

(Gp121) Rapport Grand Projet Programmation

SEHILI ELIAS¹, TERRÉ DAEL¹, VIGIER LEMONNIER HUGO¹, AND VEYRAT-PARISIEN MATHIS¹

¹ IPSA - Institut Polytechnique des Sciences Avancées, 63 boulevard de Brandebourg, 94 200 Ivry sur seine, Paris, France

* Corresponding author: aybuke.ozturk@ipsa.fr

Compiled April 23, 2021

Ceci est notre compte-rendu du grand projet programmation (GP121) du second semestre d'Aéro1. © 2021

Optical Society of America

<http://dx.doi.org/10.1364/ao.XX.XXXXXX>

1. INTRODUCTION

Notre grand projet programmation a pour objectif de créer un système d'analyse de vol de base pour un aéroport afin d'analyser les données de vol et de passagers en utilisant des bibliothèques pythons. Ce projet nous permet d'acquérir de nouvelles compétences comme l'utilisation de nouvelles bibliothèques pythons, comme Tkinter et Pandas. Cela nous a permis de découvrir et d'utiliser le Latex pour écrire notre rapport. Avec certain ordinateur, la résolution pourrait être modifiée alors l'affichage en sera décallé.

2. LOGIN

Nous avons commencé par faire l'interface graphique de notre système de vol et plus particulièrement l'interface "Login". Cette dernière permet à l'utilisateur de se connecter ou de s'inscrire. Il interagit avec l'interface via des emplacements de saisies de texte comme l'emplacement "email", "password" et aussi avec l'utilisation de boutons pour se connecter ou s'enregistrer. Ces derniers, lorsqu'on les actionnent, lancent les fonctions qui leur sont associés. Les fonctions qui ont été codées sont à retrouver dans le fichier du code source de l'application. Nous nous sommes permis de rajouter quelques fonctionnalités qui n'étaient pas explicitement demandées mais qui nous semblaient importantes. Nous avons donc rajouter la fonctionnalité de cacher le mot de passe lorsque l'utilisateur le saisi (automatiquement appliquée). Cependant si l'utilisateur veut voir son mot de passe, il clique sur le bouton symbolisé par l'oeil et réappuie pour le cacher de nouveau.

A. Connection

Pour se connecter l'utilisateur doit avoir son email et son mot de passe enregistré dans un fichier .csv "user_info". S'il essaie de se connecter sans être enregistré, un message d'erreur apparaîtra. L'utilisateur ayant enregistré au préalable son email et son mot de passe peut alors se connecter et un message de validation s'affichera. De plus, la date et l'heure de sa connexion sera écrite dans un autre fichier .csv "user_login" (son email sera aussi inscrit).

B. Inscription

Pour s'inscrire, il faut que l'utilisateur rentre un email et un mot de passe au choix dans les entrées prévus à cet effet. Après avoir cliqué sur le bouton, l'email et le mot de passe s'inscrivent dans le fichier .csv "user_info", seulement si l'email choisi par l'utilisateur n'est pas déjà existante. Un message de validation s'affichera si tout est correct. Si cette email existe déjà alors il y aura un message d'erreur. Tous les types de mots de passe sont pris en compte comme les chiffres, les lettres et les caractères spéciaux.

The screenshot shows two side-by-side login forms for the "Flight Analyses System".

Top Form (Login):

- Header: Welcome to Flight Analyses Systeme
- Fields: Email (blue placeholder), Password (red placeholder)
- Buttons: Login (grey), Signup (grey), Eye icon (grey)

Bottom Form (Signup):

- Fields: Email (blue placeholder), Password (red placeholder)
- Buttons: Signup (grey)

3. PASSENGERS

Concernant la fenêtre "Passenger", elle permet d'apporter des données sur les passagers. Ces dernières sont contenues dans le fichier .csv "passenger_info". L'utilité de celle-ci est d'obtenir des informations, comme leurs noms, leurs terminaux, leurs zones

d'embarquement ainsi que l'identifiant du passager. Pour ce faire, nous avons dû utiliser la bibliothèque Pandas et Tkinter pour traiter et utiliser les données. La spécificité de cette fenêtre est de pouvoir réaliser à l'aide des différentes bibliothèques, une recherche précise qui s'appuie sur trois critères qui sont le prix, la destination et la compagnie.

4. FLIGHT VISUALIZATION

L'onglet "Flight Visualization" sert à représenter les données contenues dans le fichier .csv "flight_info". Ainsi, nous avons dû utiliser la bibliothèque Pandas pour pouvoir traiter ces données, la bibliothèque Tkinter pour les placer dans l'application et les agencer dans l'interface graphique. Nous avons aussi construit des graphiques via l'utilisation des données et de la bibliothèque Matplotlib. Ces graphiques nous permettent de représenter le détail des vols.

5. L'INTERFACE GRAPHIQUE

Nous avons parlé de ce point dans toutes les sections précédentes, mais nous allons compléter l'explication de notre travail. L'interface graphique est le squelette de notre application et c'est la partie qui est en contact direct avec les actions de l'utilisateur. Nous avons donc défini une fenêtre et défini des onglets. Les trois onglets comme vus dans les parties précédentes sont : "Login", "Passengers" et "Flight Visualization". Nous avons aussi placé un logo d'avion en haut à gauche pour la symbolique et pour rappeler le domaine de l'aviation. Nous avons agencé les différentes parties ensemble et avons placé les différents tableaux et graphiques dans les parties correspondantes.

6. RÉPONSES AUX QUESTIONS

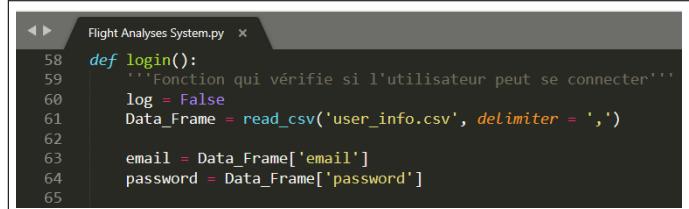
1) Nous avons utilisé les bibliothèques suivantes: Tkinter, Pandas, Matplotlib, numpy, datetime, time. Tkinter sert à créer une fenêtre et faire une interface graphique. C'est ce que l'utilisateur va voir et c'est sur cette interface qu'il va interagir. Pandas sert à manipuler et analyser des données. Matplotlib nous permet de faire des graphiques via des données qui sont traitées avec Pandas. L'utilisation de Matplotlib est liée à numpy. Ces deux bibliothèques fonctionnent ensemble, comme les bibliothèques time et datetime.

2) Les widgets Tkinter utilisés pour ce projet sont:

- les Boutton
- les Label
- les Entry
- les Canvas
- les Treeview

3) Le gestionnaire de mise en page que nous avons utilisé avec Tkinter est place(). Tkinter possède trois gestionnaires de mise en page. Les autres étant pack() et grid().

4) Pour accéder à une liste de valeurs d'une colonne avec Pandas dataframe, nous procédons comme expliqué ci-dessous. L'explication est en lien avec la fonction Login dans notre code.



```

58 def login():
59     '''Fonction qui vérifie si l'utilisateur peut se connecter'''
60     log = False
61     Data_Frame = read_csv('user_info.csv', delimiter = ',')
62
63     email = Data_Frame['email']
64     password = Data_Frame['password']
65

```

Pour commencer, nous créons une variable Data_Frame et lui assignons la méthode read_csv() qui nous permet de lire le fichier .csv. Ensuite, si nous voulons accéder aux valeurs de la colonne email alors nous créons une variable email. Cette dernière contient les valeurs de la colonne du fichier .csv car nous assignons à la variable email : Data_Frame["email"].

5) Pour ajouter de nouvelles données dans un dataframe Pandas, nous avons utilisé la méthode .loc[index] qui permet d'ajouter une ligne à un dataframe Pandas. Pour ajouter de nouvelles données nous pouvons aussi utiliser la méthode append().

6) Nous avons utilisé la fonction .get() qui permet de récupérer le contenu d'une "Entry", et de ce fait de récupérer ce que l'utilisateur écrit dans la zone de saisie.

Nous avons utilisé la fonction .to_csv() qui permet d'écrire le contenu d'une variable dans le fichier .csv. Cette fonction prend en paramètre le nom du fichier où elle doit écrire et le séparateur des éléments entre autres.

Nous avons utilisé la fonction .place() qui nous permet de placer un bouton, une zone de texte ou un label dans l'interface graphique (Tkinter). Cette fonction prend en paramètre le positionnement en x et en y (horizontal et vertical) via relx et rely, sous forme d'un nombre flottant entre 0.0 et 1.0. Elle prend aussi en paramètre la largeur et la hauteur via relheight et relwidth toujours avec un nombre compris entre 0.0 et 1.0.

7) Pour les informations de date et d'heure dans le projet, nous les avons obtenues grâce à la bibliothèque "time" et "datetime". Nous avons commencé à "prendre" les heures, les minutes, les secondes et la date en utilisant la fonction strftime() avec comme paramètres '%d/%m/%y', localtime() pour paramétrier le temps, comme jours/mois/années. Nous utilisons la fonction localtime() pour avoir accès à l'heure, les minutes et les secondes.

8) La partie qui nous a semblée la plus difficile du projet est la mise en commun du travail de chacun. C'est à dire mettre en commun tous les morceaux de code dans un seul programme et le faire fonctionner. Nous pouvons rajouter que la recherche et le traitement des bugs et des erreurs n'est pas toujours quelque chose d'aisé.

9) Nous avons rajouté quelques fonctionnalités (mentionné plus haut dans le rapport). Avec plus de temps, nous aurions pu faire en sorte que l'accès aux autres onglets soit impossible, tant que l'utilisateur ne s'est pas connecté et que sa connexion au compte ne soit pas validée. Nous aurions aussi pu embellir l'interface graphique pour la rendre visuellement plus attractive. Nous aurions pu ajouter un générateur de mots de passe automatique où, lorsque l'utilisateur clique sur le bouton "générer un nouveau mot de passe", une nouvelle fenêtre s'ouvre. Ceci afin qu'il puisse choisir le nombre de caractères qu'il souhaite et que le générateur le produise automatiquement. Nous

aurions aussi pu ajouter un globe terrestre ou un planisphère pour montrer les avions et leurs trajectoires. L'ajout des droits et accès différents pour certains comptes utilisateurs serait également un plus. Ceci afin de donner accès à des utilisateurs privilégiés (type : administrateur...) pour qu'ils puissent utiliser des fonctionnalités supplémentaires.