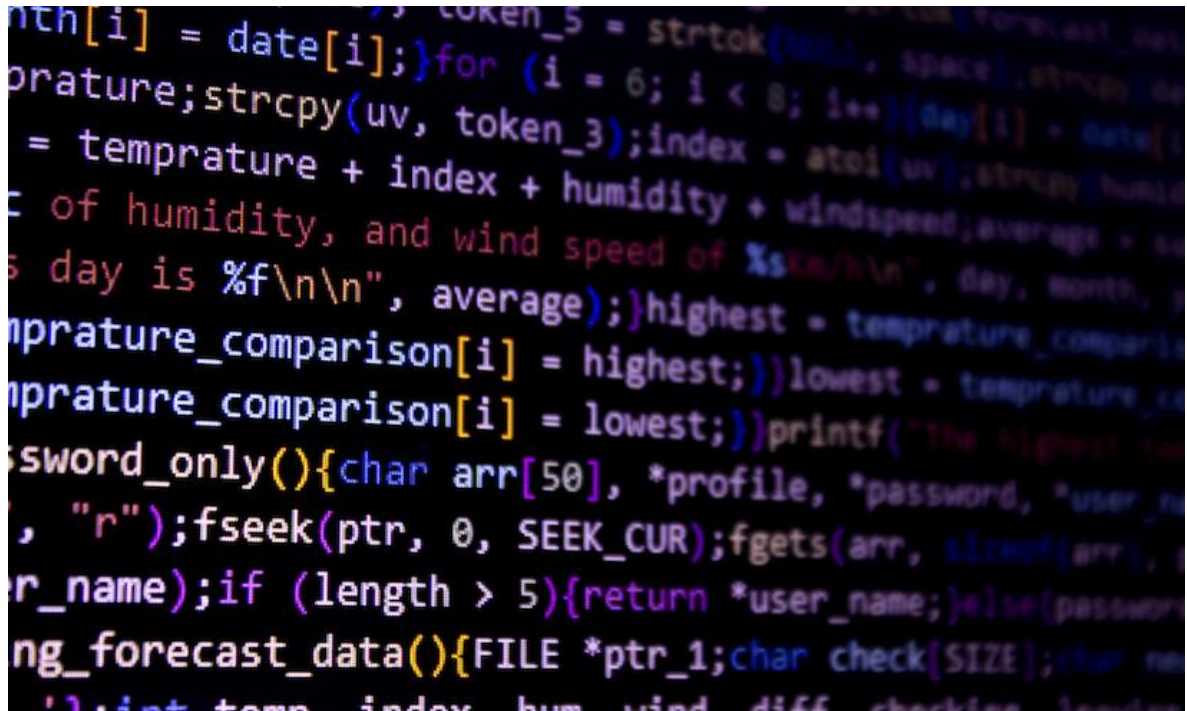


Compte rendu

- Projet n9 -



```
nth[i] = date[i];}for (i = 6; i < 8; i++){day[i] = date[i]
perature;strcpy(uv, token_3);index = atoi(uv);strcpy(humid
= temprature + index + humidity + windspeed;average = av
of humidity, and wind speed of %skm/h\n", day, month, )
s day is %f\n\n", average);}highest = temprature_compari
mprature_comparison[i] = highest;}lowest = temprature_co
mprature_comparison[i] = lowest;}printf("The highest tem
sword_only(){char arr[50], *profile, *password, *user_n
, "r");fseek(ptr, 0, SEEK_CUR);fgets(arr, sizeof(arr),
r_name);if (length > 5){return *user_name;}else{password
ng_forecast_data(){FILE *ptr_1;char check[SIZE];char ne
};int temp, index, hum, wind, diff, checkles, lowest
```

GENOT Mathys

JIOFO-DJIOMEZI Denhilson-Hyggor

Groupe : RT1 - Turing

1 Introduction

Le projet consiste à développer une solution permettant à l'enseignant d'ajouter une séance de cours dans un emploi du temps déjà bien chargé. L'objectif est d'afficher les 5 prochains créneaux de 2 heures disponibles pour n'importe quel groupe d'étudiants donné à partir d'une date spécifiée.

2 Cahier de décharge

Le cahier des charges fixe les paramètres suivants :

La date d'analyse est paramétrable, permettant à l'enseignant de spécifier la date à partir de laquelle les créneaux disponibles seront recherchés.

```
start_date_str = input("Veuillez entrer la date de début au format YYYY-MM-JJ : ")
start_date = datetime.strptime(start_date_str, "%Y-%m-%d").replace(tzinfo=None)
group = input("Veuillez entrer le groupe d'étudiants : ")
```

Le groupe d'étudiants est également paramétrable, offrant une flexibilité pour sélectionner le groupe concerné. Les jeudis après-midi sont exclus en tant que créneaux libres, respectant ainsi les contraintes de l'emploi du temps.

```
start_date_str = input("Veuillez entrer la date de début au format YYYY-MM-JJ : ")
start_date = datetime.strptime(start_date_str, "%Y-%m-%d").replace(tzinfo=None)
group = input("Veuillez entrer le groupe d'étudiants : ")
```

```
while len(emplacements_disponibles) < 5:
    if date_actuelle.weekday() in [5, 6]: # Samedi et Dimanche
        date_actuelle += timedelta(days=1)
        continue
    if date_actuelle.weekday() == 3 and date_actuelle.hour >= 12: # Jeudi après-midi
        date_actuelle += timedelta(days=1)
        continue
    if 8 <= date_actuelle.hour < 16: # Entre 8h et 18h
        end_of_slot = date_actuelle + timedelta(hours=2)
        if horaire_modifie[horaire_modifie['Début'] < end_of_slot & (horaire_modifie['Fin'] > date_actuelle)].empty:
            emplacements_disponibles.append((date_actuelle, end_of_slot))
```

Le résultat doit être présenté de deux manières :

Sous forme d'un tableau au format CSV, XLSX ou PDF, facilitant ainsi la consultation et l'utilisation des informations générées.

```
# Sauvegarde des créneaux disponibles dans un fichier PDF
creneaux_df = pd.DataFrame(creneaux, columns=['Début', 'Fin'])
pdf_filename = 'creneaux_disponibles.pdf'
pdf_pages = PdfPages(pdf_filename)

fig, ax = plt.subplots()
ax.axis('tight')
ax.axis('off')
table_data = [list(map(str, row)) for row in creneaux_df.values]
table = ax.table(cellText=table_data, colLabels=['Début', 'Fin'], cellLoc='center', loc='center')
table.auto_set_font_size(False)
table.set_fontsize(10)
table.auto_set_column_width([0, 1])

pdf_pages.savefig(fig, bbox_inches='tight')
pdf_pages.close()

plt.show()
```

Sous forme d'une frise chronologique au format libre (par exemple, PNG), fournissant une représentation visuelle des créneaux disponibles.

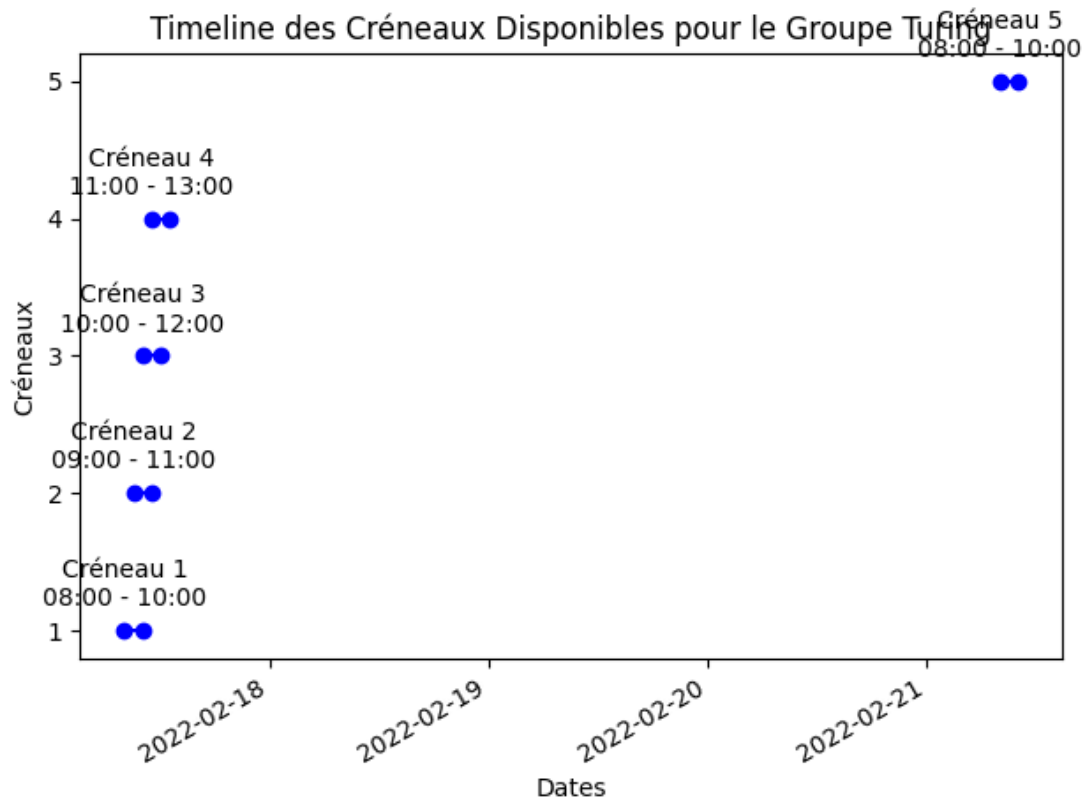
```
# Création de la frise chronologique
fig, ax = plt.subplots()
for i, creneaux in enumerate(creneaux, start=1):
    # Convertir les objets Timestamp en format matplotlib
    start_matplotlib = mdates.date2num(creneaux[0])
    end_matplotlib = mdates.date2num(creneaux[1])
    ax.plot([start_matplotlib, end_matplotlib], [i, i], marker='o', linestyle='-', color='b')
    ax.annotate(f"Créneau {i}\n{creneaux[0].strftime('%H:%M')} - {creneaux[1].strftime('%H:%M')}",
                (start_matplotlib, i), textcoords="offset points", xytext=(0,10), ha='center')

ax.xaxis.set_major_locator(mdates.DayLocator())
ax.xaxis.set_major_formatter(mdates.DateFormatter('%Y-%m-%d'))
ax.yaxis.set_major_locator(plt.MultipleLocator(1)) # Graduation de 1 en 1 sur l'axe des données

plt.title('Timeline des Créneaux Disponibles pour le Groupe ' + group)
plt.xlabel('Dates')
plt.ylabel('Créneaux')

fig.autofmt_xdate()
plt.tight_layout()

# Enregistrement de la frise chronologique sous forme de fichier PNG
plt.savefig('Diagramme.png', format='png')
```



(Un problème d'affichage est survenu sur la photo, mais sur les machines LINUX ce problème n'y apparaîtra pas.)

3 Méthode de développement

Le projet sera développé en plusieurs étapes :

Collecte des Données : Le système collectera les données pertinentes, notamment la liste des créneaux disponibles, en prenant en compte les contraintes spécifiées.

```
def emplacements_disponibles(schedule, start_date, group):
    groupe_schedule = schedule[schedule['Description'].str.contains(group, case=False, na=False)]
    start_date = pd.to_datetime(start_date).tz_localize(None)
    horaire_modifie = groupe_schedule[groupe_schedule['Début'] >= start_date]

    emplacements_disponibles = []
    date_actuelle = start_date

    while len(emplacements_disponibles) < 5:
        if date_actuelle.weekday() in [5, 6]: # Samedi et Dimanche
            date_actuelle += timedelta(days=1)
            continue
        if date_actuelle.weekday() == 3 and date_actuelle.hour >= 12: # Jeudi après-midi
            date_actuelle += timedelta(days=1)
            continue
        if 8 <= date_actuelle.hour < 16: # Entre 8h et 18h
            end_of_slot = date_actuelle + timedelta(hours=2)
            if horaire_modifie[(horaire_modifie['Début'] < end_of_slot) & (horaire_modifie['Fin'] > date_actuelle)].empty:
                emplacements_disponibles.append((date_actuelle, end_of_slot))

        date_actuelle += timedelta(hours=1)
        if date_actuelle.hour >= 18:
            date_actuelle = date_actuelle.replace(hour=8, minute=0, second=0, microsecond=0) + timedelta(days=1)

    return emplacements_disponibles
```

Ici, la recherche des jours exclue sont faite en même temps que la récupération des journées disponible.

Analyse et Traitement : Les créneaux disponibles seront analysés, les jeudis après-midi exclus, et les 5 prochains créneaux de 2 heures seront identifiés.

```
while len(emplacements_disponibles) < 5:
    if date_actuelle.weekday() in [5, 6]: # Samedi et Dimanche
        date_actuelle += timedelta(days=1)
        continue
    if date_actuelle.weekday() == 3 and date_actuelle.hour >= 12: # Jeudi après-midi
        date_actuelle += timedelta(days=1)
        continue
    if 8 <= date_actuelle.hour < 16: # Entre 8h et 18h
        end_of_slot = date_actuelle + timedelta(hours=2)
        if horaire_modifie[(horaire_modifie['Début'] < end_of_slot) & (horaire_modifie['Fin'] > date_actuelle)].empty:
            emplacements_disponibles.append((date_actuelle, end_of_slot))
```

Présentation des Résultats : Les résultats seront présentés sous forme de tableau dans les formats demandés (CSV, XLSX, PDF) pour une accessibilité facile. Une frise chronologique sera générée pour une représentation visuelle.

Début	Fin
2022-02-17T08:00:00.000000000	2022-02-17T10:00:00.000000000
2022-02-17T09:00:00.000000000	2022-02-17T11:00:00.000000000
2022-02-17T10:00:00.000000000	2022-02-17T12:00:00.000000000
2022-02-17T11:00:00.000000000	2022-02-17T13:00:00.000000000
2022-02-21T08:00:00.000000000	2022-02-21T10:00:00.000000000

Résultat sous format .PDF.

4 Résultat attendue

Les résultats attendus seront clairs, précis et faciles à interpréter. L'enseignant pourra rapidement identifier les créneaux disponibles et décider de l'ajout de la séance de cours. Avec le diagramme et le fichier .PDF, la tâche sera facilement faite.

5 Perspectives d'amélioration

Des fonctionnalités pourront être plus autonome, dans le programme python.



```
def parse_date(date_str):  
    token_5 = strtok(date_str, space).strncpy(  
    date[i] = date[i];  
    for (i = 6; i < 8; i++){  
        day[i] = date[i];  
    }  
    temperature; strcpy(uv, token_3); index = atoi(uv); strcpy(humidit  
    = temperature + index + humidity + windspeed; average = av  
    of humidity, and wind speed of %s km/h\n", day, month, y  
    s day is %f\n\n", average); } highest = temperature_compari  
    nprature_comparison[i] = highest; } lowest = temperature_co  
    nprature_comparison[i] = lowest; } printf("The highest tem  
    sward_only(){ char arr[50], *profile, *password, *user_n  
    , "r"); fseek(ptr, 0, SEEK_CUR); fgets(arr, sizeof(arr), f  
    r_name); if (length > 5){ return *user_name; } else{ passwor  
    ng_forecast_data(){ FILE *ptr_1; char check[SIZE]; char na  
    '1'; int temp, index, hum, wind, diff, checking, lowden
```

Les utilisateurs n'auront plus besoin de changer le chemin du dossier pour rendre le programme en fonctionnalité.

6 Conclusion :

Le développement de cette solution devrait faciliter le processus d'ajout de séances de cours, offrant une flexibilité accrue à l'enseignant tout en respectant les contraintes de l'emploi du temps. La présentation sous forme de tableau et de frise chronologique ajoute une dimension visuelle à la facilité d'utilisation de la solution.