer. Un seul fichier peut être importé à la fois, de sorte que l'importation d'un nouveau fichier remplace le fichier déjà chargé. Dans le cas d'un fichier incorrect, les exceptions gérées sont les suivantes:

- Fichier vide
- Pas d'entête dans le fichier
- Uniquement l'entête dans le fichier
- Mauvais type de fichier

Les erreurs sont affichées dans les logs du serveur sous la forme d'un titre et d'une description. Cependant, l'erreur générique n'affiche que la description. Pour l'instant, il n'y a pas de message d'erreur sur l'interface graphique. En cas d'erreur durant l'importation du CSV, les données déjà présentes dans l'application sont supprimées.

La partie importation est la plus facile à mettre en place au début, mais également la plus difficile à rendre robuste. Il faut gérer tous les problèmes du fichier CSV, par exemple une date de naissance en l'an 4197, des consultations de 30 secondes, des sauts de ligne et des tabulations dans le fichier. Nous utilisons la bibliothèque pandas pour trier les données, mais certains cas restent à traiter. Pour l'instant, la base de données est créée par pandas avec des types textuels qui sont

ensuite convertis. Une création personnalisée permettrait de changer le type des variables et de vérifier leur cohérence. Par ailleurs, la gestion de plusieurs fichiers permettrait de comparer des données d'une année à l'autre, par exemple le mois de novembre 2023 avec le mois de novembre 2024.

2) Filtrage

Actuellement, l'application permet de gérer une liste de filtres pour la colonne "Notes". Pour pouvoir s'adapter à de futurs fichiers de données présentant de nouvelles irrégularités, la liste des mots à exclure est sauvegardée dans le fichier ``mot_a_retirer.txt``. Ce fichier est chargé lors de l'importation du CSV. Nous réécrivons le fichier lorsqu'une entrée est modifiée, ajoutée ou supprimée dans la partie filtre de l'application. Dans le cas où l'application n'arrive pas à charger le fichier, nous considérons qu'il n'y a pas de filtre sur la colonne. Pour le moment, l'interface graphique des filtres est peu ergonomique, notamment l'ajout de nouveaux filtres qui n'est pas intuitif.

3) Export

L'application permet également d'exporter les données présentes dans la base de données. Un export est possible sur la base de données affichée sur la page principale de l'application, uniquement sous le format csv pour le moment (possibilité d'évolution afin de produire différents types de données, notamment excel). Cet export de données ouvre une fenêtre qui permet à l'utilisateur de choisir où stocker le fichier et comment le nommer. L'organisation des données dans le fichier est la même que dans les fichiers de doctolib, ainsi il est possible d'importer le fichier si besoin.

Sur la page des graphiques (et le tableau de bord, bien que le lien entre les graphiques et le tableau de bord ne soit pas fait), il est également possible d'exporter les graphiques par un simple "clic droit > enregistrer sous". Nous avons une fonction pour exporter les diagrammes avant de passer sur Django, mais puisque l'export des images est déjà possible sur un navigateur web, cette fonction a été rendue redondante et nous avons décidé de l'enlever.

Finalement, nous n'avons malheureusement pas eu le temps de faire l'export des indicateurs, bien que leur génération fonctionne.

4) Graphiques

L'application permet à l'utilisateur de visualiser les données de deux manières. Sur la page « Graphiques », l'utilisateur peut choisir entre 4 options : Age, Créé par, RDV (couvert/fait) et RDV honorés. Pour les deux premiers, un diagramme circulaire est affiché et pour le second, un diagramme à barres.

La page « Indicateurs » présente les données sous forme de tableau, divisées par mois. Il existe 6 options différentes disponibles.

Afficher les données est assez simple. Il suffit à l'utilisateur de choisir une option et de cliquer sur le bouton « Générer » et les données s'affichent.

Sur les deux pages, un filtrage par date est possible. Si l'utilisateur souhaite voir les données d'une période spécifique, il choisit une date de début et une date de fin via les boutons de date en bas de la page. S'ils ne choisissent pas de dates de début et de fin, toutes les données sont affichées. De plus, si l'utilisateur choisit uniquement la date de fin, les données depuis le début du temps dans un fichier importé jusqu'à la date de fin choisie sont affichées. S'ils choisissent uniquement une date de début, les données de cette date jusqu'à la dernière date du fichier sont affichées. Si les dates choisies ne correspondent pas à la période du fichier importé, un graphique ou un tableau vide s'affiche. Pour les travaux futurs, il serait bon d'afficher un message d'erreur pour ces cas qui indiquerait ce qui ne va pas et quoi faire.

Il existe également un bouton « Ajouter au tableau de bord », dont le but est de choisir le graphique ou le tableau spécifique à afficher sur le tableau de bord, mais nous n'avons pas eu assez de temps pour le faire fonctionner, le tableau de bord est donc encore à réaliser.

Pour cette partie du projet, Python a été utilisé. Pandas est une bibliothèque Python que nous avons utilisée pour manipuler les données, comme le filtrage par date, le regroupement par mois et également pour choisir les données à afficher sur la page. Les bibliothèques Python peuvent être installées en exécutant simplement une instruction dans la ligne de commande : « pip install <library_name> ».

Afin de connecter la partie python (filtrer les données, tracer les graphiques, réaliser les tableaux...) avec son affichage à l'écran, JavaScript a été utilisé en envoyant des HtmlRequests et des HtmlResponses. Les HtmlResponses contiennent une image pour la partie graphique et HtmlTable pour la partie indicateurs.

5) Interface

L'interface a été développée principalement en HTML, CSS et JavaScript de par l'utilisation du framework Django. Chaque vue donne lieu à un fichier HTML et si besoin un fichier JavaScript. Notamment pour les pop-ups. Un fichier CSS est commun à tous les fichiers HTML.

L'intégration du python dans ses fichiers est effectuée dans les fichiers HTML. Toute l'interface est actuellement fonctionnelle, seul le tableau de bord doit être retravaillé afin d'inclure les tableaux des indicateurs en plus des images des graphiques.

Un travail conséquent a été fait sur le CSS afin de rendre l'utilisation de l'application agréable.

IV) Technologies utilisées

Toutes informations importantes (Id de connexion si il y en a, schéma de la base de données).

1) Python

Nous avons choisi d'utiliser Python car c'est un langage que nous connaissons tous bien et qui est facile d'utilisation. De plus, la création de graphiques est très simple avec les librairies python. Nous avons choisi d'utiliser QT Python pour faire l'interface mais suite à divers problèmes avec QT nous sommes passés au framework Django.

2) Py Plots

Pour tracer les graphiques, la bibliothèque matplotlib a été utilisée. C'est une bibliothèque qui permet à l'utilisateur de tracer différents types de graphiques, comme des graphiques à barres, des diagrammes circulaires, des graphiques linéaires, etc. Il est possible de configurer les titres des graphiques, les étiquettes des données, de mettre les légendes, de choisir les couleurs pour le graphique, d'adapter la taille...

3) Django

Après un essai qui n'a pas abouti avec PyQT, nous avons décidé de changer de framework pour Django. Son principal avantage a été son intégration dans l'IDE Pycharm ainsi que le développement du frontend qui pouvait se faire avec du HTML, CSS et JavaScript. Il est nécessaire de bien configurer son environnement python pour faire fonctionner le projet en ajoutant une configuration Django Server, c'est ce qui a été fait avec PyCharm. Cela permettra de choisir la version de python que vous souhaitez, ainsi que le port sur

lequel vous souhaitez être. Il peut être nécessaire de toucher aux variables d'environnements mais ce n'est pas obligatoire.

4) BDD

On utilise SQLite pour avoir une base de données qui centralise les informations contenue dans le fichier csv. On utilise la base de données pour stocker et sélectionner les informations lors de la création des graphiques et des indicateurs. Pour se connecter on utilise la librairie SQLite3. La base de données est créée à chaque fois dans le même format que le fichier csv.