

Rapport de Stage

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du
Group

Table des matières

Table des images	3
Introduction	4
- Présentation personnelle (page.4)	4
- Expérience professionnelle (page.4)	4
- Présentation de mon stage (page.4)	4
Présentation personnelle :	5
Expérience professionnelle :	5
Présentation de mon stage :	5
Présentation	6
Présentation rapide de l'entreprise	7
Activité réaliser au sein de l'entreprise	7
Résultat annuel de 2021	9
Objectifs 2022 :	9
Organigramme :	10
Présentation de mon tuteur :	10
Problématique :	12
Résolution de la Problématique :	12
Etape 1 :	12
Etape 2 :	13
Etape 3 :	14
Etape 4 :	15
Script python en entier :	18
Résultat du programme :	25
Conclusion :	26
Annexe :	29
Algorigramme :	32

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du Group

Table des images

Photo du site Thales SIX de Gennevilliers	9
Organigramme	10
Screen étape 1	12
Screen étape 1	12
Screen étape 2	13
Screen étape 2	13
Screen étape 2	14
Screen étape 3	14
Screen étape 3	15
Etape 4	16
Screen étape 4	16
Screen étape 4	16
Screen étape 4	17
Tableau de bord 1	33
Tableau de bord 2	34
Tableau de bord 3	35
Tableau de bord 4	35

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du
Group

Présentation personnelle :

Bonjour je m'appelle Enzo Gravelier j'ai 17 ans, j'étudie actuellement au Lycée Saint Paul Bourdon Blanc en classe de première système numérique option RISC et plus tard j'aimerais travailler dans le cyber sécurité soit travailler dans l'armée de l'air en tant que technicien des systèmes embarqués.

Expérience professionnelle :

J'ai déjà effectué 5 stages, 2 semaines au garage Renault B2A à Argenteuil, 2 semaines à Tereos à Artenay, 1 semaine au restaurant des plantes à Orléans, 1 semaine à la Boulangerie Pâtisserie des Roseraies et 6 semaines chez E.G.E.A 45 à Bucy-le-Roi.

Présentation de mon stage :

Aujourd'hui je vais vous présenter mon 6^{ème} stage que j'ai effectué à Thales SIX qui se situe à Gennevilliers sous la tutelle de Benjamin. Ce stage a eu pour durée 8 semaines du 07/03 au 06/05/2022. (Vacances scolaires comprises) Durant ces 8 semaines j'ai pu y apprendre la programmation.

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du
Group

Présentation

- Présentation rapide de l'entreprise_(page.6)
- Activité réaliser au sein de l'entreprise _(page.6)
- Résultat annuel de l'entreprise_(page.7)
- Organisation/Organigramme_(page.8)
- Présentation des différents acteurs et en particulier du tuteur_(page.8)

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du Group

Présentation rapide de l'entreprise

Thales est une entreprise qui a pour statut juridique SAS, c'est un leader mondial des hautes technologies comptant plus de 81 000 collaborateurs présents sur tous les continents. Le Groupe investit dans les innovations du numérique et de la « deep tech » – big data, intelligence artificielle, connectivité, cybersécurité et quantique – pour construire un avenir de confiance, essentiel au développement de nos sociétés, en plaçant l'humain au cœur des décisions.

Thales propose des solutions, services et produits qui aident ses clients – entreprises, organisations, Etats – dans cinq grands marchés vitaux pour le fonctionnement de nos sociétés : identité et sécurité numériques, défense, aéronautique, espace, et transport

Activité réaliser au sein de l'entreprise

Différente formation python, programmation python, réorganisation de la salle technique, apport de nouvelle machine dans la salle technique, création d'une nouvelle baie et déplacement de baie d'un autre bâtiment

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du Group

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du
Group

Résultat annuel de 2021

- Prises de commandes¹ : 19,9 Mds€, en hausse de 18% (+18% en variation organique²)
- Chiffre d'affaires : 16,2 Mds€, en hausse de 5,3% (+5,3% en variation organique)
- EBIT³ : 1 649 M€, en hausse de 32,1% (+31,9% en variation organique)
- Résultat net ajusté, part du Groupe³ : 1 361 M€, en hausse de 45%
- Free cash-flow opérationnel³ : 2 515 M€, 185% du résultat net ajusté, part du Groupe
- Dividende de 2,56 €, en hausse de 45%

Objectifs 2022 :

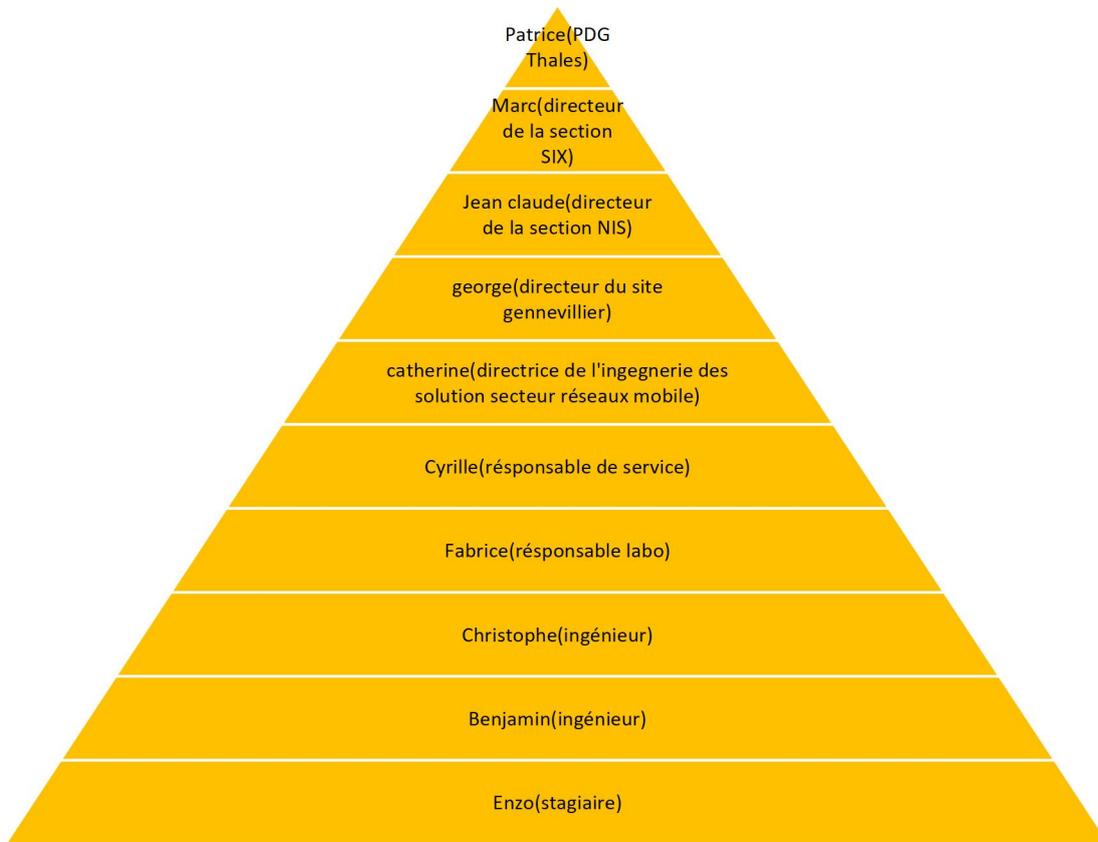
- Chiffre d'affaires compris entre 16,6 et 17,2 Mds€

Photo du site Thales SIX de Gennevilliers



Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du Group

Organigramme :



Organigramme

Présentation de mon tuteur :

Mon tuteur s'appelle Benjamin, sa fait pratiquement 2 ans et 9 mois qu'il travaille chez Thales group en tant que chef d'équipe et Responsable de l'ingénierie système mais avant cela il a travaillé 3ans et 5 mois chez idemia en tant qu'ingénieur système, 2 ans et 3 mois chez Vinci technologie en tant qu'ingénieur électronique et manager puis il a travaillé 4 ans et 1 mois chez E2_CAD en tant qu'ingénieur d'étude.

Il a fait comme étude un BTS en électronique au lycée Diderot puis une école d'ingénieur en électronique à l'ensea .

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du Group

Etude de cas :

- **Problématique (page.10)**
- **Résolution de la problématique (page.10)**
- **1^{ère} étape (page.10-11)**
- **2^{ème} étape (page.12)**
- **3^{ème} étape (page.12-13)**
- **4^{ème} étape (page.13-15)**
- **Script python en entier (page.15-22)**
- **Résultat du programme (page.23)**

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du
Group

Problématique : un collègue a besoin de générer un graphique à partir d'un fichier csv mais il ne possède pas Excel sur son poste.

Résolution de la Problématique :

Pour l'aider j'ai codé un programme en python qui permet de générer un graphique à partir d'un fichier csv depuis une IHM (interface humain-machine). Et pour se faire j'ai dû faire des formations sur python, j'ai dû regarder de quels modules j'ai besoin pour mon programme et j'ai dû m'informer de comment fonctionne ces modules.

Avant de commencer j'ai réfléchi a comment j'allais construire mon code à l'aide d'un algorithme. Voir annexe 2 p.

Etape 1 :

J'ai commencé mon programme par importer les modules dont j'ai besoin. Donc dans ce programme j'ai eu besoin que de 2 modules « Tkinter » et « matplotlib ». Le modules «Tkinter» sert à créer une IHM et «matplotlib» sert à générer un graphique

```
"""
On importe des modules qui nous serviras a créé une IHM et de générer un graphique
"""
import tkinter as tk #on rename Tkinter en tk pour simplifier
from tkinter import filedialog as fd#on importe des sous classe de tkinter
from tkinter import ttk#on importe des sous classe de tkinter
from tkinter import *#on importe des sous classe de tkinter
from matplotlib import pyplot as plt#on importe matplotlib puis une sous clase de matplotlib que l'on rename.
```

Screen étape 1

Ensuite j'ai créé une classe qui va nous permettre de partager les données entre les méthodes.

```
"""
On crée une class pour y crée des méthodes qui vont nous permettre de stocker des valeur dans l'IHM et de pouvoir
générer une graphique a partir des données stocké dans l'IHM
"""
class generation_graph:
```

Screen étape 1

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du Group

Etape 2 :

Puis je créer la première méthode qui contient le constructeur « `__init__` » et qui a pour paramètre « `self` » ce qui va nous permettre de partager les variable et les données. Dans cette première méthode je vais initialisez des variables dont j'en aurais besoins plus tard.

```

"""
on crée une méthode avec le constructeur __init__ ou on va venir créer des variables que l'on va réutiliser dans d'autres méthodes
et on va ouvrir une IHM qui va nous permettre d'utiliser les méthodes grâce à des boutons
"""
def __init__(self):
    self.element_Abscisse= []
    self.element_ordonnee= []
    self.y=[]
    self.x=[]
    self.nom_des_colones1= []
    self.nom_des_colones2= []

```

Screen étape 2

Dans cette même méthode je vais créer mon IHM, pour commencer je vais affecter ma fenêtre à une variables si qui va me permettre de les réutiliser au moment de la création des boutons pour pouvoir avoir tous les boutons et labels sur la même fenêtre, puis je vais donner un titre à ma fenêtre et ensuite je vais définir la taille de ma fenêtre. Après je créé tous les boutons et labels dont je vais avoir besoin. Après avoir créé tous mes Boutons et labels je vais tous les cachez grâce à la commande `forget` sauf le premier bouton.

```

self.root = Tk()
self.root.title("génération de graphique ")
self.root.geometry('400x400+550+200')

#errmsg = 'Error!'
self.bouton1 = Button(self.root, text='Click to Open File', command=self.open_text_file,font=("taille",10))
self.bouton1.pack(fill=tk.X)

self.label1 = Label(self.root, text="éléments Abscisses ", bg="grey",font=("taille",10))
self.label1.forget

self.bouton2 = ttk.Combobox(self.root, values=self.nom_des_colones1,font=("taille",10))
self.bouton2.forget

self.label2 = Label(self.root, text="éléments ordonnée ", bg="grey",font=("taille",10))
self.label2.forget

self.bouton3 = ttk.Combobox(self.root, values=self.nom_des_colones2,font=("taille",10))
self.bouton3.forget

self.bouton5 = Button(self.root, text="générer le graphique", command=self.graphique,font=("taille",10))
self.bouton5.forget

self.label_error = Label(self.root, text="aucun fichier sélectionner ou fichier invalide", fg="orange", bg="black",
font=("taille",15))
self.label_error.forget

self.button_destroy = Button(self.root, text="quitter",command=self.quit,fg="dark orange",font=("taille",15))
self.button_destroy.forget

```

Screen étape 2

Ensuite je créer une nouvelle méthode qui va être associer au bouton « `Button_destroy` » qui va permettre de détruire la fenêtre tkinter se qui va arrêter le programme.

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du Group

```
def quit(self):  
    self.root.destroy()
```

Screen étape 2

Etape 3 :

Ensuite je vais créer une troisième méthode qui va être liée au premier bouton, qui va nous permettre de choisir un fichier et de l'ouvrir. Pour pouvoir récupérer la première ligne du fichier qui contient les noms des colonnes. Et pour éviter qu'il y ait une erreur au moment de l'ouverture du fichier j'ai fait un « Try », « except » ce qui permet d'exécuter la première partie du code donc le Try, et si python n'arrive pas à l'exécuter alors il exécute la deuxième partie du code donc le except. Donc dans mon cas si un fichier est ouvert alors on lit la première ligne qui contient les noms des colonnes pour les affecter au bouton 2 et 3, faire apparaître les autres boutons et on désactive le premier bouton. Si aucun fichier n'est ouvert alors on affiche un message d'erreur « aucun fichier sélectionné ou fichier invalide » et on fait apparaître que le dernier bouton qui permettra de quitter le programme

```
def open_text_file(self):  
    self.name = fd.askopenfilename()  
    try:  
        self.fichier = open(self.name, 'r')  
        print(self.name)  
        self.elements = list()  
        compteur = 0  
        while 1: # La condition est donc toujours True  
  
            ligne = self.fichier.readline() # On tente de lire la ligne  
  
            if not (ligne): # Si ligne n'existe pas, elle contient False  
  
                break # On sort de la boucle  
  
            else: # Sinon, c'est que ligne contient quelque chose  
  
                ligne.replace('\n', '') # On supprime le passage à la ligne finale de chaque ligne  
  
                if compteur == 0:  
                    self.elements = ligne.split(';') # On crée une liste contenant les valeurs de chaque ligne  
  
                compteur += 1  
                self.nom_des_colones1 = self.elements  
  
                self.nom_des_colones2 = self.elements  
  
        self.bouton2.config(values=self.nom_des_colones1)  
        self.bouton3.config(values=self.nom_des_colones1)
```

Screen étape 3

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du Group

```
self.fichier.close()
self.label1.pack(fill=tk.X)
self.bouton2.pack(fill=tk.X)
self.label2.pack(fill=tk.X)
self.bouton3.pack(fill=tk.X)
self.bouton5.pack(fill=tk.X)
self.label_error.forget()
self.bouton1.config(state=DISABLED)

except:
self.label_error.pack()
self.label1.forget()
self.label2.forget()
self.bouton2.forget()
self.bouton3.forget()
self.bouton5.forget()
```

Screen étape 3

Etape 4 :

Et pour finir on va créer une dernière méthode qui va être relié au bouton qui va servir à générer le graphique. Tout d'abord on vient désactiver le bouton lorsque l'on appuie dessus pour éviter de générer plusieurs graphiques sur la même fenêtre. Après avoir désactiver le bouton on va récupérer les valeurs que l'on a sélectionné dans les boutons 2 et 3 et avec ses valeurs on va pouvoir récupérer le numéro de leur emplacement dans le fichier ce qui va nous permettre de les retrouver. On réouvre le fichier mais cette fois on va lire l'intégralité du fichier que l'on va venir apporter dans des listes. Mais lors du transfert du fichier vers les listes on va mettre comme paramètre les numéros d'emplacement des noms des colonnes que l'on a sélectionné. Ce qui va nous permettre de transférer que les colonnes que l'on a sélectionnées. Après avoir récupérer les colonnes on va les convertir en floats que si se ne sont pas des dates, si ce sont des dates alors on l'es convertie pas. La convention va nous permettre d'avoir les valeurs dans l'ordre croissant au moment de la génération du graphique. Une fois convertie on va pouvoir générer le graphique puis lui mettre une légende. On va pouvoir aussi l'enregistrez automatiquement. Et voilà on a généré on graphique à partir d'un fichier csv. Après avoir fermer le graphique on réactive les bouton 1 et 4 et on efface les listes pour pouvoir refaire des graphiques mais avec des valeurs différentes

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du Group

```
def graphique(self): # Fonction qui permet de générer le graphique
    self.bouton4.config(state=DISABLED)
    """
    on met les données que l'on a sélectionner dans les boutons 2 et 3 dans deux nouvelles variable pour éviter de
    modifier les données des boutons.
    """
    self.element_Abscisse = (self.bouton2.get())
    self.element_ordonnee = (self.bouton3.get())
    """
    On recupere le numero de l'emplacement des colonnes selectionne pour pouvoir les retrouver dans le fichier
    """
    element_graph_1 = self.nom_des_colones1.index(self.element_Abscisse)
    element_graph_2 = self.nom_des_colones2.index(self.element_ordonnee)
    """
    On réouvre le fichier pour cette fois lire tout le fichier
    """
    self.fichier = open(self.name, 'r')
    compteur = 0
    while 1: # La condition est donc toujours True

        ligne = self.fichier.readline() # On tente de lire la ligne

        if not (ligne): # Si ligne n'existe pas, elle contient False

            break # On sort de la boucle

        else: # Sinon, c'est que ligne contient quelque chose

            ligne.replace('\n', '') # On supprime le passage à la ligne finale de chaque ligne
```

Screen étape 4

```
plt.xlabel(self.element_Abscisse)
plt.ylabel(self.element_ordonnee)

self.Nom_fichier = self.name.replace(".csv", "") + " " + self.element_ordonnee + " " + self.element_Abscisse + \
    ".jpg"
plt.savefig(self.Nom_fichier) # on sauvegarde le graph en lui mettant comme nom la variable

plt.show() # on ferme la fenêtre
self.bouton1.config(state=NORMAL)
self.bouton4.config(state=NORMAL)
self.y.clear() # on efface la liste
self.x.clear() # on efface la liste
```

Screen étape 4

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du Group

```
ligne.replace('\n', '') # On supprime le passage à la ligne finale de chaque ligne
self.elements = ligne.split(';') # On crée une liste contenant les valeurs de chaque ligne

"""
On ajoute les colonnes que l'on a sélectionné au liste x et y si le compteur et différents de 0
"""
if compteur != 0:
    self.y.append(self.elements[element_graph_1])
    self.x.append(self.elements[element_graph_2])

compteur += 1

"""
Sa va nous permettre de convertir les chaînes de caractère en str sauf les dates
"""
if self.element_Abscisse != 'DATE':
    self.y = list(map(float, self.y))

if self.element_ordonnee != 'DATE':
    self.x = list(map(float, self.x))

self.fichier.close() # on ferme le fichier

"""
On génère le graphiaue a partir des lis x et y
"""
plt.plot(self.y, self.x) # on génère un graphique à partir des listes
"""
On vient créé une légende avec les variable element_Abscisse et element_ordonne
"""
```

Screen étape 4

Grace a ce travaille j'ai pu apprendre à programmer en python et sa ma aussi permit d'apprendre le fonctionnement de python. Voir programme complet annexe

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du Group

Script python en entier :

```
"""
On importe des modules qui nous servirons à créer une IHM et de générer un
graphique
"""
#import sys
import tkinter as tk
from tkinter import filedialog as fd
from tkinter import ttk
from tkinter import *

from matplotlib import pyplot as plt

"""
On instancie une class pour y créé des méthodes qui vont nous permettre de
stocker des valeur dans l'IHM et de pouvoir
générer une graphique a partir des données stocké dans l'IHM
"""

class generation_graph:
    """
    On crée une méthode avec le constructeur __init__ ou on va venir créer des
variables que l'on va réutiliser dans d'autres méthodes
    Et on va ouvrir une IHM qui va nous permettre d'utiliser les méthodes
grâce à des boutons
    """

    def __init__(self):
        self.element_Abscisse = []
        self.element_ordonnee = []
        self.y = []
        self.x = []
        self.nom_des_colones1 = []
        self.nom_des_colones2 = []

        self.root = Tk()
        self.root.title("génération de graphique ")
        self.root.geometry('400x200+550+200')

        #Le bouton 1 va nous permettre d'exécuter la méthode 'open_text_file'

        self.bouton1 = Button(self.root, text='Click to Open File',
activebackground="red",command=self.open_text_file,font=("taille",12))
        self.bouton1.pack(fill=tk.X)
```

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du
Group

```
#Le label 1 nous permet de savoir a quoi correspond le bouton 2

self.label1 = Label(self.root, text="éléments Abscisses ",
bg="grey",font=("taille",12))
self.label1.forget

#Le bouton 2 sert a selectionne une colone que l'on utilisera dans la
methode 'graphique'

self.bouton2 = ttk.Combobox(self.root,
values=self.nom_des_colones1,font=("taille",12))
self.bouton2.forget

#Le label 2 nous permet de savoir a quoi correspond le bouton 3

self.label2 = Label(self.root, text="éléments ordonnée ",
bg="grey",font=("taille",12))
self.label2.forget

# Le bouton 3 sert a selectionne une colone que l'on utilisera dans la
methode 'graphique'

self.bouton3 = ttk.Combobox(self.root,
values=self.nom_des_colones2 ,font=("taille",12))
self.bouton3.forget

#Le bouton 5 va nous permettre d'executer la methode 'graphique'

self.bouton4 = Button(self.root, text="générer le graphique",
command=self.graphique,activebackground="red",font=("taille",12))
self.bouton4.forget

self.label_error = Label(self.root, text="aucun fichier sélectionner
ou fichier invalide", fg="orange",bg="black",font=("taille", 15))
self.label_error.forget

#Le bouton_destroy permet d'executer la methode 'quit'

self.button_destroy =
Button(self.root,text="quitter",command=self.quit,fg="red",bg='light
grey',width=36,activebackground="red",font=("taille",15))
self.button_destroy.pack(side='bottom')

"""
```

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du
Group

Une methode qui permet de détruire la fenetre tkinter , se qui permet d'arreter le programme

```
"""
```

```
def quit(self):  
    #sys.exit()  
    self.root.destroy()
```

```
"""
```

On crée une méthode qui a pour nom open_text_file et qui va nous permettre d'aller chercher un fichier dans

l'explorateur du fichier.

après avoir sélectionner un fichier on ouvre le fichier puis on créer une boucle while

qui va nous permettre de récupérer la premiere ligne qui contient les noms des colones pour les stockés dans une variable.

et pour pouvoir les mettre dans les Combobox que l'on à créer dans la méthode précédente .

Une fois que la boucle while est fini on ferme le fichier et on fait apparaitre les bouton que l'on a créer dans la méthode

précédente et si aucun fichier n'a était sélectionné alors on affiche un message d'erreur.

```
"""
```

```
def open_text_file(self):  
    self.name = fd.askopenfilename()#on ouvre l'explorateur de fichier et  
on selctionne un fichier  
    try: #on essaie d'executer le code mais si on y arrive pas on affiche  
un message d'erreur et on fait disparaître les boutons  
        self.fichier = open(self.name, 'r')#on ouvre le fichier  
selectionner  
        print(self.name)#on ecrit se que contient la variable donc le  
chemin du fichier selectionner  
        self.elements = list()  
        compteur = 0  
        while 1: # La condition est donc toujours True  
  
            ligne = self.fichier.readline() # On tente de lire la ligne  
  
            if not (ligne): # Si ligne n'existe pas, elle contient False  
  
                break # On sort de la boucle
```

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du Group

```
else: # Sinon, c'est que ligne contient quelque chose

    ligne.replace('\n', '') # On supprime le passage à la
ligne finale de chaque ligne

    if compteur == 0:
        self.elements = ligne.split(';') # On crée une liste
contenant les valeurs de chaque ligne

        compteur += 1
"""
On vient mettre les données de éléments au bouton 2 et 3
"""

self.nom_des_colones1 = self.elements
self.nom_des_colones2 = self.elements
self.bouton2.config(values=self.nom_des_colones1)
self.bouton3.config(values=self.nom_des_colones2)

self.fichier.close()#on ferme le fichier
"""
On vient afficher tout les boutons et on desactive le premier
bouton pour éviter de cliquer dessus
"""
self.label1.pack(fill=tk.X)
self.bouton2.pack(fill=tk.X)
self.label2.pack(fill=tk.X)
self.bouton3.pack(fill=tk.X)
self.bouton4.pack(fill=tk.X)
self.label_error.forget()
except:
    "On affiche le message d'erreur et on cache tout les boutons sauf
le 1"

self.label1.forget()
self.label2.forget()
self.bouton2.forget()
self.bouton3.forget()
self.bouton4.forget()
self.label_error.pack()
print("ahahahahahaha")

"""
Je créé nouvelle méthode et qui pour nom 'graphique' et le but de celui-
ci est de générer un graphique à partir des
```

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du
Group

colonnes que nous avons sélectionnées et pour cela nous commençons en récupérant les noms des colonnes que nous avons sélectionnées.pour être en mesure de récupérer leur numéro de localisation par la suite.

Ensuite, nous allons créer une variable 'Nom_de_fichier' qui sera utilisé lors de l'enregistrement du graphique.

puis nous créerons une variable.element_graph_1 et element_graph_2 qui contiendront le numéro d'emplacement

de colonne du fichier.pouvoir extraire toutes les données contenues dans les colones.

Puis nous ouvrons le fichier que nous avons sélectionné dans la méthode précédente et puis nous créons une boucle

while qui va nous permettre de lire tout le fichier ligne par ligne et des les stockés dans une variable 'éléments'.

Après avoir stocké toutes les données du fichier dans une variable,nous allons venir stocker les données d'une colonnes

que nous avons sélectionné dans une liste x et.on fait pareil pour l'autres colones mais cette fois on le stocke dans une liste y .

Ensuite on va venir convertir la liste x en floats si les données sont différentes d'une date sinon

on ne convertie pas la liste x et on fait pareil pour la liste y

ce qui nous permettra d'avoir les nombres dans l'ordre dans le graphique.

Après avoir fait sa nous allons fermer le fichier et nous allons générer le graphique a partir des liste y et x.

nous plaçons une légende sur le graphe grâce aux variables "elements_Abscisse" et "elements_ordonnee".

Enregistre le graphique que nous venons de générer qui aura comme nom la variable "nom_fichier".

Et pour finir on clear les liste x et y pour pouvoir refaire un graphique.

"""

```
def graphique(self): # Fonction qui permet de générer le graphique
    """
```

```
    on met les donnees que l'on a selectionner dans les boutons 2 et 3
    dans deux nouvelles variable pour eviter de
    modifier les donnees des boutons.
    """
```

```
    self.element_Abscisse = (self.bouton2.get())
```

```
    self.element_ordonnee = (self.bouton3.get())
```

```
    """
```

```
    On recupere le numero de l'emplacement des colones selctionne pour
    pouvoir les retrouver dans le fichier
    """
```

```
    element_graph_1 = self.nom_des_colones1.index(self.element_Abscisse)
```

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du
Group

```
element_graph_2 = self.nom_des_colones2.index(self.element_ordonnee)
"""
On réouvre le fichier pour cette fois lire tout le fichier
"""
self.fichier = open(self.name, 'r')
compteur = 0
while 1: # La condition est donc toujours True

    ligne = self.fichier.readline() # On tente de lire la ligne

    if not (ligne): # Si ligne n'existe pas, elle contient False

        break # On sort de la boucle

    else: # Sinon, c'est que ligne contient quelque chose

        ligne.replace('\n', '') # On supprime le passage à la ligne
        finale de chaque ligne
        self.elements = ligne.split(';') # On crée une liste
        contenant les valeurs de chaque ligne

        """
        On ajoute les colonnes que l'on a sélectionné au liste x et y
        si le compteur et différents de 0
        """
        if compteur != 0:
            self.y.append(self.elements[element_graph_1])
            self.x.append(self.elements[element_graph_2])

        compteur += 1

    """
    Sa va nous permettre de convertir les chaînes de caractère en str
    sauf les dates
    """
    if self.element_Abscisse != 'DATE':
        self.y = list(map(float, self.y))

    if self.element_ordonnee != 'DATE':
        self.x = list(map(float, self.x))

self.fichier.close() # on ferme le fichier

"""
On génère le graphique à partir des liste x et y
"""
```

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du
Group

```
"""
plt.plot(self.y, self.x) # on génère un graphique à partir des listes
"""
On vient créé une légende avec les variable element_Abscisse et
element_ordonne
"""
plt.xlabel(self.element_Abscisse)
plt.ylabel(self.element_ordonnee)

self.Nom_fichier = self.name.replace(".csv", "") + " " +
self.element_ordonnee + " " + self.element_Abscisse + \
    ".jpg"
plt.savefig(self.Nom_fichier) # on sauvegarde le graph en lui mettant
comme nom la variable

plt.show() # on ferme la fenêtre
self.y.clear() # on efface la liste
self.x.clear() # on efface la liste

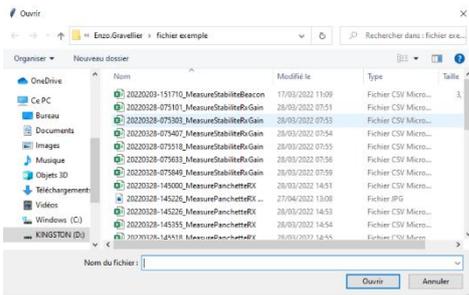
a = generation_graph() # On appel la class generation_graph
tk.mainloop() # On ferme l'IHM
```

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du
Group

Résultat du programme :



Etape 1 on lance le programme et on appuie sur le premier bouton



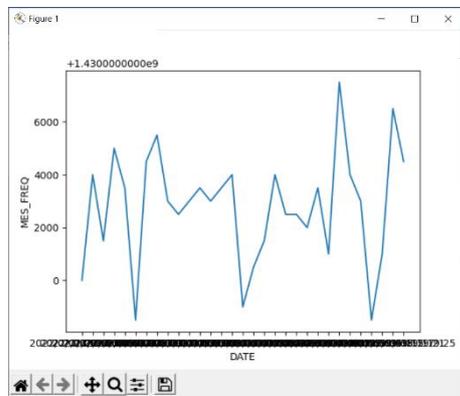
Etape 2 on sélectionne un fichier



Etape 3 : des boutons apparaissent



Etape 4: on sélectionne des éléments et on appui sur le bouton générer le graphique



Etape 5: le graphique apparait

Etape 6: On ferme le programme en appuyant sur le bouton quitter



Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du Group

Conclusion :

Rappel (page.25)

Compétence acquise durant le stage (page.25)

Relations humaines : (page.25)

Projet d'avenir (page.25)

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du
Group

Rappel :

Pour rappel j'effectue mon stage chez Thales six (system and information) à Gennevilliers ou j'y fait de la programmation python orienter objet et du câblage sur des baie.

Compétence acquise durant le stage :

Durant ce stage j'ai pu apprendre a programmée du python et comment fonctionne python.

Relations Humaines :

Je me suis bien intégré à l'équipe, je m'entends bien avec tous les membre de l'équipe, ils ont bienveillant et patient à mon égard.

Projet d'avenir :

C'est un métier que je pourrais envisagez à l'avenir, mais pour le coup se stage ne ma pas éclairer sur le métier que j'aimerais faire plus tard, sa à juste rajouter un métier dans la liste

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du
Group

Remerciements :

Je tiens à remercier vivement Thales SIX de m'avoir accueilli pour effectuer mon stage, en m'offrant ainsi la possibilité d'acquérir une expérience professionnelle très enrichissante.

Je remercie ensuite tous les membres de l'équipe du labo démo pour leur soutien et leur attention dont ils ont fait preuve à mon égard tout au long du déroulement de ce stage.

Enfin, j'adresse mes remerciements aussi à mes collègues et amis stagiaires, qui ont montré durant ce stage une amabilité et une forte compréhension.

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du
Group

Annexe :

Les différents services de Thales (page.27-28)

Algorithme (page.29)

Tableau de bord (page.30)

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du
Group

Annexe :

Qu'elles sont les services de Thales :

Thales est divisé en 6 service bien distinct, qui sont plus communément appelé « business unit »

LAS = Land and Air Systems:

Thales LAS développe des munitions, de l'armement, la cryogénie, les radars, des objectifs photographique et cinématographique, la désignation laser, de là l'optronique et les autodirecteurs radar de missile

TAS = Aliena space:

Thales TAS développe toutes sorte de satellite. Donc ils font des satellites de télécommunications, des constellations de satellites, des satellites de télédétection, des satellites scientifiques, des satellites militaires, des satellites de navigation et ils font aussi des grands systèmes à base de satellites. Mais ils ne font pas que ça. Ils font de la météorologie spatiale, de l'aéronautique avec leur projet de ballon stratosphérique : Stratobus et il participe aussi à l'ISS Avec trois modules : Harmony, Columbus et Tranquility

DIS = digital identify and security:

Activité de Thales DIS :

Les banques et les paiements,

Les solutions d'identité et biométrie,

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du Group

L'internet des objets – solution lot pour 2022 et au-delà,

Solution et service

DMS = defense mission system :

Thales DMS développe tout ce qui est en rapport avec l'aéronautique comme par exemple le radar qui se trouve à l'avant du rafale

AVS = avionics:

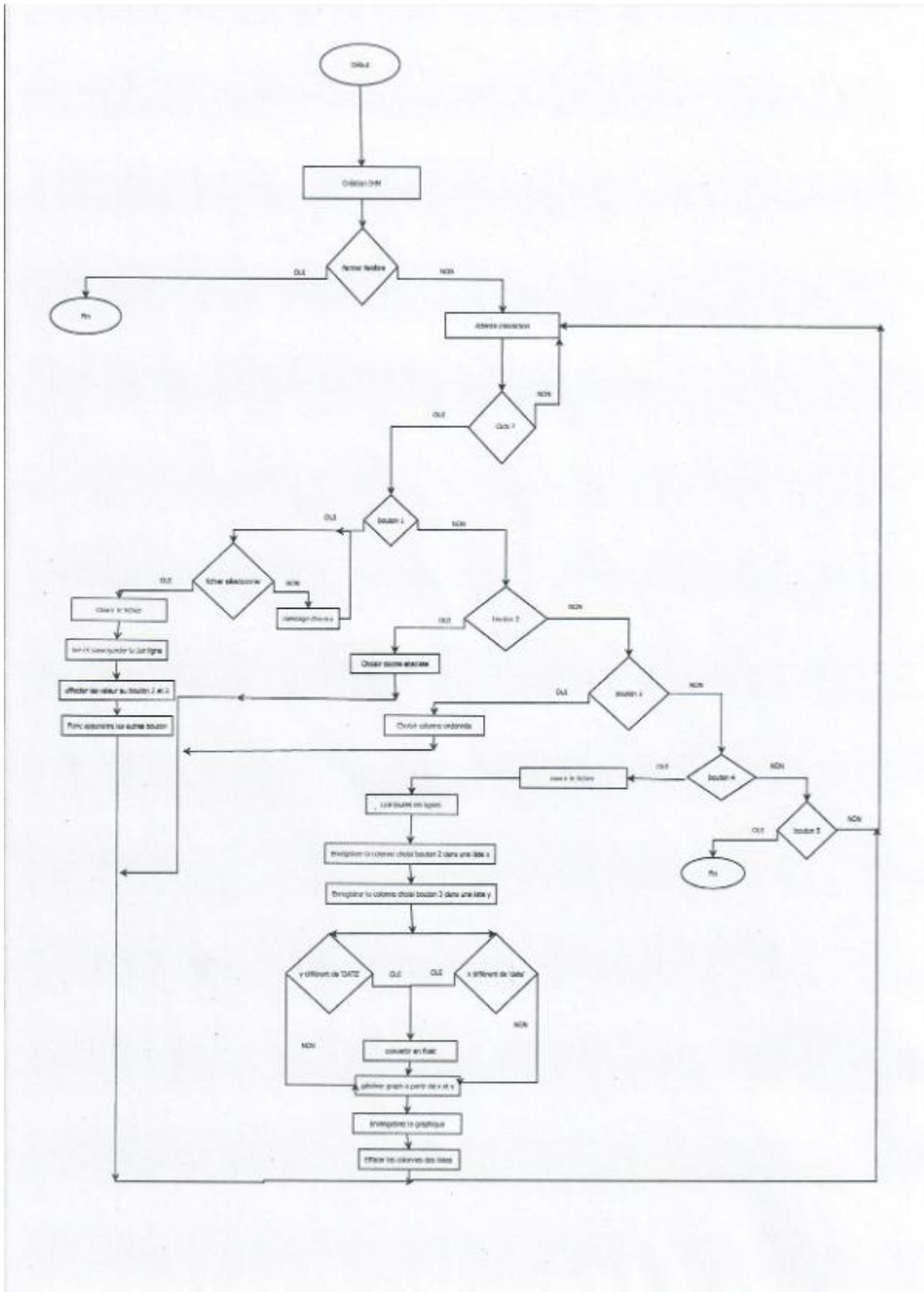
Thales AVS développe les systèmes qui permettent aux avions de voler en toutes circonstances. De la gestion du trafic aérien, des systèmes d'entraînement et de simulation aux services multimédias embarqués, en passant par la connectivité intégrale de l'avion, les solutions interfacent tous les éléments de l'écosystème aéronautique, au sol ou en vol

SIX = system and information:

Thales SIX développe la communication par satellite sécuriser dans tous types de situation que ce soit en déplacement ou immobile.

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du
Group

Algorithme :



Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du Group

Tableau de bord :

Journal de bord Stage 1er PSN

Du lundi 7 mars au vendredi 15 avril et du 25 Avril au 6 mai

Date	Activité réalisée	Ressources matérielles à disposition	Ressources logicielles à disposition	Personnes avec qui vous êtes en relation pour réaliser cette activité (autre salarié, supérieur hiérarchique, client, ...)	Résultat attendu	Difficultés rencontrées et moyens mis en œuvre pour les surmonter	Remarques éventuelles	Degré d'autonomie
07 March 2022	Télétravail: Formation Python	Ordinateur	pycharm et internet	personne	conaitre la boucle if, else, elif et la boucle for, savoir utiliser les classe et les dictionnaires	incompréhension de certain passage de la formatio n. Monntuteur ma aidée à mieux comprendre	La formation n'était pas très compréhensible	Autonome
08 March 2022	Télétravail: Formation Python	Ordinateur	pycharm et internet	personne	conaitre la boucle if, else, elif et la boucle for, savoir utiliser les classe et les dictionnaires	incompréhension de certain passage de la formatio n. Monntuteur ma aidée à mieux comprendre	La formation n'était pas très compréhensible	Autonome
09 March 2022	Modification de la Baie 1, changement de câble DVI-D en câble DVI-A, décablage d'une antenne satellite et étiquetage des différents câbles de la baie 1 et client	etiqueteuse, tournevis	aucun	personne	finir d'étiqueté tous les câble présent dans la baie 1	aucune	aucune	Autonome
10 March 2022	visite à un autre salarié qui s'occupe des antenne du site, commencement de la mise au propre de la baie 1 et cour sur les fréquence de mon tuteur	etiqueteuse, tournevis, serre-câble, serre-câble scratch, pince coupante, pince a bec et échelle	aucun	personne	aucun	aucune	aucune	Autonome
11 March 2022	fin de la mise au propre e la Baie 1	etiqueteuse, tournevis, serre-câble, serre-câble scratch, pince coupante, pince a bec et échelle	aucun	personne	câblage propre et pratique de la baie 1	aucune	aucune	Autonome
14 March 2022	photo pour le badge, commencement de la mise au propre des baie 2,3 et 4 puis sensibilisation	etiqueteuse, tournevis, serre-câble, serre-câble scratch, pince coupante, pince a bec et échelle	aucun	sensibilisation avec mon tuteur	avoir le badge	aucune	aucune	Spéctateur
15 March 2022	fin de la mise au propre des baie 2,3 et 4 puis réorganisation de la baie 4 et installation d'un kvm dans la baie 4	etiqueteuse, tournevis, serre-câble, serre-câble scratch, pince coupante, pince a bec et échelle	aucun	personne	fin de la mise au propre de toute les baie	aucune	aucune	Autonome
16 March 2022	installation de 7 serveur et 5 routeur dans la baie 4, branchement du kvm vers serveur avec câble dvi -A	tournevis, équerre métallique et câble dvi -A	aucun	personne	fixation de toute les nouvelles machine apporter à la baie 4	parfois les trou pour fixer les routeur ne tomber pas au bonne endroit donc j'ai surélévé les routeurs dont les trou ne tomber pas juste pour pouvoir fixer les deux vis du bas et ensuite j'ai mis des équerrres métallique pour soutenire l'arrière du routeur	aucune	Autonome

Tableau de bord 1

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du Group

17 March 2022	télétravail : programmation python mais au final rapport de stage	ordinateur	pycharm, fichier csv, excel et word	personne	créé un programme permettant d'afficher des courbes issues d'une liste de données	pour cette activité j'ai ramené l'ordinateur du travail pour pouvoir commencer le programme mais je n'ai pas pu installer tout ce qu'il fallait car il demandait un mot de passe administrateur, que je n'avais pas	aucune	Autonome
18 March 2022	Montage d'un support d'écran de télé, branchement des câbles d'alimentation sur toute les nouvelles machines de la baie 4 et commencement de la mise au propre de la baie de brassage	pince coupante, serre-câble, scratch et échelle	aucun	personne	retirer tous les câbles qui étaient branchés que d'un seul côté	aucune	aucune	Autonome
21 March 2022	enlèvement des câbles branchés que d'un seul côté, enlèvement d'un support de convertisseur fibre vers ethernet, câblage électrique et commencement de la mise au propre des caissons	pince coupante, tournevis, serre-câble, scratch, échelle et prise électrique	aucun	personne	aucun	aucune	aucune	Autonome
22 March 2022	mise au propre des caissons de démonstration	serre-câble, scratch et échelle	aucun	personne	fin de la mise au propre des caissons de démonstration	aucune	aucune	Autonome
23 March 2022	j'ai vidé la puissance de toutes les machines qui la composent pour les mettre dans une baie	tournevis, équerre métallique et pince coupante	aucun	personne	finir le travail	Il manquait 4 équerres donc je n'ai pas pu finir. Pour y remédier nous avons commandé de nouvelles équerres.	aucune	Autonome
24 March 2022	commencement de la programmation python. Lecture de fichier csv avec une génération de graphique à partir du fichier csv et pour simplifier tout ça une IHM (Interface Homme-machine)	Ordinateur	pycharm	personne	Faire la première partie c'est à dire la lecture du fichier csv	aucune	aucune	Autonome
25 March 2022	ajout d'une machine dans la baie 5 et programmation python	Ordinateur, tournevis et vis	pycharm	personne	terminer l'ouverture du fichier csv	aucune	aucune	Autonome
28 March 2022	Programmation python	ordinateur	pycharm	personne	création du générateur de graphique	aucune	aucune	Autonome
29 March 2022	Programmation python	ordinateur	pycharm	personne	création du premier bouton de l'ihm	aucune	aucune	Autonome
30 March 2022	Programmation python	ordinateur	pycharm	personne	création des 3 autres boutons	Les boutons ne se mettent pas sur la même interface	aucune	Autonome
31 March 2022	Programmation python	ordinateur	pycharm	personne	assignation des fonctions aux boutons	aucune	aucune	Autonome

Tableau de bord 2

Toute reproduction ou publication d'éléments du rapport doivent faire l'objet d'un accord écrit du Group

Tableau de bord 3

27 Avril 2022	commencer a recâbler la baie de brassage	scratch (serre-câble)	aucun	personne	faire le plus possible avec le peu de serre câble	plus de serre cable	aucune	Autonome	
28 Avril 2022	essai de la démo, nettoyage et rapport de stage	ordinateur et caisson de démo	eddi remote, word	personne	faire des teste	aucune	aucune	Autonome	
29 Avril 2022	fin du recablage de la baie de brassage	serre-câble scratch et échelle	aucun	personne	finir la baie de brassage	aucune	aucune	Autonome	
02 May 2022	circuit de départ	aucun	outlook	tuteur	préparer mon départ de l'entreprise	aucune	aucune	Aidé	
03 May 2022	circuit de départ	aucun	outlook	tuteur	préparer mon départ de l'entreprise	aucune	aucune	Aidé	
04 May 2022	Rapport de stage	aucun	Word	personne	Finir mon rapport de stage	aucune	aucune	Autonome	
05 May 2022	Rapport de stage	aucun	Word	personne	Finir mon rapport de stage	aucune	aucune	Autonome	
06 May 2022	circuit de départ et départ de l'entreprise	aucun	aucun	tuteur	aucun	aucune	aucune	aidé	
13 Avril 2022	Formation au modem et commencement du remplissage de baie will castle	tournevis , vis et pince a bec	aucun	tuteur	rien	aucune	aucune	Observateur et Autonome	
14 Avril 2022	fin du remplissage de la baie will castle puis commencement du câblage de la baie	tournevis , vis et pince a bec	aucun	personne	finir la baie will castle	aucune	aucune	Autonome	
15 Avril 2022	fin du câblage de la baie will castle et commenter mon code	ordinateur	pycharm	personne	finir entièrement la baie will castle	aucune	aucune	Autonome	
25 Avril 2022	suppression de passage inutile a mon code et ajout de nouvelle fonctionnalité	ordinateur	pycharm	personne	ne plus avoir de difficulté avec python	aucune	sa rentre vite dans la tête	Autonome	
26 Avril 2022	Rapport de stage, régler un problème sur mon code	ordinateur	pycharm, word	personne	continuer un peu mon rapport de stage	aucune	aucune	Autonome	