**Analyse UML :application suivi sportif**

**Introduction :**

L’application envisagée vise à offrir une plateforme intuitive permettant aux utilisateurs d’enregistrer leurs séances d’entraînement et de suivre leur consommation alimentaire, y compris les macronutriments. En s’appuyant sur les principes de l’ingénierie logicielle, nous élaborerons des diagrammes UML pour conceptualiser la structure et le comportement de l’application. Cette démarche garantira une architecture logicielle robuste et une expérience utilisateur optimale.

**Cahier des charges :**

**1. Objectifs de l'application :**

-application tout en un qui permet le suivi d’entrainement et suivi calorique avec une interface utilisateur épuré.

- Faciliter la planification d'entraînements personnalisés en fonction des objectifs, du niveau de forme physique et des préférences de l'utilisateur.

- Offrir une variété d'exercices et de programmes d'entraînement adaptés à différents types d'activités physiques (cardio, musculation, yoga, etc.).

- Suivre les progrès des utilisateurs au fil du temps, en enregistrant les données telles que le poids, les performances, les séances d'entraînement réalisées, etc.

- Fournir des conseils et des recommandations basés sur les données recueillies, pour aider les utilisateurs à ajuster et à optimiser leurs routines d'entraînement.

- Intégrer des fonctionnalités sociales pour permettre aux utilisateurs de partager leurs réussites, défier leurs amis et trouver de l'inspiration au sein de la communauté.

-analyse des performances

-objectifs personalisés

**2. Fonctionnalités :**

**a. fonctionnalités essentielles**

- **Création de compte** : Les utilisateurs peuvent créer un compte personnel en fournissant des informations telles que leur nom, leur adresse e-mail, leur mot de passe, l'âge, le sexe, le poids, la taille et les objectifs de remise en forme. leur permettant ainsi d'accéder à toutes les fonctionnalités de l'application.

- **Enregistrement de l'entraînement :** Les utilisateurs peuvent enregistrer leurs séances d'entraînement, en spécifiant les exercices effectués, la durée et l'intensité de l'entraînement. En parcourant une Base de données d'exercices avec instructions détaillées, vidéos et démonstrations.

- **Ajout d'amis** : Les utilisateurs ont la possibilité de connecter et d'interagir avec leurs amis au sein de l'application, partageant ainsi leurs progrès, se lançant des défis et se motivant mutuellement.

- **Enregistrement du poids** : Les utilisateurs peuvent enregistrer leur poids régulièrement, ce qui leur permet de suivre leur évolution et d'ajuster leurs objectifs en conséquence.

- **Envoi de messages** : Les utilisateurs peuvent communiquer entre eux via un système de messagerie interne, facilitant ainsi les échanges et le partage d'informations.

- **Consultation des statistiques** : Les utilisateurs ont accès à des statistiques détaillées sur leurs activités physiques, y compris les calories brûlées, le temps d'entraînement, etc., les aidant à évaluer leurs progrès et à ajuster leurs routines.

- **Consultation du leaderboard** : Les utilisateurs peuvent consulter un classement des performances parmi leurs amis, les encourageant à rester motivés et à atteindre de nouveaux objectifs.

**- Modification du profil** :Les utilisateurs ont la possibilité de modifier leurs informations personnelles et leurs préférences à tout moment, assurant ainsi une expérience utilisateur personnalisée.

- **Enregistrement de la nourriture et des calories** :Les utilisateurs peuvent enregistrer leur consommation alimentaire et estimer les calories consommées, permettant ainsi une meilleure gestion de leur régime alimentaire en lien avec leurs objectifs de remise en forme.

- **Création et consultation de programmes d'entraînement** : Les coach ont la possibilité des programmes d'entraînement.

- **Création de groupe** : Les utilisateurs peuvent créer des groupes thématiques, tels que des groupes d'étude ou des groupes de motivation, où ils peuvent partager des conseils, des défis et des progrès.

**b. Fonctionnalités bonus :**

-**planification entrainement à l’avance** :en sélectionnant une date du futur on pourrait choisir de planifier un entrainement a une certaine heure ce qui enverrait une notification sur notre téléphone avant l’entrainement.

- **Tracking des pas** : Les utilisateurs peuvent suivre leur activité quotidienne en enregistrant le nombre de pas effectués, ce qui contribue à une meilleure compréhension de leur niveau d'activité physique global.

- **Ajouts aliments par qr code** : Les utilisateurs ajouter les aliments en scannant leur code barres avec leur smartphone.

-**Intégration avec des dispositifs portables (wearables**) : pour une collecte automatique de données.

En mettant en œuvre ces fonctionnalités, notre application deviendra un outil indispensable pour les étudiants et toute personne désireuse de concilier avec succès études, travail et bien-être physique.

**3. Plateformes prises en charge :**

- Application mobile pour iOS et Android.

-Bonus : Version web pour une accessibilité depuis n'importe quel navigateur.

**4. Technologies utilisées :**

- Développement iOS et Android sous React native.

- Utilisation de bases de données pour le stockage et la gestion des profils utilisateur, des programmes d'entraînement et des données de suivi.

- Intégration d'API pour l'accès à des bases de données d'exercices, des services de géolocalisation (pour les activités en extérieur), et éventuellement des dispositifs portables.

**5. Sécurité et confidentialité :**

- Protection des données personnelles des utilisateurs conformément aux réglementations en vigueur (RGPD, CCPA, etc.).

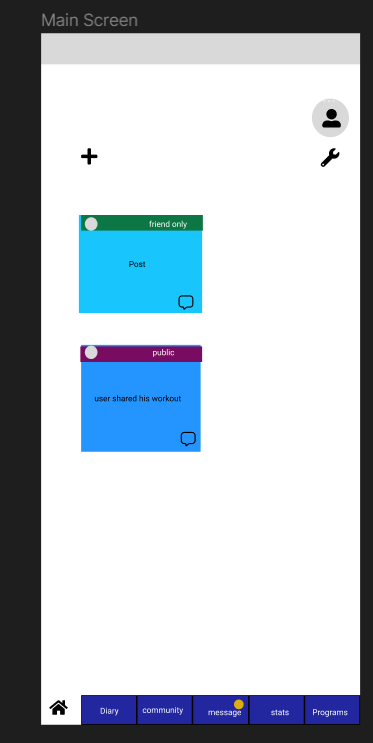
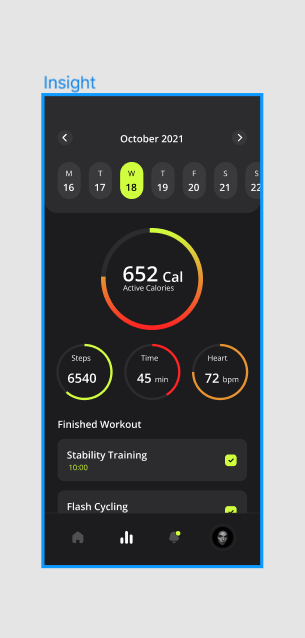
- Utilisation de mesures de sécurité telles que le chiffrement des données et l'authentification sécurisée.

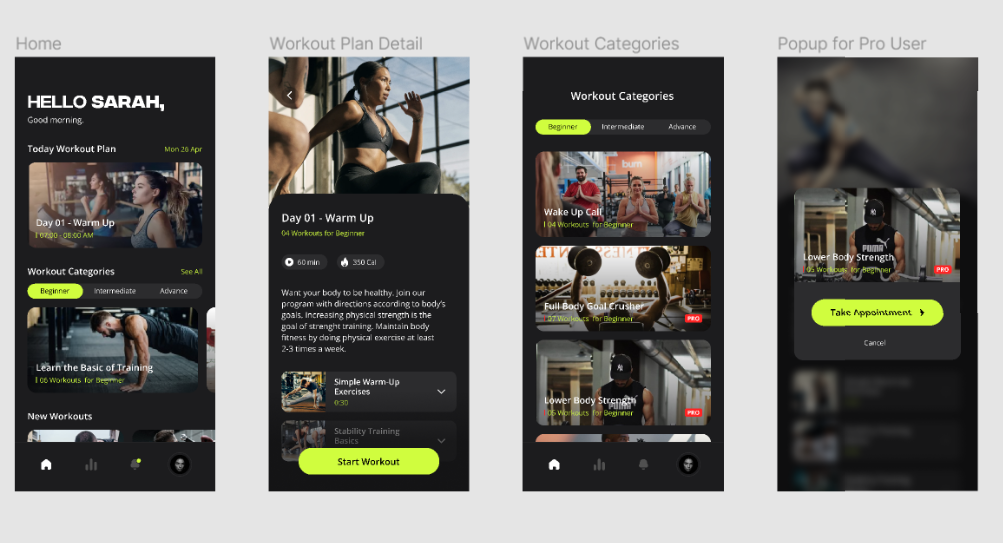
**6. Monétisation :**

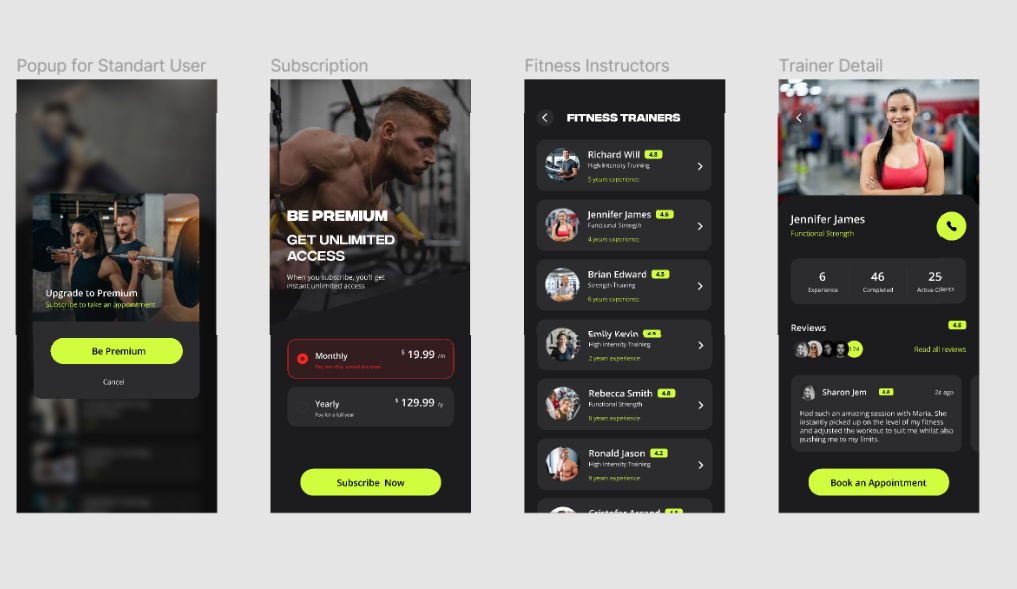
- Modèle freemium avec une version de base gratuite offrant des fonctionnalités limitées et des abonnements premium pour un accès complet aux fonctionnalités avancées.

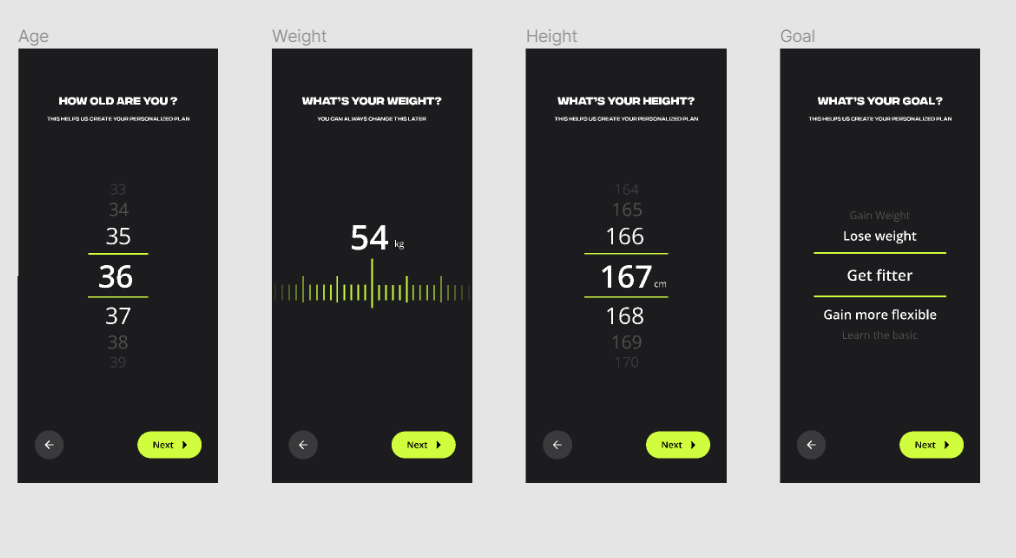
- Possibilité de générer des revenus supplémentaires grâce à des publicités ciblées et des partenariats avec des marques de fitness et de bien-être.

**IHM :**

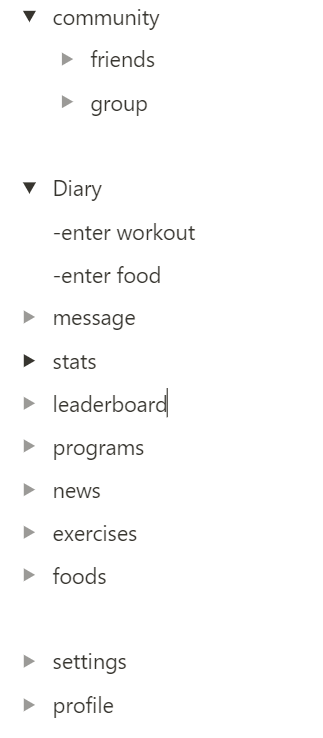
 







Vue des different onglet:



**Acteurs :**

**-Membres : utilisateur normal de l’application**

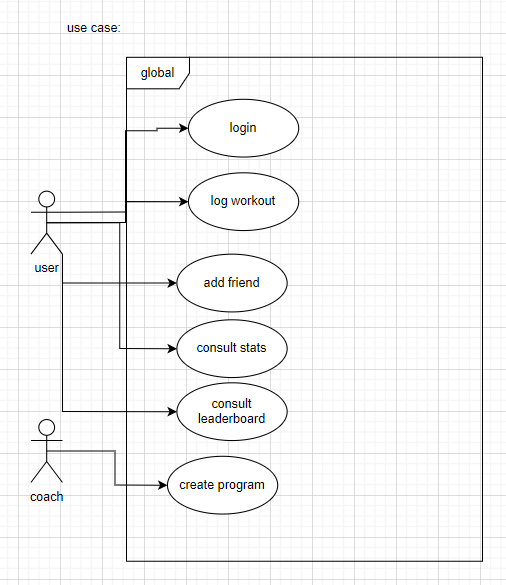
**-Coach : utilisateur certifié qui peut créer des programmes d’entrainement et ajouter des clients**

**-Administrateur :utilisateur qui peut gérer l’application**

**Cas d’utilisations :**

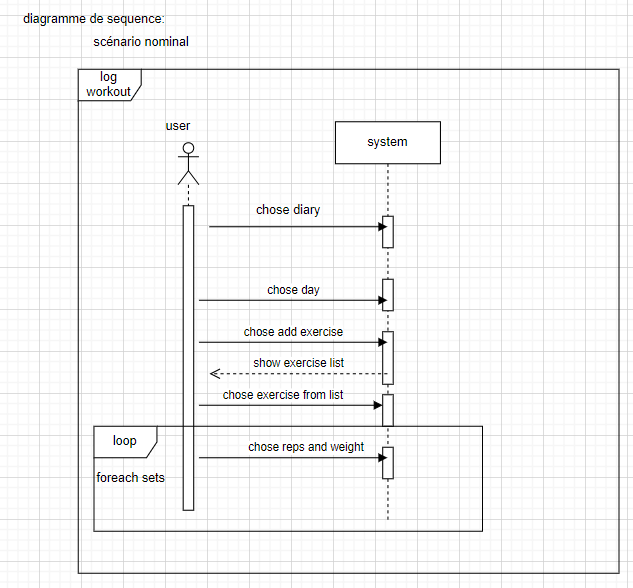
**-Vue global**

|  |  |
| --- | --- |
| Nom | Granularite faible,vue global |
| Description | Vue globale du fonctionnement de l’app. |
| Acteurs | Un utilisateur, un coach |
| Précondition | L’utilisateur ouvre l’application |
| Déclencheur | / |
| Fréquence | / |
| postcondition | / |
| commentaire | / |
| Scénario nominal | / |



**-Enregistrer son entrainement**

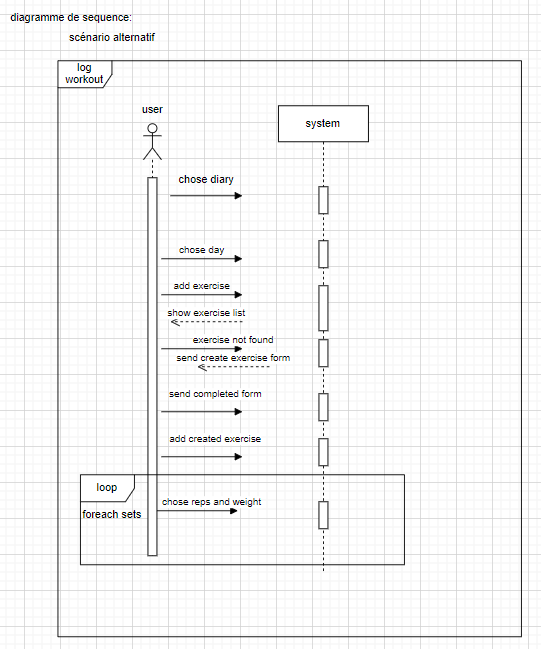
**Nominal :**



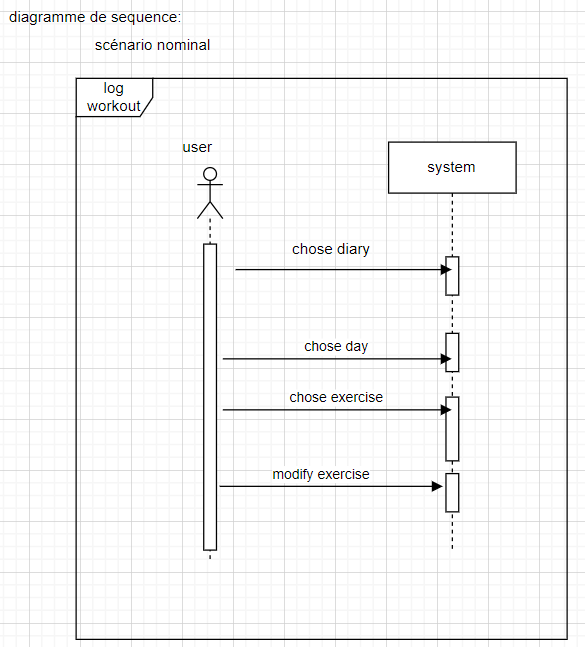
|  |  |
| --- | --- |
| Nom | **Enregistrer son entrainement-exercices trouvés** |
| Description | L’utilisateur enregistre son entrainement dans l’application |
| Acteurs | Utilisateur |
| Précondition | L’utilisateur doit trouver son exercice dans la liste |
| Déclencheur | Il doit vouloir enregistrer son entrainement du jour |
| Fréquence | À chaque fois qu’il veut enregistrer son entrainement. |
| Postcondition | Son entrainement est enregistré dans le système. |
| Commentaire | L’utilisateur va dans son journal,choisis le jour,clique sur ajouter exercice,le système lui montre une liste d’exercice disponible dans la database,il choisit et il ajoute ensuite le poids et répétitions pour chaques series.L’utilisateur peut ajouter plusieurs exercise si il le souhaite de cette façon. |

**Alternatif :**

|  |  |
| --- | --- |
| Nom | **Enregistrer son entrainement-exercice non trouvé** |
| Description | L’utilisateur enregistre son entrainement dans l’application |
| Acteurs | Utilisateur |
| Précondition | L’utilisateur ne trouve pas son exercice dans la liste |
| Déclencheur | Il doit vouloir enregistrer son entrainement du jour |
| Fréquence | À chaque fois qu’il veut enregistrer son entrainement. |
| Postcondition | Son entrainement est enregistré dans le système. |
| Commentaire | L’utilisateur va dans son journal,choisis le jour,clique sur ajouter exercice,le système lui montre une liste d’exercice disponible dans la database,il clique sur exercice pas trouvé,il crée l’exercice ila ajoute son exercise créé et il ajoute ensuite le poids et répétitions pour chaques series.L’utilisateur peut ajouter plusieurs exercise si il le souhaite de cette façon. |



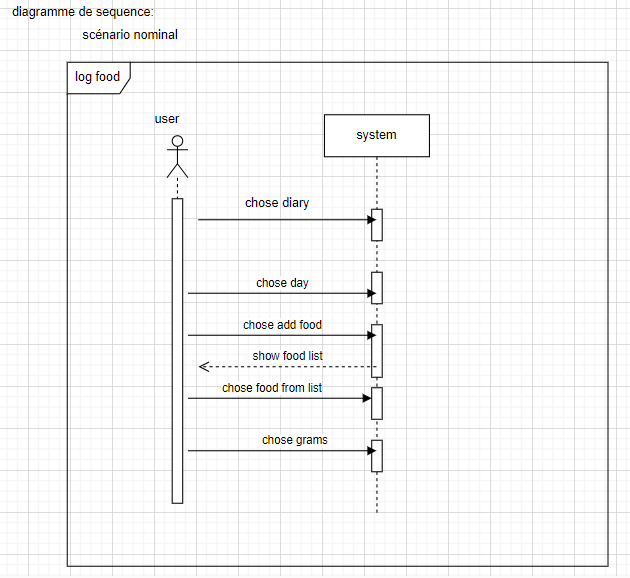
|  |  |
| --- | --- |
| Nom | **Enregistrer son entrainement-modifier exercices** |
| Description | L’utilisateur veut modifier son entrainement |
| Acteurs | Utilisateur |
| Précondition | L’utilisateur à déjà enregistrer des exercices |
| Déclencheur | Il doit vouloir modifier un entrainement |
| Fréquence | À chaque fois qu’il veut modifier un entrainement |
| Postcondition | Son entrainement a été modifié |
| Commentaire | L’utilisateur va dans son journal,choisis le jour,clique sur un exercice qu’il avait enregistrer,modifiele poids repetition series,. |



**-Ajouter des aliments**

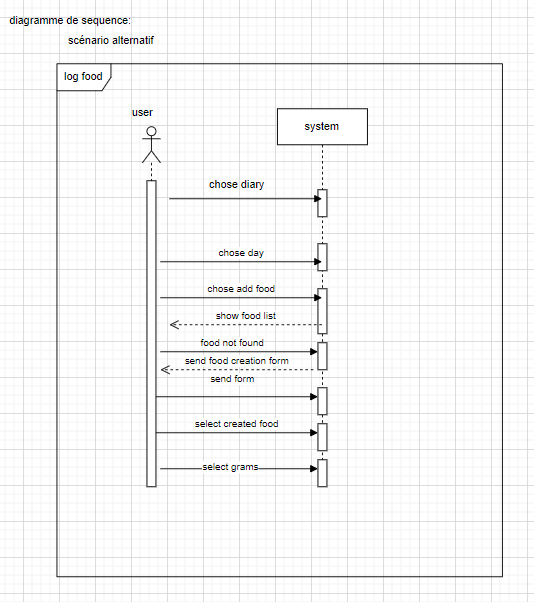
**Nominale :**

|  |  |
| --- | --- |
| Nom | **Ajouter des aliments-aliments trouvé** |
| Description | L’utilisateur enregistre son entrainement dans l’application |
| Acteurs | Utilisateur |
| Précondition | L’utilisateur trouve les aliments dans la base de donné |
| Déclencheur | Il doit vouloir enregistrer ses aliments |
| Fréquence | À chaque fois qu’il veut enregistrer son entrainement. |
| Postcondition | Son entrainement est enregistré dans le système. |
| Commentaire | L’utilisateur va dans son journal,choisis le jour,clique sur ajouter aliemnts,le système lui montre une liste d’aliments disponible dans la database,il choisis sont aliment,il choisis les grames.Il peut ajouter plusieurs aliments de cette façon. |



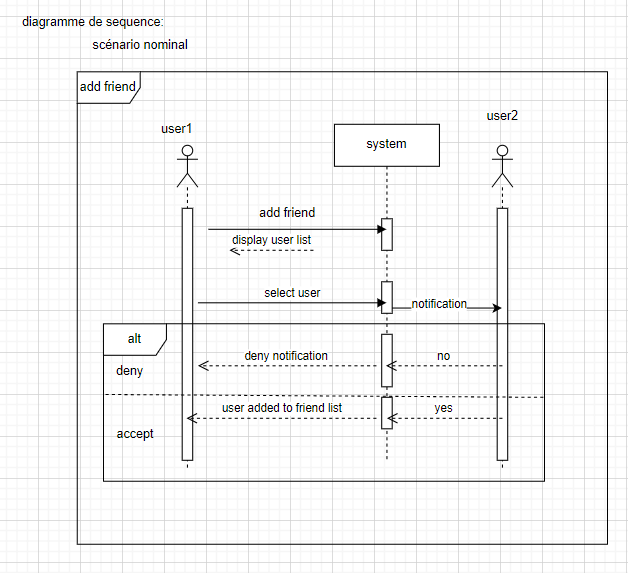
**Alternatif :**

|  |  |
| --- | --- |
| Nom | **Ajouter des aliments-aliments non trouvé** |
| Description | L’utilisateur enregistre son entrainement dans l’application |
| Acteurs | Utilisateur |
| Précondition | L’utilisateur ne trouve pas les aliments dans la base de donné |
| Déclencheur | Il doit vouloir enregistrer ses aliments |
| Fréquence | À chaque fois qu’il veut enregistrer ses aliments |
| Postcondition | Son entrainement est enregistré dans le système. |
| Commentaire | L’utilisateur va dans son journal,choisis le jour,clique sur ajouter aliemnts,le système lui montre une liste d’aliments disponible dans la database,il ne trouve pas,il crée un aliment,il choisis son aliment créé en mettant les grammes. Il peut ajouter plusieurs aliments de cette façon. |



-**ajouter un ami**

|  |  |
| --- | --- |
| Nom | **Ajouter un ami** |
| Description | L’utilisateur cherche et ajoute un ami |
| Acteurs | Utilisateur1 et utilisateur 2 |
| Précondition | L’utilisateur trouve l’ami dans la base de donné |
| Déclencheur | Il veut ajouter un ami |
| Fréquence | À chaque fois |
| Postcondition | Il a envoyé sa demande d’ami |
| Commentaire | L’utilisateur clique sur ajouter ami potentiel,le système montre une liste d’utilisateur il le parcourt pour trouver son ami le selectionne .L’utilisateur 2 recoit la notification,si il refuse utilisateur 1 recoit la notif de refus, sinon il est ajouté a sa liste |



**Modèle MVC** :

* + **Modèle** : Contiendrait la logique de données de votre application, comme les informations sur les utilisateurs, les séances d’entraînement, les statistiques de progression, etc.
  + **Vue** : Serait responsable de l’affichage de l’interface utilisateur, permettant aux utilisateurs de voir leurs plans d’entraînement, leurs progrès, et d’interagir avec l’application.
  + **Contrôleur** : Gérerait la logique de traitement des entrées de l’utilisateur, en communiquant entre la vue et le modèle pour mettre à jour les données ou récupérer les informations nécessaires.

**Classes participatives :**

**Pour les cas d’enregistrer son entrainement : on devra surement avoir besoin,**

1. **Classe Utilisateur** : Représente l’utilisateur de l’application, avec des attributs tels que nom, âge, poids, taille, et des méthodes pour mettre à jour ses informations ou consulter ses progrès.
2. **Classe Exercice** : Représente un exercice spécifique, avec des attributs comme nomExercice, description, nombreDeRépétitions, nombreDeSéries, et peut-être une référence à une **Image** ou une **Vidéo** démonstrative.
3. **Classe jour** : Représente une instance d’une séance d’entraînement, avec des attributs pour la date, commentaire, et une liste des **Exercices** effectués.

**Pour les cas d’enregistrer sa nourriture : on devra surement avoir besoin,**

1. **Classe Utilisateur** **et Jour cité plus haut .**
2. **Classe Aliment** : Représente un aliment spécifique, avec des attributs comme nom, description, grames, calories.

**Pour les cas ajouter un ami: on devra surement avoir besoin,**

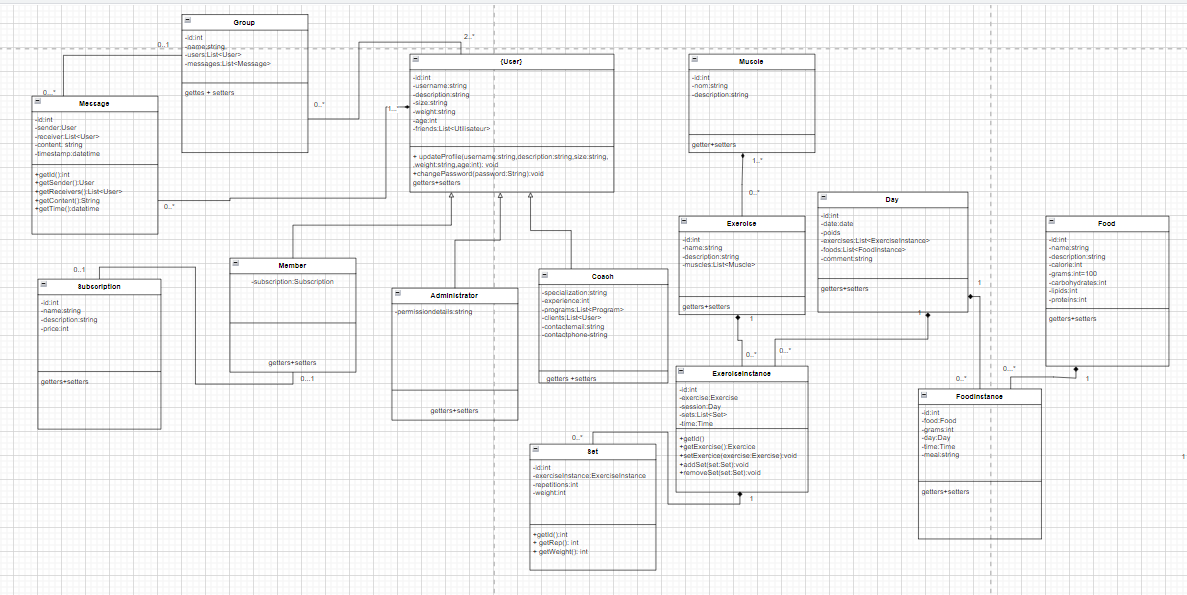
1. . **Classe Utilisateur** :et on ajoute une liste d’amis dans les attributs

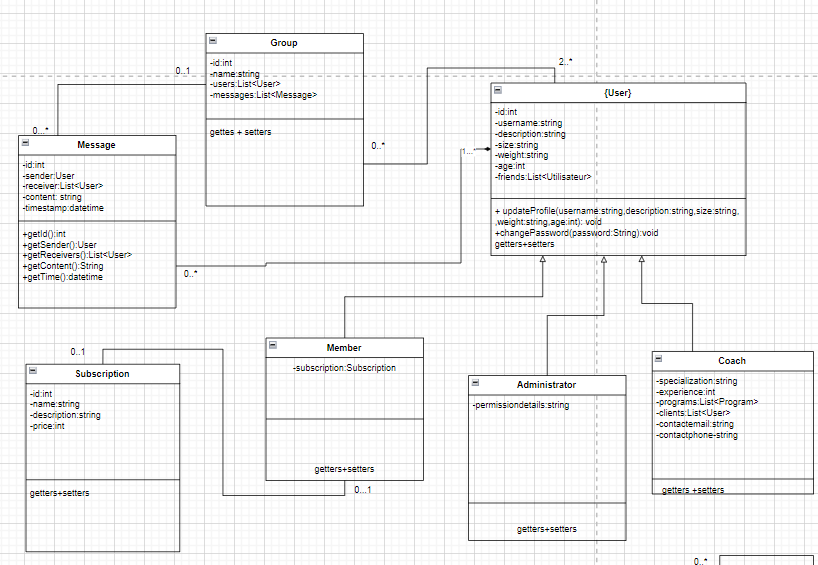
**Classe d’analyse :**

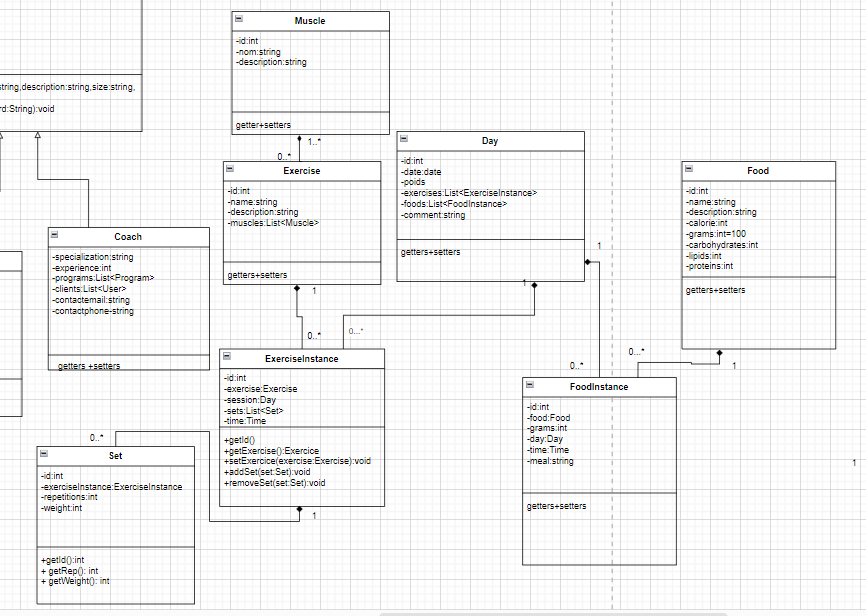
De ce fait je peux dresser une liste de classes pourraient inclure :

* **Utilisateur** : Contient des informations sur l’utilisateur, comme l’identifiant, le mot de passe, l’âge, le sexe, le poids, la taille et les objectifs de remise en forme. Avec comme sous classe membre,admin,coach
* **Jour** : Représente une instance d’une séance d’entraînement, avec des attributs pour la date, commentaire, et une liste des **Exercices** effectués.
* **Exercice** : Décrit un exercice spécifique, incluant son nom, une description, les instructions, et éventuellement une vidéo de démonstration.
* **Muscle :** database avec les différents muscles qui sera lié aux exercices
* **Sets :** sera lié aux exercices du jour contient le poids et les répétitions
* **Abonnement :** database avec type d’abonnements
* **Message** : database des messages
* **Groupe** : Permet aux utilisateurs de créer et de gérer des groupes thématiques pour la motivation et le partage d’expériences.
* **Aliment :** database d’aliments contenant description calories macronutriments.

**Diagramme de classe :**







**Commentaire :**

-utilisateur est une classe abstraite ; un utilisateur peut être soit un membre ,un admin ou un coach

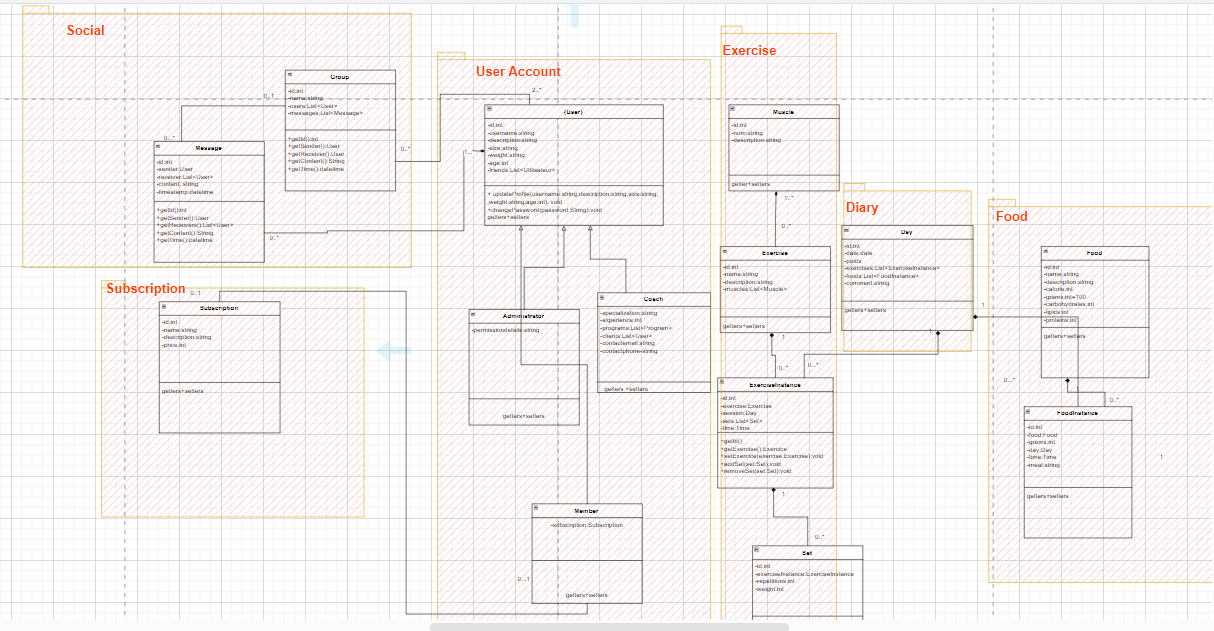
-Un groupe est constitué de minimum 2 utilisateurs et contient des messages. Un message n’est pas forcement lié à un groupe. on peut envoyer un message directement à un ami. C’est pour ca que le receveur peut être une liste d’utilisateur.

-Un membre peut avoir un abonnement.

-Un jour a une liste d’exercice et aliments.

-une liste d’aliments et d’exercices sont stocke en database,lors de l’enregistrement on va créer une instance de ceux-ci lié au jour choisi avec des attributs ajoutés.

**Découpage en package :**



**Diagramme composants :**

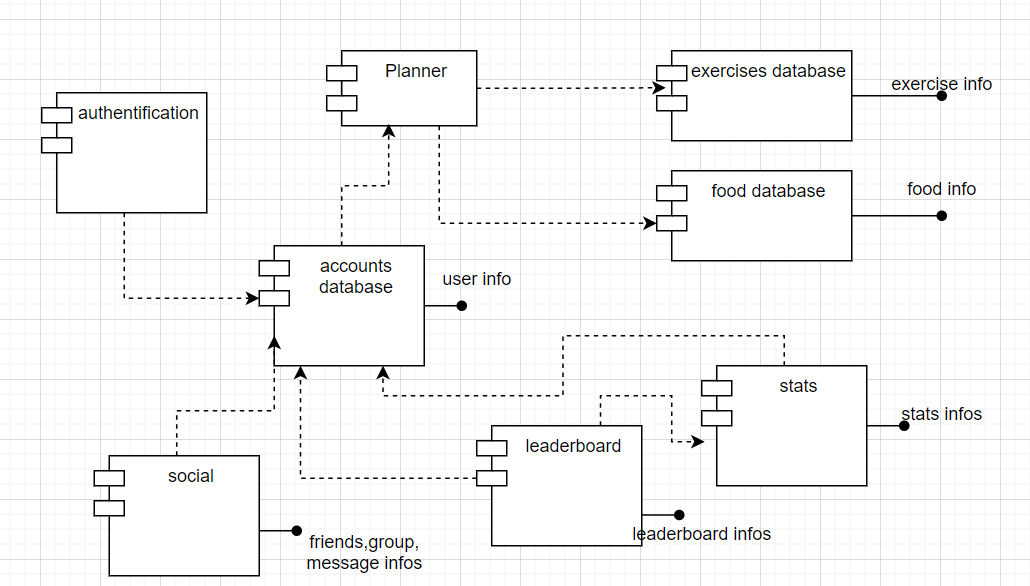


Diagramme activité :

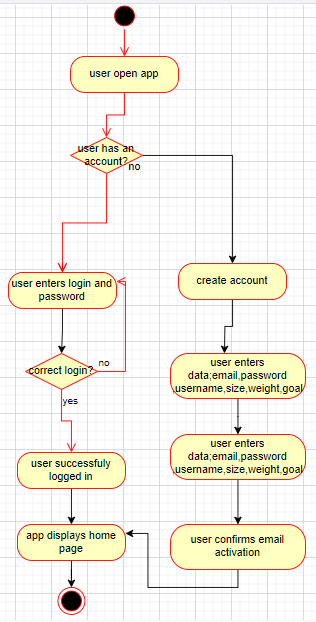


Diagramme de déploiement

