

**PICTU**

## **Rapport n°1**

Par *JAST*:

Sarah Santiago

Jimmy Fabre

Thuy-Trang Nguyen

Alban Richard

---

---

# **Sommaire :**

## **1. Introduction**

1.1 Pictu.....

## **2. Répartition des tâches**

## **3. Avancement du Projet**

3.1 Algorithme.....

3.2 Communication.....

3.3 Données.....

3.4 Interface.....

3.5 Réseau.....

3.6 Site Web.....

3.7 Installateur / Lanceur.....

## **4. Nos impressions**

## **5. Prévisions pour la prochaine soutenance**

## **6. Conclusion**

---

## 1/Introduction :

Dans le cadre de notre projet de quatrième semestre, nous avons choisi de réaliser une application capable de proposer un contenu, ici des images, aux utilisateurs de cette dernière.

Ceci est notre premier rapport de soutenance. Nous allons détailler ici nos progrès et difficultés concernant le développement de notre projet, à savoir notre application “Pictu”. Nous aborderons tous les aspects de nos avancées: recherches, idées, algorithmes, codes ou livrables que nous avons en l’état actuel.

### 1.1 Pictu:

Pictu est notre application qui permet à l'utilisateur de consommer du contenu, comme des photos ou encore des vidéos, qui sont recommandés par rapport à ses goûts. En effet, il lui sera possible de se divertir en regardant des choses qu'il apprécie, et pourra aimer ce contenu, laisser des commentaires afin d'avoir un contenu similaire par la suite. Le principe algorithmique de ce projet est tout l'algorithme de recommandation qui va prendre en compte les attirances de l'utilisateur mais va aussi les comparer aux autres utilisateurs pour essayer de lui proposer du contenu inédit qui réponde à ses attentes.

---

## 2/Répartition des tâches :

Voici un rappel de la répartition des tâches de notre groupe:

Répartition des tâches	Sarah Santiago	Jimmy Fabre	Thuy-Trang Nguyen	Alban Richard
Algorithme		❖	⊗	
Communication		⊗	❖	
Données	⊗			❖
Interface	❖			⊗
Réseaux	❖	⊗		
Site Web	⊗			❖
Installateur/ Lanceur			❖	⊗

Legende : ⊗ = Responsable ; ❖ = Suppleant

Pour la première soutenance, chaque responsable a travaillé en solitaire sur sa catégorie:

- ★ Thuy-Trang Nguyen sur l'**Algorithme de recommandation**
- ★ Alban Richard sur l'**Interface de l'application**
- ★ Sarah Santiago sur le **Site web**
- ★ Jimmy Fabre sur le **Serveur de l'application**

---

### 3/Avancement du projet :

Lors de ces derniers mois, nous avons fait des progrès dans de multiples domaines concernant le projet, que nous allons présenter ici en les regroupant par domaine dans un souci de clarté, en les listant du plus concret au plus théorique. Nous avons aussi effectué des travaux de recherche dans les différents domaines du projet afin d'étayer notre progression.

#### 3.1 Algorithme:

Pour mettre en œuvre ce projet, nous utilisons l'algorithme de filtrage collaboratif. Le filtrage collaboratif est la technique la plus couramment utilisée lorsqu'il s'agit de créer des systèmes de recommandation intelligents qui peuvent apprendre à donner de meilleures recommandations à mesure que davantage d'informations sur les utilisateurs sont collectées.

La plupart des sites Web comme Amazon, YouTube et Netflix utilisent le filtrage collaboratif dans le cadre de leurs systèmes de recommandation sophistiqués. Vous pouvez utiliser cette technique pour créer des recommandations qui donnent des suggestions à un utilisateur sur la base des goûts et des aversions d'utilisateurs similaires.

##### a. Qu'est-ce que le filtrage collaboratif ?

Le filtrage collaboratif est une technique qui peut filtrer les éléments qu'un utilisateur pourrait aimer sur la base des réactions

---

d'utilisateurs similaires. Cela fonctionne en recherchant un grand groupe de personnes et en trouvant un plus petit ensemble d'utilisateurs ayant des goûts similaires à un utilisateur particulier. Il examine les éléments qu'ils aiment et les combine pour créer une liste classée de suggestions.

b. L'ensemble de données

Pour tester les algorithmes de recommandation, nous aurons besoin de données contenant un ensemble d'éléments et un ensemble d'utilisateurs ayant réagi à certains des éléments. La réaction peut être explicite (notation sur une échelle de 1 à 5, aime ou n'aime pas) ou implicite (visionner un article, l'ajouter à une liste de souhaits, le temps passé sur un article).

Nous collectons ici un ensemble de données des avis de 548 clients sur 9125 articles.

---

### c. Étapes impliquées dans le filtrage collaboratif

- *Création d'une matrice utilitaire de notes entre les utilisateurs et les articles*

```
void get_utility_matrix(double *utility_matrix, char *s, int No_of_movies, int No_of_users, int uid){
    char *line, *record;
    char tmp[1024];
    int i=0, j=0, k=0;
    FILE *fstream = fopen(s,"r");
    while((line=fgets(tmp,sizeof(tmp),fstream))!=NULL){
        record = strtok(line,",");
        while(record!=NULL){
            if(k==0){ //first string is user id, which will give our row
                i = atoi(record)-1;
            }else if(k==1){ //second string is movie id which will give our coloumn
                j = atoi(record)-1;
            }else { //third is the actual rating
                /*if(l == uid){
                    utility_matrix[i*No_of_movies+j] = 0.0;
                }*/
                //else{
                    utility_matrix[i*No_of_movies + j] = atof(record); //converting string to float/double
                //}
            }
            record = strtok(NULL,",");
            k++;
        }
        k=0;
    }
    fclose(fstream);
    free(line);
    free(record);
}
```

Lorsque nous travaillons avec de telles données, nous les voyons principalement sous la forme d'une matrice composée des réactions données par un ensemble d'utilisateurs à certains articles d'un ensemble des articles. Chaque ligne contiendrait les évaluations données par un utilisateur, et chaque colonne contiendrait les évaluations reçues par un article.

Dans la plupart des cas, les cellules de la matrice sont vides, car les utilisateurs n'évaluent que quelques éléments. Il est très peu probable que chaque utilisateur évalue ou réagisse à chaque article disponible. Une



---

matrice avec des cellules principalement vides est appelée clairsemée, et l'opposé de cela (une matrice principalement remplie) est appelée dense.

- ***Trouver la similarité d'un nouvel utilisateur (ou d'un utilisateur concerné) avec d'autres utilisateurs à l'aide de la similarité en cosinus centré***

La similarité cosinus est une mesure de similarité entre deux vecteurs non nuls définis dans un espace de produit interne. La similarité cosinus est le cosinus de l'angle entre les vecteurs; c'est-à-dire qu'il s'agit du produit scalaire des vecteurs divisé par le produit de leurs longueurs. Il s'ensuit que la similarité cosinus ne dépend pas des grandeurs des vecteurs, mais seulement de leur angle.

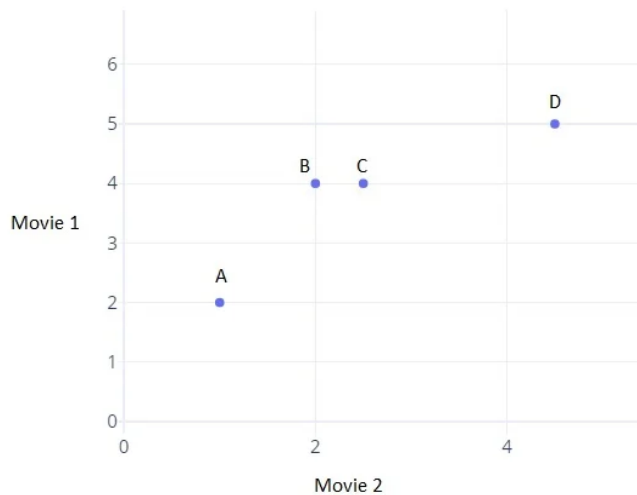
Pour comprendre le concept de similarité, créons d'abord un jeu de données simple. Les données incluent quatre utilisateurs A, B, C et D, qui ont noté deux films. Les classements sont stockés dans des listes, et chaque liste contient deux nombres indiquant le classement de chaque film :

Les notes par A sont [1,0, 2,0]

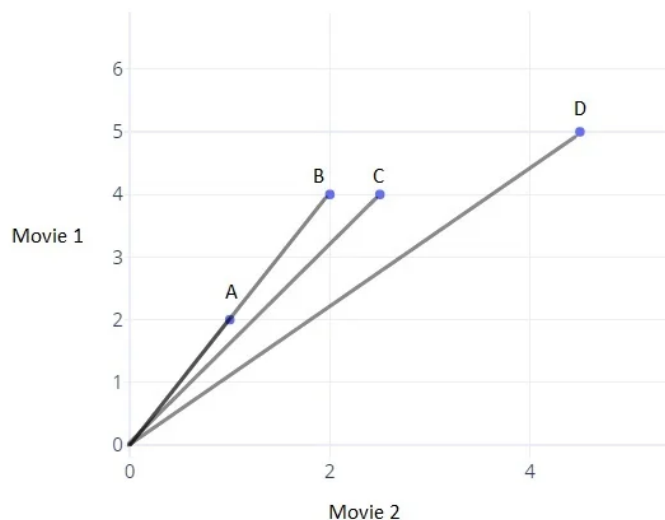
Les notes par B sont [2,0, 4,0]

Les notes par C sont [2,5, 4,0]

Les notes par D sont [4,5, 5,0]



Dans le graphique ci-dessus, chaque point représente un utilisateur et est tracé par rapport aux notes qu'il a attribuées à deux films.



Le graphique montre quatre lignes joignant chaque point à l'origine. Les lignes pour A et B coïncident, rendant l'angle entre elles égal à zéro. Vous pouvez considérer que si l'angle entre les lignes augmente, la similitude diminue et si l'angle est nul, les utilisateurs sont très similaires. Pour calculer la similarité à l'aide de l'angle, vous avez besoin d'une fonction qui renvoie une similarité plus élevée ou une distance plus petite pour un angle inférieur et une similarité

---

inférieure ou une distance plus grande pour un angle supérieur. Le cosinus d'un angle est une fonction qui décroît de 1 à -1 lorsque l'angle augmente de 0 à 180.

Vous pouvez utiliser le cosinus de l'angle pour trouver la similitude entre deux utilisateurs. Plus l'angle est élevé, plus le cosinus sera bas et donc, plus la similarité des utilisateurs sera faible. Vous pouvez également inverser la valeur du cosinus de l'angle pour obtenir la distance cosinus entre les utilisateurs en la soustrayant de 1.

Notez que les utilisateurs A et B sont considérés comme absolument similaires dans la métrique de similarité cosinus malgré des notes différentes. C'est en fait un phénomène courant dans le monde réel, et les utilisateurs comme l'utilisateur A sont ce que vous pouvez appeler des évaluateurs difficiles. Un exemple serait un critique de cinéma qui attribue toujours des notes inférieures à la moyenne, mais le classement des éléments de sa liste serait similaire à celui des évaluateurs moyens comme B.

Pour prendre en compte ces préférences individuelles des utilisateurs, vous devrez amener tous les utilisateurs au même niveau en supprimant leurs préjugés. Vous pouvez le faire en soustrayant la note moyenne donnée par cet utilisateur à tous les éléments de chaque élément évalué par cet utilisateur. Voici à quoi cela ressemblerait :

Pour l'utilisateur A, le vecteur de notation [1, 2] a la moyenne de 1,5. Soustraire 1,5 de chaque note vous donnerait le vecteur [-0,5, 0,5].

---

Pour l'utilisateur B, le vecteur d'évaluation  $[2, 4]$  a la moyenne de 3. En soustrayant 3 de chaque évaluation, vous obtiendrez le vecteur  $[-1, 1]$ .

En faisant cela, vous avez changé la valeur de la note moyenne donnée par chaque utilisateur à 0. Essayez de faire la même chose pour les utilisateurs C et D, et vous verrez que les notes sont maintenant ajustées pour donner une moyenne de 0 pour tous les utilisateurs, ce qui les ramène tous au même niveau et supprime leurs préjugés. Le cosinus de l'angle entre les vecteurs ajustés est appelé cosinus centré. Cette approche est normalement utilisée lorsqu'il y a beaucoup de valeurs manquantes dans les vecteurs et que vous devez placer une valeur commune pour remplir les valeurs manquantes.

Remplir les valeurs manquantes dans la matrice des notations avec une valeur aléatoire peut entraîner des inexactitudes. Un bon choix pour remplir les valeurs manquantes pourrait être la note moyenne de chaque utilisateur, mais les moyennes originales des utilisateurs A et B sont respectivement de 1,5 et 3, et remplir toutes les valeurs vides de A avec 1,5 et celles de B avec 3 serait en faire des utilisateurs différents. Mais après avoir ajusté les valeurs, la moyenne centrée des deux utilisateurs est 0, ce qui vous permet de saisir plus précisément l'idée que l'élément est supérieur ou inférieur à la moyenne pour les deux utilisateurs, toutes les valeurs manquantes dans les vecteurs des deux utilisateurs ayant la même valeur 0.

- 
- ***Prédire les notes des éléments qui ne sont pas encore notés par un utilisateur***

Après avoir déterminé une liste d'utilisateurs similaires à un utilisateur U, vous devez calculer la note R que U donnerait à un certain élément I. Encore une fois, tout comme la similarité, vous pouvez le faire de plusieurs façons. Vous pouvez prédire que la note R d'un utilisateur pour un élément I sera proche de la moyenne des notes attribuées à I par les 5 ou 10 meilleurs utilisateurs les plus similaires à U. La formule mathématique de la note moyenne donnée par n utilisateurs ressemblerait à comme ça:

$$R_U = \left( \sum_{u=1}^n R_u \right) / n$$

Cette formule montre que la note moyenne attribuée par les n utilisateurs similaires est égale à la somme des notes qu'ils ont attribuées divisée par le nombre d'utilisateurs similaires, soit n. Il y aura des situations où les n utilisateurs similaires que vous avez trouvés ne sont pas également similaires à l'utilisateur cible U. Les 3 premiers d'entre eux peuvent être très similaires, et le reste peut ne pas être aussi similaire à U que les 3 premiers. Dans ce cas, vous pouvez envisager une approche dans laquelle la note de l'utilisateur le plus similaire compte plus que celle du deuxième utilisateur le plus similaire, etc. La moyenne pondérée peut nous aider à y parvenir. Dans l'approche de la moyenne pondérée, vous multipliez chaque note par un facteur de similarité (qui indique à quel point les utilisateurs sont similaires). En multipliant par le facteur de

---

similarité, vous ajoutez des pondérations aux notes. Plus le poids est lourd, plus la note compte. Le facteur de similarité, qui agirait comme des poids, devrait être l'inverse de la distance discutée ci-dessus car moins de distance implique une plus grande similarité. Par exemple, vous pouvez soustraire la distance cosinus de 1 pour obtenir la similarité cosinus. Avec le facteur de similarité  $S$  pour chaque utilisateur similaire à l'utilisateur cible  $U$ , vous pouvez calculer la moyenne pondérée à l'aide de cette formule :

$$R_U = (\sum_{u=1}^n R_u * S_u) / (\sum_{u=1}^n S_u)$$

Dans la formule ci-dessus, chaque note est multipliée par le facteur de similarité de l'utilisateur qui a donné la note. La note finale prédite par l'utilisateur  $U$  sera égale à la somme des notes pondérées divisée par la somme des pondérations. Avec une moyenne pondérée, vous accordez plus d'importance aux évaluations d'utilisateurs similaires par ordre de similarité.

### 3.2 Communication:

Le projet étant au stade de développement la communication pour le moment n'a pas lieu d'être mais par la suite elle sera très importante pour ramener le plus d'utilisateur et récolter des données qui nous seront très utiles pour l'algorithme de recommandation.

---

### 3.3 Données:

Nous n'avons pas encore commencé cette partie, elle sera réalisée après cette première soutenance.

### 3.4 Interface:

Nous sommes pour l'instant en train de comparer les différentes mises en page que nous avons produites pour chacune des sections de l'application dans le but de choisir celle qui allie le mieux la pratique et l'esthétique. Pour cela, nous avons utilisé les outils Paint et Figma, qui permettent de produire rapidement plusieurs schémas de pages, afin de pouvoir voir à quoi ressemblerait l'interface. L'objectif pour cette première soutenance était d'avoir choisi la mise en forme de l'interface, ainsi que d'autres choix, pour la plupart esthétiques, tels que celui de la palette de couleurs ou de la police d'écriture qui seront utilisées dans la version finale.

---

### 3.5 Réseau:

Tout d'abord il faut savoir que nos Cours et Travaux Pratiques de Programmation traitent du sujet des serveurs et plus particulièrement à travers les TP 2 et 3 où nous avons dû implémenter respectivement un client et un serveur avec le protocole TCP.

Nous avons donc des bases que nous avons pu utiliser avec quelques améliorations, par exemple avec l'utilisation des fonctions `send()` et `recv()` qui permettent un transfert de données entre le serveur et le client plus simplement.

Le serveur utilise la fonction `select()` qui va permettre de surveiller des descripteurs de fichier, ici dans notre cas les sockets des clients. On va l'utiliser pour détecter quand un socket est disponible pour la lecture, si c'est le cas la fonction `FD_ISSET()` va renvoyer vrai si on lui passe en paramètre un descripteur de fichier qui a été sélectionné précédemment.



---

Architecture global du serveur :

```
fd_set rdfs;

FD_ZERO(&rdfs);
FD_SET(STDIN_FILENO, &rdfs);
FD_SET(socketServer, &rdfs);

for (int i = 0; i < actual; i++)
    FD_SET(clients[i].sock, &rdfs);

select(max + 1, &rdfs, NULL, NULL, NULL)

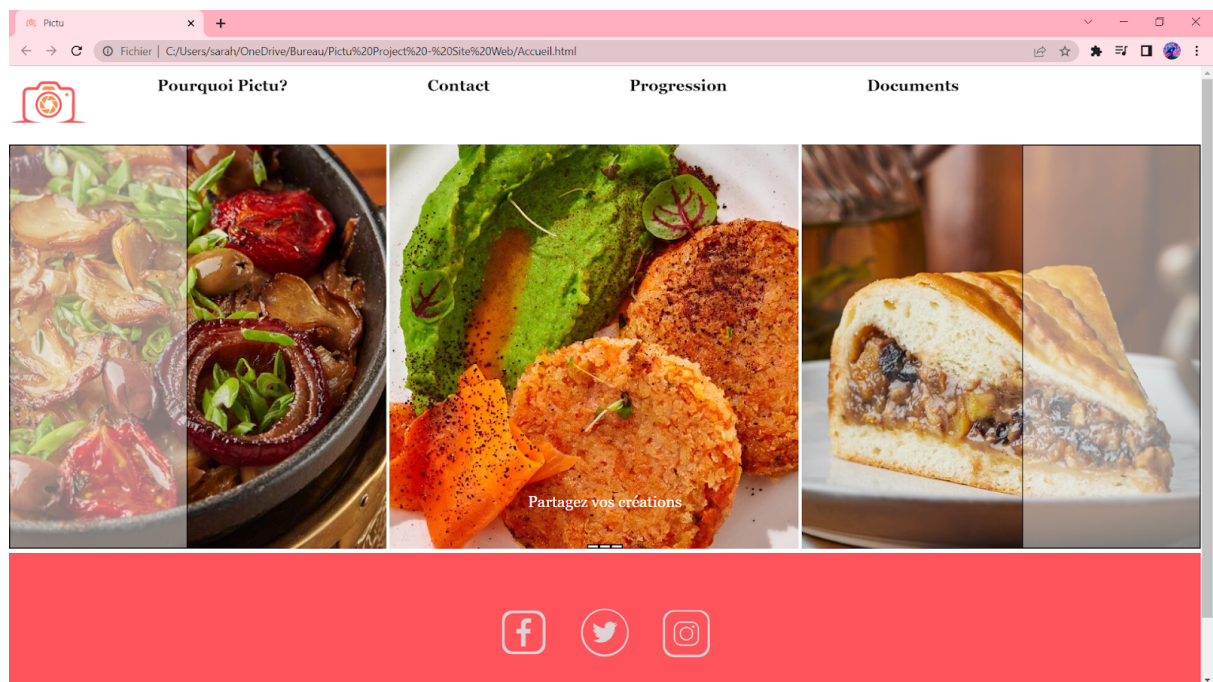
//Input dans le server -> stop
if(FD_ISSET(STDIN_FILENO, &rdfs))
    break;
//Input dans le socket du serveur -> Nouvelle connection
else if (FD_ISSET(socketServer, &rdfs))

else //On test tout les autres socket qui sont des clients
    for(int i = 0; i < actual; i++)
        if(FD_ISSET(clients[i].sock, &rdfs))
            //Client is talking
```

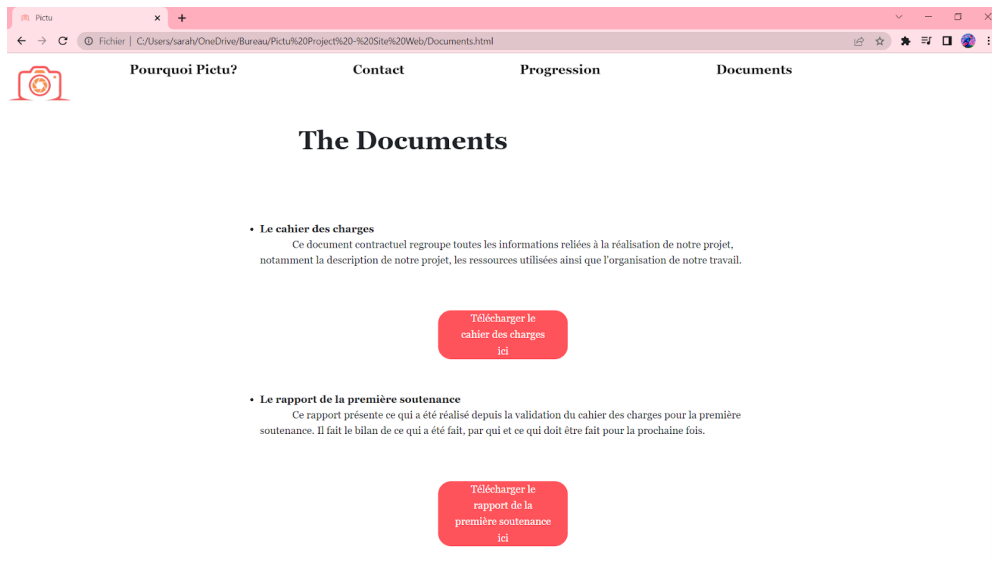
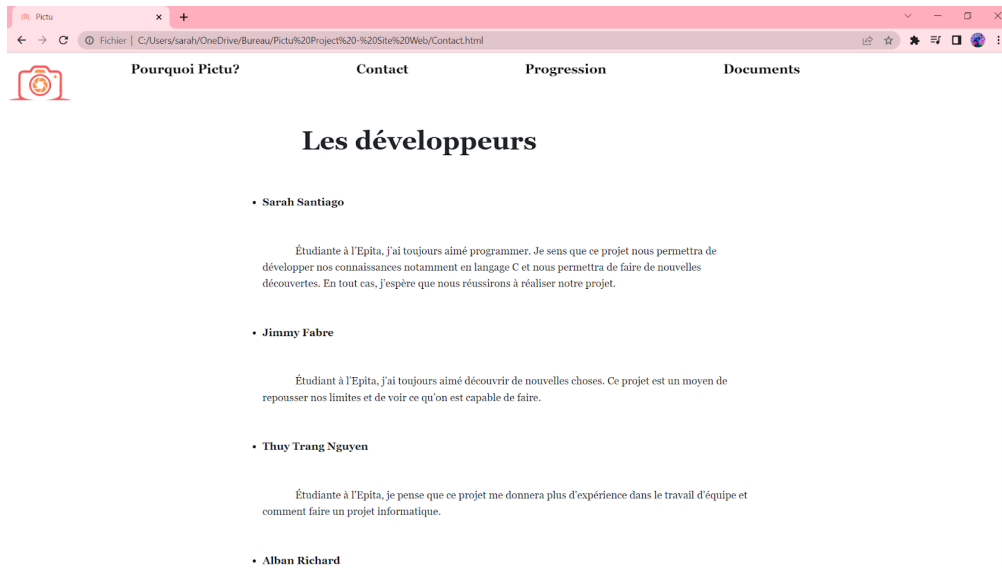
Pour l'instant le serveur permet à l'utilisateur de se connecter et faire passer un identifiant au serveur. Il n'est pas encore implémenté avec des Threads, il va donc traiter une demande après l'autre.

### 3.6 Site Web:

En ce qui concerne cette partie, nous avons commencé à programmer un site local. Nous nous sommes servis d'une couleur de notre logo pour le site, c'est une couleur douce, non agressive qui attire le regard.



Nous avons déjà commencé à compléter plusieurs parties notamment les membres ou encore les documents, il est aussi possible de télécharger les rapports que nous avons réalisés jusque là.



### 3.7 Installateur / Lanceur:

Nous n'avons pas encore commencé cette partie, elle sera réalisée après la deuxième soutenance.

---

#### **4/Nos impressions :**

Sarah Santiago:

Pour l'instant, je pense personnellement que nous avançons à un bon rythme qui nous permettra de réussir ce projet.

J'ai aussi aimé le fait que j'ai pu améliorer mes bases en création de site web, ce qui est très enrichissant et pourra m'être utile plus tard.

Jimmy Fabre:

Le groupe avance à un bon rythme, j'ai beaucoup aimé faire des recherches sur comment fonctionne un serveur et comment l'implémenter.

Thuy Trang Nguyen:

Pour l'instant, je pense personnellement que nous avançons à un bon rythme qui nous permettra de réussir ce projet. J'aime beaucoup ce travail car il me donne la capacité de lire des documents, même si c'est très difficile.

Alban Richard:

Je trouve que nous progressons pour l'instant à une bonne vitesse. Je m'occupe de parties que je ne fais pas habituellement, ce qui est à la fois un avantage et un inconvénient. En effet, cela me permet d'apprendre de nouvelles choses, mais je suis ralenti par le fait que je doive apprendre justement. Aussi, je trouve l'atmosphère du groupe agréable.

---

## 5/Prévisions pour la prochaine soutenance :

Pour la prochaine soutenance, nous comptons maintenir le rythme et atteindre les objectifs que nous avons fixés dans notre cahier des charges.

Pour rappel, nous avons fixés ce tableau:

Répartition des tâches	1re soutenance	2e soutenance	3e soutenance
Algorithme	☆ ☆	☆ ☆	☆ ☆ ☆
Communication		☆	☆ ☆ ☆
Données		☆	☆ ☆ ☆
Interface	☆	☆ ☆	☆ ☆ ☆
Réseaux	☆	☆	☆ ☆ ☆
Site Web	☆	☆	☆ ☆ ☆
Installateur / Lanceur			☆ ☆ ☆

Legende : ☆ = Tâche commencée ; ☆ ☆ = Tâche avancée ; ☆ ☆ ☆ = Tâche terminée

Cependant, des petites étoiles ne représentent rien de concret. Il serait plus judicieux de donner des objectifs réels afin de ne pas rester trop vague. Ainsi, voici une liste d'objectifs que nous nous fixons pour la prochaine soutenance:

---

## Algorithme:

Nous comprenons toutes les étapes qui doivent être suivies pour compléter la recommandation, mais nous n'avons pas encore fait le code avec succès. Dans la soutenance suivante, nous allons compiler le code.

## Communication:

Pour la prochaine soutenance, nous allons commencer à parler de notre projet, potentiellement créer un discord pour créer une petite communauté et fixer une date de sortie.

## Données:

Pour la prochaine soutenance, nous comptons commencer cette partie en récoltant des set de photos.

## Interface:

Le but que nous nous sommes fixés pour la prochaine soutenance est d'avoir un prototype fonctionnel à présenter. Il ne manquera alors plus qu'à peaufiner les détails et à la relier au serveur.

---

## Réseau:

Comme dit précédemment le serveur ne peut gérer qu'une instruction à la fois, dû à la fonction `recv()` qui va bloquer le programme jusqu'à attendre une information envoyée par le client mais l'utilisation de threads réglera ce problème.

Mais aussi, nous avons rencontré quelques problèmes inattendus avec le serveur, car étant donné que nous allons devoir traiter un set de données assez important une des premières idées qui était d'utiliser le format csv pour stocker les données est donc devenu impossible, après plusieurs recherches nous avons découvert qu'il était possible d'utiliser Sqlite3 à partir de code en C.

Donc pour la prochaine soutenance l'objectif dans cette partie est d'implémenter toutes les fonctions nécessaires pour traiter avec la base de données et d'implémenter les Threads pour rendre le serveur multi-tâche.

## Site Web:

Pour ce qui est du site, nous continuerons à le remplir ainsi qu'à le rendre plus accueillant et attrayant pour les utilisateurs.

## Installateur / Lanceur:

Pour la prochaine soutenance, nous ne comptons pas encore commencer cette partie.

---

## **6/Conclusion :**

En conclusion, grâce aux informations rassemblées et aux essais dans les divers domaines que nous avons définis pour ce projet, nous sommes à même de le mener à bien. En ayant respecté les objectifs que nous nous sommes fixés, nous nous rendons toutefois compte que certains points peuvent être améliorés, d'autres sont encore complètement à construire, mais nous restons confiants en notre capacité à atteindre les objectifs que nous nous sommes fixés afin de délivrer une expérience bien construite et agréable pour l'utilisateur.