

Rapport de la Solution

« Club de tennis »

Programmation Orientée Objet

ESILV - 2020

Sommaire

I. Explication Diagramme UML.....	3
1. Les Membres/Coachs/Staffs.....	3
2. Les Evènements.....	3
3. Le Club.....	4
II. Explication de l'IHM.....	5
1. Section: Liste des Membres (Tous).....	5
2. Section: Ajouter membre.....	5
3. Section Compétition.....	5
4. Section Tournoi.....	6
III. Bilan.....	6

I. Explication Diagramme UML

La plupart des collections utilisées sont des **SortedSet**(des ensembles) afin d'éviter les doublons. Ce qui veut dire que de nombreux objets implémentent l'interface **Comparable**(**Person**, **TeamMatch**, **Event...**), des classes implémentent également **Comparer** ont été créé notamment pour les filtres de la classe **ListMemberPage**.

1. Les Membres/Coachs/Staffs

Nous avons créé la classe intermédiaire **TennisPlayer**, pour permettre dans le futur, la création de la classe **Opponent**, qui sera un adversaire de Tennis sans être membre du club. C'est-à-dire qui héritera de **TennisPlayer**, mais pas de **Membre**. Cela dans le but de pouvoir modéliser des adversaires lors de la création de TeamMatch (*Rencontre entre deux binômes de deux clubs différents*).

Ici, tous les Coachs ainsi que les Staffs sont automatiquement des **Membre** et gratuitement. Les Coachs possèdent un attribut enum(**Status**) pour savoir s'ils sont indépendants ou salariés.

De même, les Membre possèdent un attribut **RegistrationType** pour savoir s'ils ont une adhésion de compétition ou de loisir seulement. Les membres possèdent un ensemble d'IDs de **PaidEvent** pour savoir quels événements payants ont été payé par le membre en question.

2. Les Evènements

L'interface **Event** permet la modélisation d'évènements qui sont **MemberReservation** (*Réservation d'un court de tennis de deux membres*), **TennisTraining**(*Un cours de tennis*), **FamilyTournament**(*Les tournois amicaux au sein du club*), **PaidEvent**(*Evènements payants, classe abstraite, mère de **Competition** et **TennisClinic**(Stage de tennis pour les jeunes)*)).

Les évènements reliés à l'objet **TennisCourt**, sont des évènements qui ont besoin de 1 à n terrains de tennis, cela permet de vérifier que deux évènements ne peuvent pas avoir lieu sur le même terrain à la même date.

Les dates de début et de fin d'une **Competition** sont calculés automatiquement en fonction des dates des **TeamMatch** qui y sont contenues.

3. Le Club

Ici, nous avons créé un objet club dans l'hypothèse que dans le futur, cette application ressemblera de nombreux **Clubs**. Dans notre cas présent, nous avons créé une seule variable globale **Club** dans la classe **Club** elle-même(*Pas une très bonne pratique, mais simplifiant notre cas d'utilisation*).

Le club possède un **ISet** de **Member**(*membres simples uniquement*), un **ISet** de **Coach**, un **ISet** de **Staff** ainsi qu'un **ISet** pour chacun des types **d'Event**. Cette centralisation des objets du club permet la

II. Explication de l'IHM

Ce projet ne possède que deux **Windows** possible, celle principale et celle pour ajouter des membres dans une compétition ou un tournoi. De plus, le package Material Design a été installé pour l'aspect visuel plus agréable de la solution.

1. Section: Liste des Membres (Tous)

Sur cette page, s'affiche la liste de tous les membres du club incluant le personnel. Les radios buttons permettent de trier et ordonner cette liste en fonction des besoins de l'utilisateur. Les boutons « **Modifier** » contenues dans le GridData permettent de modifier le membre correspondant en redirigeant l'utilisateur vers la page « Ajouter un Membre ». Ainsi la page s'auto-remplie avec les informations du membre et quand l'utilisateur valide les modifications apportées, le **Membre** se met à jour au lieu de s'ajouter à la liste des membres. Le bouton «Supprimer Sélectionnés» permet de supprimer tous les membres sélectionnés dans le DataGrid.

2. Section: Ajouter membre

Cette page permet d'ajouter ou de modifier un membre. Si la page a été appelée pour une modification, le champs «**Type**» du membre ne peut pas être modifié(**Membre**, **Coach** ou **Staff** reste figé). De nombreuses vérifications ont lieu lorsque l'utilisateur veut valider le membre (tous les champs sont bien remplis, date de naissance cohérente...), de plus la **TextBox** « CodePostale » ne peut être remplie qu'avec des chiffres. Des éléments de la page sont cachés ou visible en fonction du type du membre qui y est ajouté et/ou en fonction des radios buttons de la page.

3. Section Compétition

La page qui liste les compétitions fonctionne tout comme la page qui liste les membres du club. Il est possible de supprimer les compétitions sélectionnées d'en ajouter et/ou modifier.

Lorsque l'on Ajoute/Modifie une compétition, il est possible d'y ajouter des **TeamMatch**(*Matches de binômes*). Lorsque l'utilisateur voudra choisir les joueurs, il devra choisir une date avant cela. Ainsi lorsqu'il choisira le premier joueur, la page n'affichera que les membres disponibles ce jour-là. Lorsqu'il choisira le deuxième joueur la page n'affichera que les membres disponibles ce jour-là et du même niveau que le premier joueur sélectionné. Sinon ici aussi, tous les tests nécessaires ont lieu sur les champs lorsque l'utilisateur clique sur le bouton « Valider ». Nous avons pris l'initiative de permettre aux Hommes et Femmes de jouer dans un même binôme car les matches mixtes existent dans la vraie vie.

4. Section Tournoi

La page liste des tournois permettent également de lister, d'ajouter et de modifier un tournoi. Dans la page Ajouter/Modifier un tournoi, l'utilisateur doit valider une date avant d'y ajouter des joueurs dans le but de savoir quels joueurs peuvent participer à cet événement ce jour même. Si l'utilisateur veut modifier la date du tournoi, la liste des joueurs est remise à zéro car il n'est plus possible de certifier que les membres pourront être présent durant la nouvelle date donnée.

III. Bilan

Un effort particulier a été donné dans la gestion des données de notre solution. En effet, un membre ne peut pas être disponible sur deux événements en même temps dans notre programme. Un membre non compétiteur ne peut pas être ajouté à une compétition. Un tournoi ne peut pas être ajouté si sa liste de participants ne possède pas au moins 2 coachs. Un membre ne peut pas être ajouté deux fois dans un même événement. De plus la liste des membres peut être triée selon de nombreux critères superposables si ceux-ci sont indépendants. La Window de sélection des membres permet de filtrer les membres en y affichant que ceux disponibles à une date donnée avec possibilité d'y ajouter un filtre en fonction du niveau du joueur nécessaire.