
TARIF'PREP



NANCY

MANUEL D'UTILISATION

Mehdi SHAHID, stagiaire été 2017

mehdi.shahid@telecomnancy.net

18/07/2017

SOMMAIRE

1. PRÉSENTATION DE L'APPLICATION

| | |
|--------------------------------|-----------|
| 1.1. FONCTION PRINCIPALE | PAGE 3/14 |
| 1.2. FONCTIONNALITÉS | PAGE 3/14 |
| 1.3. INTERFACE | PAGE 4/14 |

2. ANALYSE DES RÉSULTATS

| | |
|---------------------------------------|-----------|
| 2.2. PAGE DE RÉSULTATS ET BLOCS | PAGE 8/14 |
|---------------------------------------|-----------|

3. REMARQUES ET COMPLÉMENTS TECHNIQUES

| | |
|---|------------|
| 3.1. TECHNOLOGIES UTILISÉES, STRUCTURE DU PROJET ET TRAITEMENT DES DONNÉES | PAGE 11/14 |
| 3.2. STRUCTURE DES FICHIERS EXCEL | PAGE 13/14 |
| 3.3. DEBUGGING | PAGE 14/14 |

1. PRÉSENTATION DE L'APPLICATION

Park'Tarif Nancy (dite 'PTN') est une application de modélisation et d'aide à la prise de décision développée par Mehdi SHAHID (mehdi.shahid@telecomnancy.net) durant son stage de fin de deuxième année à TELECOM Nancy.

1.1. FONCTION PRINCIPALE

La fonction principale de PTN est de fournir grâce à des données de fréquentation d'un parking donné une simulation du chiffre d'affaires engendré par une grille tarifaire potentielle que l'utilisateur peut créer et modifier.

L'utilisateur peut ainsi comparer (dans une certaine marge d'erreur qui est également calculée) l'éventuel bénéfice ou déficit que va engendrer la grille tarifaire considérée par rapport au chiffre d'affaire dit 'réel' qui est issu des fréquentations du mois considéré.

On note également que l'application permet d'aisément détecter les anomalies de paiement lorsqu'une marge d'erreur est trop importante pour une voiture donnée. Cette donnée est visible facilement sur un graphe dynamique adapté à la simulation.

1.2. FONCTIONNALITÉS

Pour répondre à cette fonction principale, on dénote les fonctionnalités suivantes:

- conversion d'un fichier Excel en données utilisables par l'application,
- génération d'une grille tarifaire à partir de données rentrées par l'utilisateur,
- calcul du chiffre d'affaires 'réel' pour le fichier Excel donné,

- calcul du chiffre d'affaire 'estimé' (correspondant à la grille tarifaire en vigueur pour le parking considéré) pour le fichier Excel donné,
- calcul du chiffre d'affaire 'simulé' (correspondant à la nouvelle grille tarifaire entrée par l'utilisateur) pour le fichier Excel donné,
- comparaison de ces différents éléments entre eux pour rendre les données pertinentes,
- visualisation graphique et dynamique de ces dites données.


1.3. INTERFACE


Dès son arrivée sur l'application, l'utilisateur est présente cette page:


The screenshot displays a user interface with a light gray background. At the top, there is a navigation bar with a gear icon on the left and two icons (a square with a plus and a square with a minus) on the right. Below the navigation bar, the text '1. MOIS' is displayed in blue, followed by a calendar icon. A white rectangular box contains the text 'Quel mois souhaitez-vous considérer?' and the word 'Debug' in blue, with a downward arrow on the right. Below this, the text '2. GRILLE TARIFAIRE' is displayed in blue, followed by a grid icon. Another white rectangular box contains the text 'Quelle grille tarifaire souhaitez-vous considérer?' and the text 'Toutes_options' in blue, with a downward arrow on the right. At the bottom center, there is a blue circular button with a white wrench icon.

Cette page, dite '**page principale**' ou '**page de pré-simulation**' est la page permettant à l'utilisateur de sélectionner le mois et la grille tarifaire à considérer pour lancer la simulation en appuyant sur le bouton bleu ou en appuyant sur 'entrée'.


En cliquant sur la deuxième icône du menu, l'utilisateur est alors redirigé vers la page suivante:









uploadData(*tableName*)




Fonction permettant de transformer un fichier excel en données utilisables par l'application.

Merci de fournir un fichier au format *.xls(x).






deleteData(*tableName*)




Fonction permettant de supprimer des données de fréquentations.


 Attention, cette opération est irrévocable.

Quel mois souhaitez-vous supprimer?


Fevrier_2017




getPricingPolicy(*parkingName*)




Fonction permettant de générer une grille tarifaire utilisable par l'application.



deleteData(*parkingName*)




Fonction permettant de supprimer des grilles tarifaires.

 Attention, cette opération est irrévocable.

Quelle grille souhaitez-vous supprimer?

Toutes_options



Page 5/14

C'est sur cette page que l'utilisateur peut fournir les données nécessaires au bon fonctionnement de l'application.




uploadData()

En fournissant le fichier Excel correspondant aux données de fréquentation du parking considéré sur le mois considéré, cette fonction va automatiquement analyser le fichier et convertir ces données.

L'utilisateur est alors redirigé vers la page de **pré-simulation** sur laquelle il retrouvera le mois considéré dans le menu déroulant prévu à cet effet.

getPricingPolicy()

En cliquant sur l'icône, la page suivante apparaît:



GÉNÉRATEUR DE GRILLE TARIFAIRE

Données générales

NOM DU PARKING

MINUTES GRATUITES

30

Tarification à la journée

PRIX INITIAL

0

| | | | | | |
|-------|-------|-------|-------|-------|-------|
| 1H | 2H | 3H | 4H | 5H | 6H |
| 1.6 | 4.40 | 6 | 7.20 | 8.40 | 9.60 |
| 7H | 8H | 9H | 10H | 11H | 12H |
| 10.80 | 12 | 12.80 | 13.60 | 14.20 | 15.20 |
| 12H | 13H | 14H | 15H | 16H | 17H |
| 15.20 | 15.60 | 16.00 | 16.40 | 16.80 | 17.20 |
| 18H | 19H | 20H | 21H | 22H | 23H |
| 17.6 | 18.00 | 18.40 | 18.80 | 19.20 | 19.60 |
| 24H | | | | | |
| 20.00 | | | | | |

Tarification de nuit et dimanche

Tarif spécial le dimanche

TARIF <4H

TARIF 4-12H

TARIF >24h

Oui

▼ 1

2

0.9



L'utilisateur peut alors rentrer toutes les données nécessaires à la génération d'une grille tarifaire selon les options désirées.

Celui-ci sera alors redirigé vers la page de **pré-simulation** qui sera mise à jour avec la nouvelle grille.

deleteData()

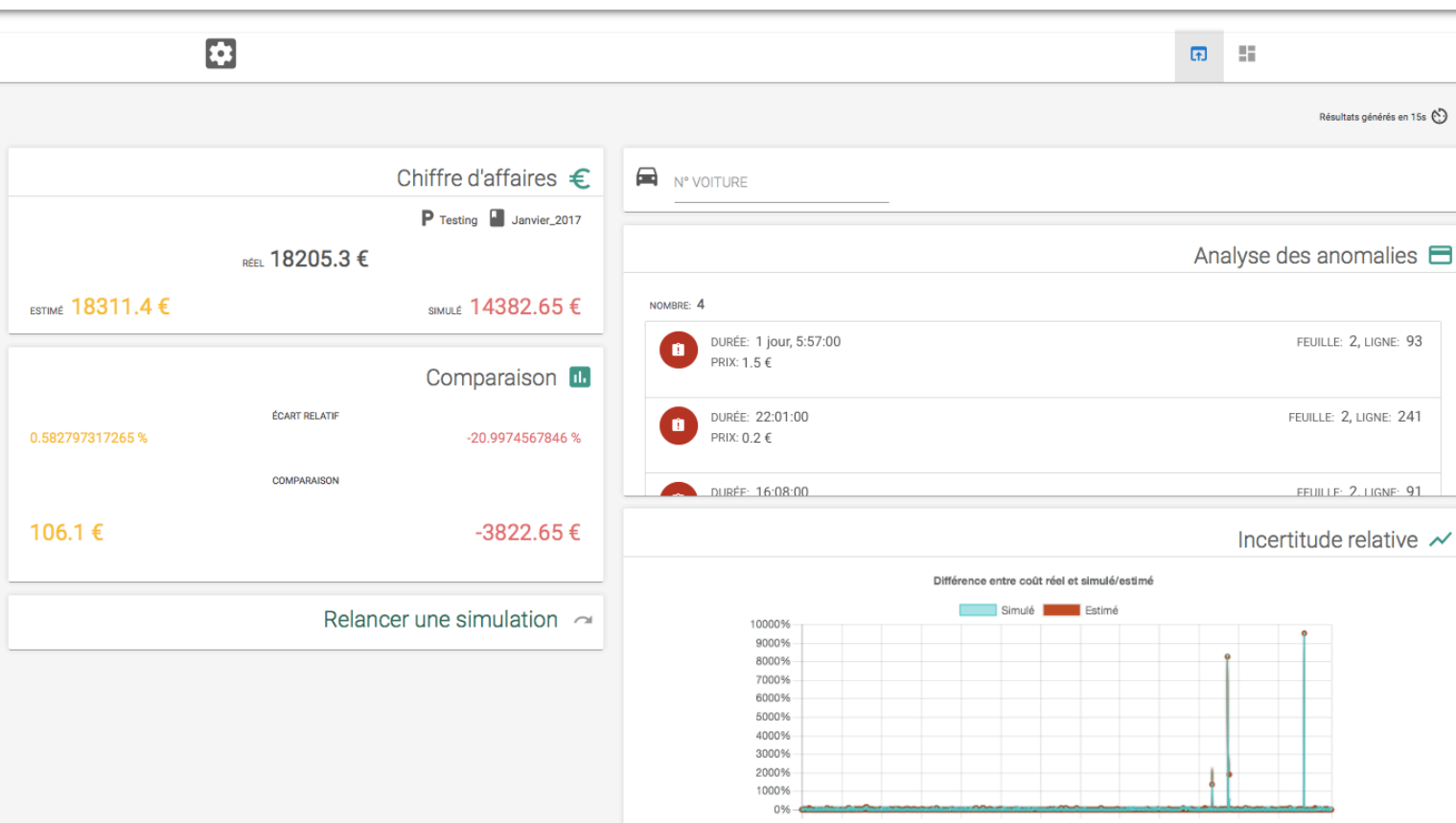
Les deux fonctions deleteData() permettent respectivement de supprimer les données de fréquentations pour le mois considéré et de supprimer la grille tarifaire considérée. Il est important de noter que cette opération est irrévocable et que la récupération des données n'est alors plus possible.

2. ANALYSE DES RÉSULTATS

Une fois les données générées, à l'aide de la page de **pré-simulation**, l'utilisateur n'a plus qu'à sélectionner dans les menus déroulants le mois qu'il souhaite considérer ainsi que la grille qu'il souhaite simuler.

2.2. PAGE DE RÉSULTATS ET BLOCS

La simulation terminée, la page suivante est affichée:



Décomposons et précisons les différents blocs:

1) Chiffre d'affaires

Ce bloc permet d'afficher pour le mois considéré et la grille tarifaire de simulation les différents chiffres d'affaires estimés par l'application.

Soit:

- **le chiffre d'affaires 'réel'** qui correspond au chiffre d'affaires obtenu en sommant toutes les transactions du fichier Excel fourni par l'utilisateur. On note que celui-ci ne correspond pas exactement à celui obtenu en sommant "manuellement" depuis l'Excel pour une raison qui m'échappe. La marge d'erreur est cependant insignifiante à la vue de l'utilisation de cette donnée.
- **le chiffre d'affaires 'estimé'** qui correspond à un montant calculé 'automatiquement' à partir d'une modélisation de la grille tarifaire en vigueur. On note une marge d'erreur d'environ dix centimes d'euros sur une grande majorité des cas considérés, hormis anomalies (sujet abordé plus bas dans le manuel).
- **le chiffre d'affaires 'simulé'** qui correspond au montant obtenu par modélisation de la grille tarifaire fournie par l'utilisateur.

2) Comparaison

Cette partie aide l'utilisateur à obtenir une meilleure compréhension des données en les comparant entre elles. On trouve l'écart relatif entre le montant estimé et le montant réel à gauche, et l'écart relatif pour le montant simulé à droite. La différence brute entre les deux montants est également affichée en dessous.

3) Tarification individuelle

C'est une zone clé de l'analyse des résultats. En liant cette zone avec la zone d'incertitude en dessous, l'utilisateur peut rentrer un numéro de voiture particulier (les voitures sont numérotées dans le même ordre que les lignes du fichier Excel de base, les plaques d'immatriculation n'étant pas présentes pour des raisons d'anonymat) et obtenir les données relatives à la voiture considérée.

Il peut donc analyser plus en détail les différents tarifs et détecter le mode de tarification qui est problématique pour la voiture considérée.

4) Incertitude relative et détection des anomalies

En adéquation avec les blocs avoisinants, l'utilisateur peut remonter aux lignes concernées dans le fichier Excel fourni pour la simulation et éventuellement remonter l'anomalie au prestataire. Les anomalies sont listées par ordre décroissant de la durée de séjour du client pour accentuer l'amplitude de l'erreur commise.

5) Incertitude relative et visualisation graphique des anomalies

Cette dernière partie permet à l'utilisateur d'obtenir une visualisation graphique de l'incertitude relative. En abscisse se trouve les différentes voitures ayant fréquenté le parking pour le mois considéré; on trouve en ordonnée le montant de l'incertitude relative. On peut alors discerner la marge d'erreur à la fois pour le montant estimé et pour le montant simulé. En liant cette partie au bloc de tarification individuelle, nous obtenons un outil d'analyse et de détection des anomalies de paiement.

3. REMARQUES ET COMPLÉMENTS TECHNIQUES

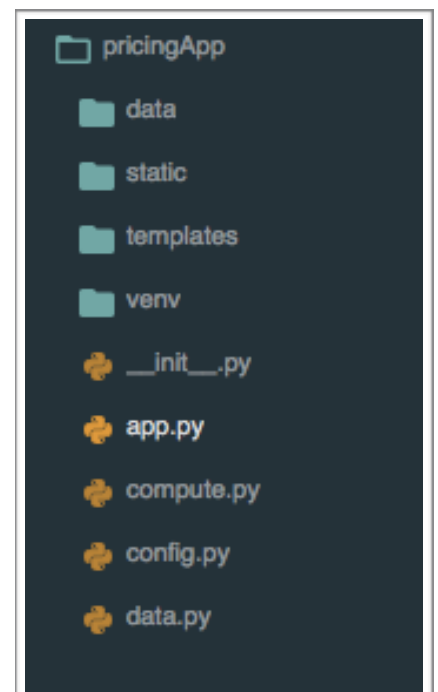
Cette partie a pour vocation d'aider à la compréhension du côté technique du projet et devrait permettre, je l'espère, d'au moins limiter le temps de compréhension du projet par le personnel compétent. Les mots jugés clés ont été mis en gras.

3.1. TECHNOLOGIES UTILISÉES, STRUCTURE DU PROJET ET TRAITEMENT DES DONNÉES

La partie front-end de l'application correspond à du simple HTML+CSS/JavaScript et se base graphiquement sur le framework **Materialize** de Google. Le coeur de l'application et le back-end est porté par le framework **Flask**, en python.

Le projet possède la structure suivante:

- ◆ Le dossier '**data**' contient tous les fichiers nécessaires au bon fonctionnement de la simulation, à savoir les grilles tarifaires (qui correspondent à des fichiers au format ***.JSON**) et les Excel de fréquentations (inchangés).
- ◆ Le dossier '**static**' contient tous les fichiers au format ***.css** et ***.js** utiles au front-end.
- ◆ Le dossier '**templates**' contient tous les fichiers ***.html** permettant de générer la partie front-end en adéquation avec les spécifications de **Flask**.
- ◆ Le fichier '**app.py**' contient le *main*, les fonctions principales de *rendering* des pages, les *helpers* et les fonctions utiles à l'API.



-
- ◆ Le fichier '**compute.py**' est le coeur de l'algorithmique de l'application. Il héberge toutes les fonctions de tarification.
 - ◆ Le fichier '**config.py**' contient les variables d'environnement utiles à **Flask**.
 - ◆ Le fichier '**data.py**' permet le traitement de toutes les données. C'est dans ce fichier que se situe le code nécessaire à la lecture des fichiers Excel et le traitement des données de grille tarifaire.

Pour les fréquentations, le fichier Excel fourni est analysé pour en ressortir toutes les lignes correspondant à une fréquentation, chacune de ces fréquentations est ensuite transmise à une base de données **SQLITE3** dénommée '**transactions.db**' sous le schéma (*id, date d'arrivée, date de départ, prix*).

Pour les grilles tarifaires, le *formulaire* que remplit l'utilisateur permet la génération d'un fichier *.**JSON** qui suit un template prédéfini (tout ceci se trouve dans le dossier '**data**'). Dans le but de lister dynamiquement toutes les grilles tarifaires présentes, *et parce que cela me semblait être la solution de facilité*, une base de données '**pricings.db**' a également été créée et permet uniquement de gérer ce dit recensement selon le schéma basique (*id, date de création, chemin du fichier*).

Les fonctions utiles au traitement de ces données sont réparties dans le fichier '**compute.py**'.

3.2. STRUCTURE DES FICHIERS EXCEL

Précisons la structure des fichiers Excel à fournir à l'application dans le cas où ceux-ci sortiraient de la norme imposée au prestataire.

| Plaque | Stationnement | Montant(s) couvert(s) par chèque parking utilisé (Euro) | | | | Total |
|--------|-----------------------------------|---|------------|------------|------------|-------|
| | | Chèque N°1 | Chèque N°2 | Chèque N°3 | Chèque N°4 | |
| | 1/1/2017 16:28 - 1/2/2017 08:07 | 16,70 | - | - | - | 16,70 |
| | 1/1/2017 16:18 - 1/2/2017 08:07 | 16,80 | - | - | - | 16,80 |
| | 12/31/2016 16:21 - 1/2/2017 08:18 | 39,70 | - | - | - | 39,70 |
| | 12/31/2016 15:50 - 1/2/2017 09:00 | 40,00 | - | - | - | 40,00 |
| | 1/2/2017 08:27 - 1/2/2017 09:01 | 0,80 | - | - | - | 0,80 |

Deux colonnes sont nécessaires au bon fonctionnement de l'application. Une colonne 'stationnement' qui contient les informations sur le datage d'arrivée et de départ de chaque véhicule, et une colonne 'total' qui contient le prix total payé par chaque client.

La première ligne traitée est nécessairement la 16ème (incluse), la colonne 'stationnement' doit nécessairement être la double colonne E-F et la colonne 'total' doit nécessairement être la colonne M. Il peut y avoir autant de lignes que nécessaire, et le programme tiendra compte de toutes les feuilles de l'Excel (**sauf la dernière**, voir plus bas) tant qu'elles respectent chacune les conditions spécifiées ci-dessus.

Le format MM/JJ/AAAA HH:MM pour les dates est standard et doit également être respecté.

La dernière feuille (qui semble toujours correspondre à l'émetteur 'Vitrine Nancy') n'est pas prise en compte. Dans l'éventualité où cette feuille n'est pas présente, il suffit de rajouter une feuille vierge à la fin et d'y insérer des données quelconques.

Cette structure correspond à une norme que le prestataire semble suivre sur l'ensemble des fichiers de fréquentation, il ne devrait donc pas y avoir de problèmes. Dans

le cas où une erreur intervient malgré les instructions ci-dessus, merci de se référer au fichier '**data.py**'.

3.3. DEBUGGING

La partie 3.1. de ce manuel devrait fournir une première piste de réflexion sur la direction dans laquelle chercher la solution, selon la nature de l'éventuel problème rencontré.

On peut cependant dénoter les thématiques suivantes:

| NATURE DU PROBLÈME | FICHIER OÙ REGARDER |
|--------------------------------------|---------------------------------------|
| Interface, affichage, front-end | ./templates/; ./static/; app.py |
| Excel | data.py |
| Grille tarifaire | app.py pour les formulaires ; data.py |
| Tarification, valeurs, algorithmes | compute.py |
| Ordre général, bug de fonctionnement | app.py |