

Cahier des charges

Table des matières

Contexte et présentation du projet.....	2
Contexte.....	2
Objectif.....	2
Informations concernant le complexe sportif.....	2
Les points.....	2
Les activités.....	2
Les spécificités.....	3
Besoins et contraintes liés au projet.....	4
Besoins fonctionnels.....	4
Conception.....	4
Contraintes.....	8
Résultats attendus.....	9
Résultats attendus.....	9
Exigences.....	9

Contexte et présentation du projet

Contexte

Application pour un complexe sportif qui permettra de gérer les adhérents et les activités.

Objectif

Permettre au complexe sportif de pouvoir d'être géré plus simplement.

Informations concernant le complexe sportif

Les points

1 point coûte 0.5 €.

Les activités

Une séance est limitée à 50 personnes

Liste des activités disponibles et de leur coût :

- Une séance de musculation : 10 points
- Une séance d'aquagym : 15 points
- Une séance de SPA : 20 points
- Une séance de danse : 10 points
- Une séance de loisirs créatifs : 10 points
- Une séance de basket : 10 points
- Une séance de football : 10 points
- Une séance d'escape game : 20 points
- Une séance d'athlétisme : 10 points

- Une séance de massage : 15 points
- Une séance de boxe française : 10 points
- Une séance d'escrime : 10 points
- Une séance de gym : 10 points
- Une séance d'équitation : 15 points

Les spécificités

Les points utilisés

Si un adhérent à utiliser 50 points, il obtient 10 points et s'il est mineur ou qu'il a plus de 60 ans, il obtient 5 points gratuits en plus.

Fréquentation du centre

Un adhérent ne peut fréquenter le centre qu'une fois par jour.

Un adhérent ne peut réserver que trois activités en même temps.

Besoins et contraintes liés au projet

Besoins fonctionnels

- Ajouter un adhérent
- Alimenter une carte
- Activer/Désactiver une carte
- Supprimer un adhérent
- Changer l'âge d'un adhérent
- Afficher les informations d'un adhérent
- Afficher les informations de tous les adhérents
- Voir la liste des activités disponibles
- Réserver une activité
- Afficher le nombre d'entrées par activités

Conception

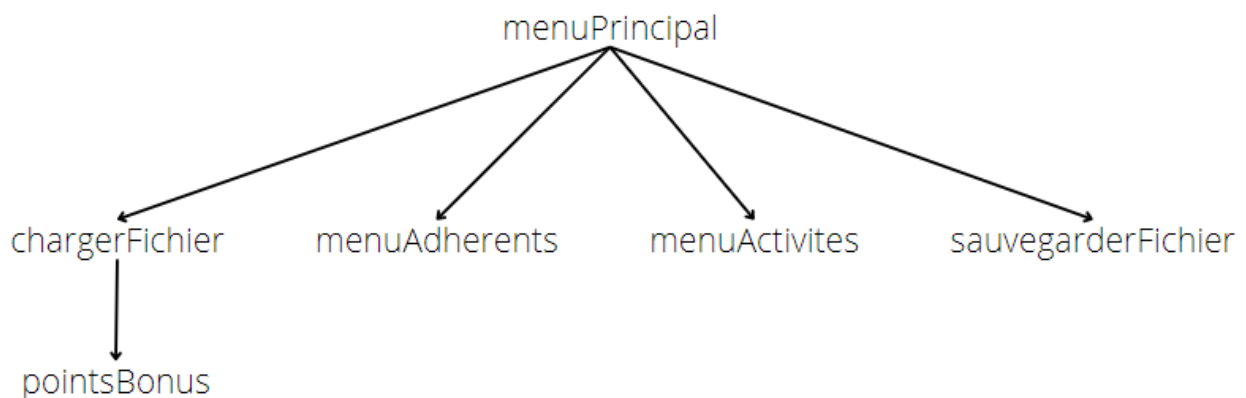
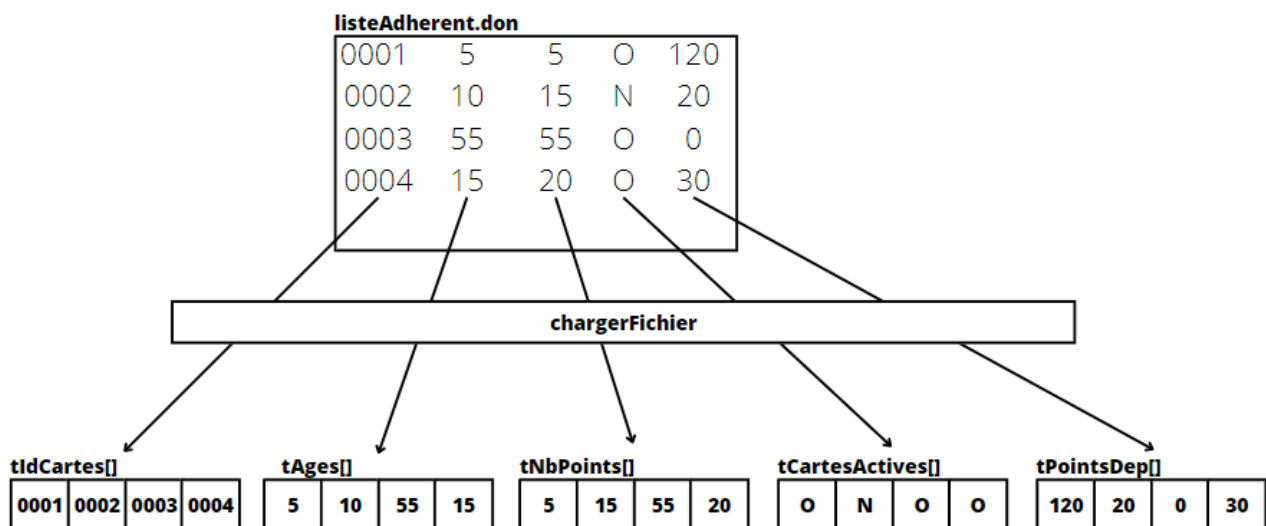
Le fichier listeAdherents.don est composé :

- d'une colonne qui contient l'identifiant d'un adhérent
- d'une tabulation ;
- d'une colonne qui contient l'âge de l'adhérent ;
- d'une tabulation ;
- d'une colonne qui contient le nombre de points de l'adhérent ;
- d'une tabulation ;
- d'une colonne qui contient O si la carte est activée et N si la carte est désactivée ;
- d'une tabulation ;
- d'une colonne qui contient le nombre de points dépensé par l'adhérent ;

À l'ouverture de l'application, la fonction menuPrincipal est lancée. Elle appelle la fonction chargerFichier donc le fichier listeAdherents.don est ouvert, les données de chaque colonne sont enregistrées dans le tableau adéquat :

- La première colonne est enregistrée dans le tableau tIdCartes[] ;
- La deuxième colonne est enregistrée dans le tableau tAges[] ;
- La troisième colonne est enregistrée dans le tableau tNbPoints[]. À cette valeur, est ajouté des points de bonus par rapport à l'âge et au nombre de points dépenser par l'adhérent grâce à l'appel de la fonction pointsBonus ;
- La quatrième colonne est enregistrée dans le tableau tCartesActives[] ;
- La cinquième colonne est enregistrée dans le tableau tPointsDep[] ;

Et le fichier listeAdherents.don est fermé.



Ensuite, elle affiche un menu qui permet d'accéder à deux autres menus. Si le menu des adhérents est sélectionné en tapant le chiffre 1, la fonction `menuAdherents` est appelée. Cette fonction permet d'afficher à un menu qui permet lui-même d'accéder aux fonctions concernant les adhérents.

Si la fonctionnalité Ajouter un adhérent est sélectionné en tapant le chiffre 1, la fonction `ajouterAdherent` est appelée. Elle commence par vérifier s'il y a assez de place dans les tableaux pour ajouter un adhérent. Ensuite, elle demande l'identifiant que l'adhérent souhaite avoir. La question est posée tant que l'identifiant n'est pas compris entre 0 et le nombre maximum d'adhérents, et tant que l'identifiant n'existe déjà. Ce dernier cas est vérifié avec l'appel de la fonction `rechercherAdherent` et un passage par adresse. Ensuite, la fonction demande l'âge de l'adhérent et repose la question tant que l'âge est inférieur à 0. Puis, le nombre de points à ajouté sur la carte est demandé jusqu'à ce que ce nombre soit supérieur à 0. À chaque question posée, si la réponse est -1, on sort de la fonction et on retourne dans la fonction `menuAdherents`. Ensuite, on décale des tableaux à droite à partir de l'indice où doit être inséré l'adhérent récupéré grâce à l'entier renvoyé par la fonction `rechercherAdherent` pour que les tableaux restent triés en fonction de l'adhérent. Et on insère les valeurs dans chaque tableau au bon indice. Ensuite, elle recommence toutes les actions depuis le début jusqu'à ce que l'utilisateur entre -1.

Si la fonctionnalité Alimenter une carte est sélectionné en tapant le chiffre 2, la fonction `alimenterCarte` est appelée. Elle demande l'identifiant de l'adhérent. Elle appelle la fonction `rechercherAdherent` est vérifie si l'adhérent existe déjà grâce à un passage par adresse. Tant que l'identifiant n'est pas trouvé, la question est posée. Ensuite, le nombre de points souhaité est demandé jusqu'à ce que ce chiffre soit supérieur à 0. Ensuite, le nombre de points à ajouter est ajouté dans le tableau `tNbPoints[]` à l'indice récupérer grâce à la fonction `rechercherAdherent`. À chaque question posée, si la réponse est -1, on sort de la fonction et on retourne dans la fonction `menuAdherents`. Puis, les actions sont recommencées depuis le début jusqu'à ce que l'utilisateur entre -1.

Si la fonctionnalité Activer/Désactiver une carte est sélectionné en tapant le chiffre 3, la fonction `changerEtatCarte` est appelée. Elle demande l'identifiant de l'adhérent. Elle appelle la fonction `rechercherAdherent` est vérifie si l'adhérent existe déjà grâce à un passage par adresse. Tant que l'identifiant n'est pas trouvé, la question est posée. Ensuite, elle donne l'état de la carte de l'adhérent grâce à l'indice récupéré avec la fonction `rechercherAdherent`. Si la carte est activée, elle demande s'il faut la désactiver et si elle est désactivée, elle demande s'il faut l'activer. Elle repose la question tant que la réponse n'est pas o, O, n ou N. Puis, elle change d'état de la carte au bon indice. À chaque question posée, si la réponse est -1, on sort de la fonction et on retourne dans la fonction `menuAdherents`. Puis, les actions sont recommencées depuis le début jusqu'à ce que l'utilisateur entre -1.

Si la fonctionnalité Supprimer un adhérent est sélectionné en tapant le chiffre 4, la fonction `supprimerAdherent` est appelée. Elle demande l'identifiant de l'adhérent. Elle appelle la fonction `rechercherAdherent` est vérifie si l'adhérent existe déjà grâce à un passage par adresse. Tant que l'identifiant n'est pas trouvé, la question est posée. Elle demande si l'adhérent doit vraiment être

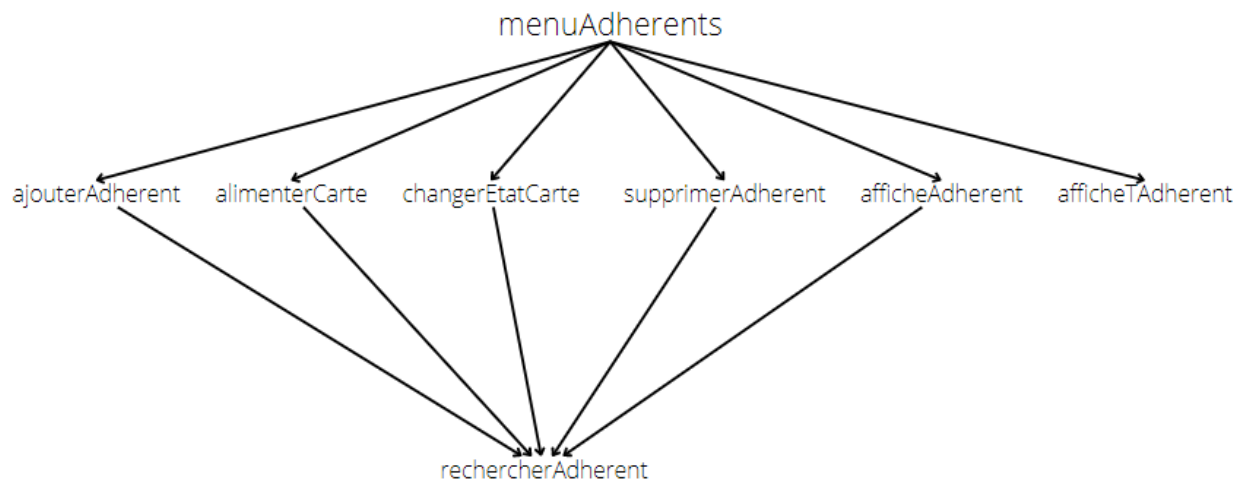
supprimé. Elle repose la question tant que la réponse n'est pas o, O, n ou N. Ensuite les tableaux concernant les adhérents sont décalé à gauche à partir de l'indice de l'adhérent à supprimer récupérer grâce à la fonction `rechercherAdherent` pour que les données de l'adhérent soient supprimé, mais que le tableau reste trié. À chaque question posée, si la réponse est -1, on sort de la fonction et on retourne dans la fonction `menuAdherents`. Puis, les actions sont recommencées depuis le début jusqu'à ce que l'utilisateur entre -1.

Si la fonctionnalité Changer l'âge d'un adhérent est sélectionné en tapant le chiffre 5, la fonction `changerAgeAdherent` est appelé. Elle demande l'identifiant de l'adhérent. Elle appelle la fonction `rechercherAdherent` est vérifie si l'adhérent existe déjà grâce à un passage par adresse. Tant que l'identifiant n'est pas trouver, la question est reposé. Elle demande le nouvel âge de l'adhérent et elle repose la question jusqu'à ce que l'âge soit supérieur ou égal à 0. À chaque question posée, si la réponse est -1, on sort de la fonction et on retourne dans la fonction `menuAdherents`. Puis, les actions sont recommencées depuis le début jusqu'à ce que l'utilisateur entre -1.

Si la fonctionnalité Afficher les informations d'un adhérent est sélectionné en tapant le chiffre 6, la fonction `afficheAdherent` est appelé. Elle demande l'identifiant de l'adhérent. Elle appelle la fonction `rechercherAdherent` est vérifie si l'adhérent existe déjà grâce à un passage par adresse. Tant que l'identifiant n'est pas trouver, la question est reposé. Ensuite, elle donnait toutes les informations concernant l'adhérent en lisant les tableaux avec l'indice correspondant à l'adhérent récupéré grâce à la fonction `rechercherAdherent`. À chaque question posée, si la réponse est -1, on sort de la fonction et on retourne dans la fonction `menuAdherents`. Puis, les actions sont recommencées depuis le début jusqu'à ce que l'utilisateur entre -1.

Si la fonctionnalité Afficher les informations de tous les adhérents est sélectionné en tapant le chiffre 7, la fonction `afficheTAdherent` est appelé. Elle parcourt tous les indices dans l'ordre de 0 au nombre d'élément présent dans les tableaux. Et elle affiche les informations de tous les tableaux pour chaque indice. Puis, quand l'affichage est terminé, elle retourne dans la fonction `menuAdherent`.

Si la fonctionnalité Quitter est sélectionnée en tapant le chiffre 9, on sort de la fonction et on retourne dans la fonction `menuPrincipale`.



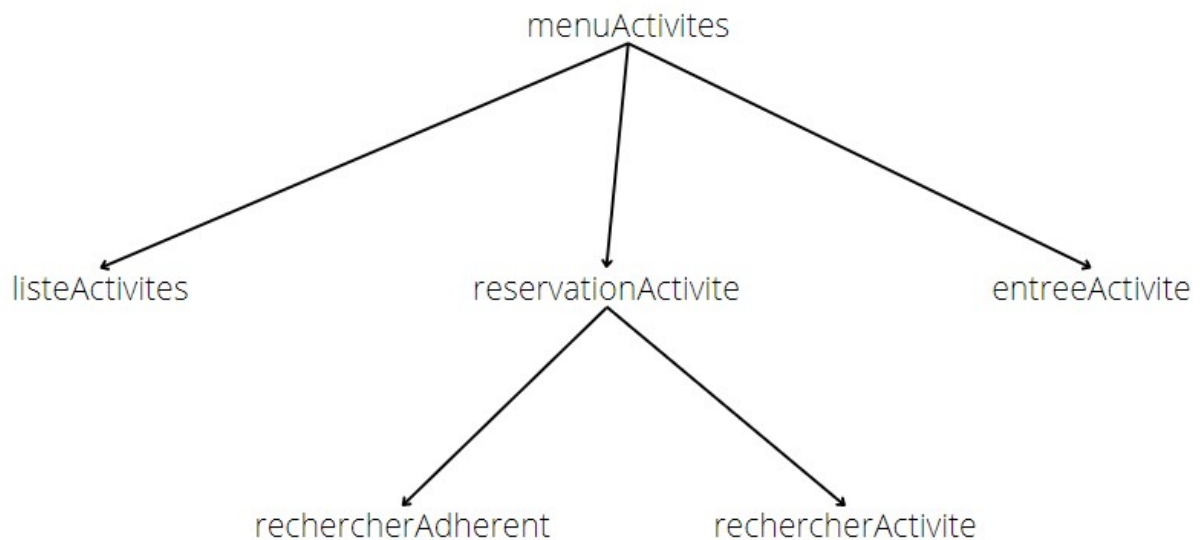
Dans la menuPrincipal, si le menu des activités est sélectionné en tapant le chiffre 2, la fonction menuActivites est appelé. Cette fonction permet d'afficher à un menu qui permet lui-même d'accéder aux fonctions concernant les activités. Puis, quand l'affichage est terminé, elle retourne dans la fonction menuActivites.

Si la fonctionnalité Voir la liste, des activités disponibles est sélectionné en tapant le chiffre 1, la fonction listeActivites est appelé. Elle affiche ligne par ligne les informations concernant une activité : son numéro, son nom et le coût de l'activité.

Si la fonctionnalité Réserver une activité est sélectionné en tapant le chiffre 2, la fonction reservationActivite est appelé. Elle demande l'identifiant de l'adhérent. Elle appelle la fonction rechercherAdherent est vérifie si l'adhérent existe déjà grâce à un passage par adresse. Tant que l'identifiant n'est pas trouver, la question est reposé. Ensuite, elle vérifie si la carte de l'adhérent est active et si l'adhérent n'a pas fréquenté le centre grâce aux tableaux concernant les adhérents et l'indice qui correspond aux informations de l'adhérent récupérer grâce à la fonction rechercherAdherent. Puis, elle répète au maximum trois fois les actions suivantes. Elle appelle la fonction listeActivites. Elle demande le numéro de l'activité à réserver. Elle répète la question jusqu'à ce que le numéro soit trouvée dans le tableau tActivites grâce au passage par adresse de la fonction rechercherActivite. Elle vérifie ensuite le nombre de points de l'adhérent trouvé dans le tableau du nombre de points, soit supérieur ou égal au nombre de points que coûte l'activité trouvée dans le tableau de coût d'une activité. Si ce n'est pas le cas, elle reprend les étapes à partir de l'appel à la fonction listeActivites. Sinon si, le nombre d'entrées pour cette activité est inférieur ou égale à 50 grâce à la position récupérée avec la fonction rechercherActivite. Si c'est le cas, elle reprend les étapes à partir de l'appel à la fonction listeActivites. Sinon, elle enlève le nombre de points que coût l'activité à l'adhérent, elle les ajoute aux nombre de points dépensés, elle note que l'adhérent à fréquenté le centre et elle ajoute une entrée à l'indice de l'activité concerné. À chaque question posée, si la réponse est -1, on sort de la fonction et on retourne dans la fonction menuAdherents.

Si la fonctionnalité Afficher le nombre d'entrées par activités, est sélectionné en tapant le chiffre 3, la fonction `entreeActivite` est appelé. Elle parcourt tous les indices dans l'ordre de 0 au nombre d'élément présent dans les tableaux. Et elle affiche les informations de tous les tableaux pour chaque indice. Puis, quand l'affichage est terminé, elle retourne dans la fonction `menuActivites`.

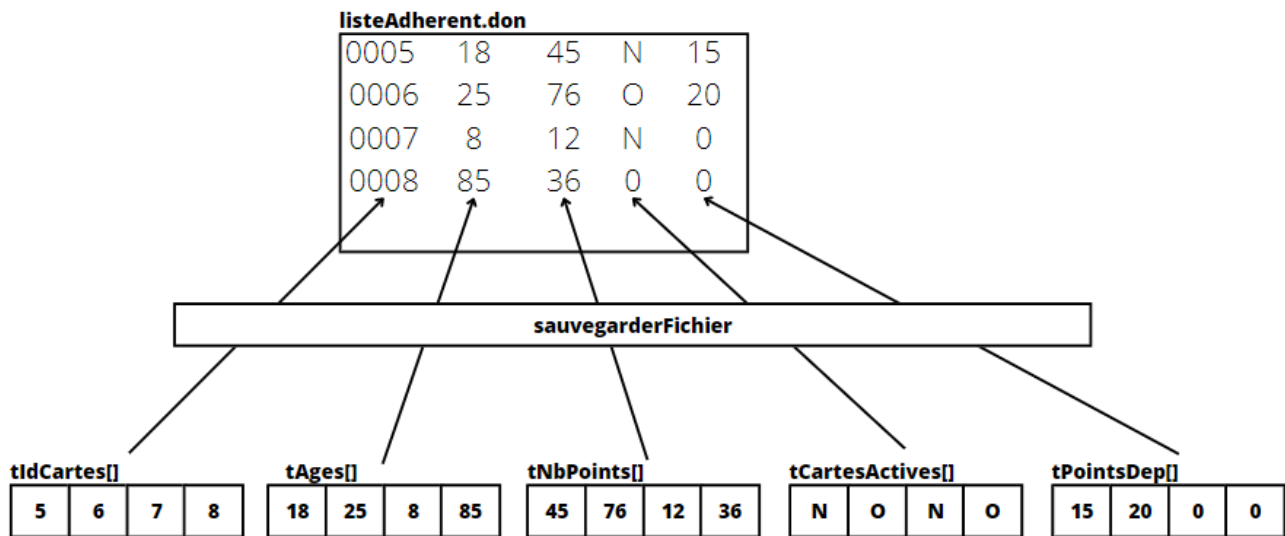
Si la fonctionnalité Quitter est sélectionnée en tapant le chiffre 9, on sort de la fonction et on retourne dans la fonction `menuPrincipale`.



Dans la `menuPrincipale`, si Quitter est sélectionné en tapant le chiffre 9, la fonction `sauvegarderFichier` est lancée donc le fichier `listeAdherents.don` est ouvert et écrasé, les données de chaque tableau sont enregistrées dans la colonne adéquate :

- Le tableau `tIdCartes[]` est enregistré dans la première colonne (autant de 0 que nécessaire sont ajouté à l'identifiant pour donner un nombre à 4 chiffres) ;
- Le tableau `tAges[]` est enregistré dans la deuxième colonne ;
- Le tableau `tNbPoints[]` est enregistré dans la troisième colonne ;
- Le tableau `tCartesActives[]` est enregistré dans la quatrième colonne ;
- Le tableau `tPointsDep[]` est enregistré dans la cinquième colonne.

Le fichier `listeAdherents.don` est fermé et on sort de la fonction.



Contraintes

Le développement doit être réalisé en C avec les notions apprises en algorithmique.

Résultats attendus

Résultats attendus

Dépôt avec le cahier des charges, le code de l'application, la documentation, les traces d'exécution des jeux d'essais.

Partage du travail

Les fonctions concernant les adhérents ont été faites ensemble. Anthony a été chargé des fonctions des menus et des jeux essais ; et Léana a été chargée de faire les fonctions des activités et le cahier des charges. Chacun a fait le brief des fonctions qu'ils ont réalisé.

Exigences

Le dépôt doit être rendu le 11 novembre 2022.