

Rapport de réalisation

SAÉ 2.01-2.07

Conception d'une application pour simuler les
jeux olympiques 2024



Organisé par le BUT informatique de l'IUT d'Orléans

Réalisé par Clermont Shanka, Bastien Monet,
Clément renaudin, «Imad Assou» et Malo Dantec

RAPPORT DE RÉALISATION

Jeux Olympiques et Paralympique de Paris 2024



Table des matières

1. Introduction	4
2. Analyse UML (Diagrammes et cas d'utilisation, organisation BD)	4
2.1 Diagramme de cas d'utilisation	5
2.2 Diagramme UML/ Diagramme de classe	6
2.3 Diagramme de séquence	11
2.4 Base de données	14
3. Interface Homme Machine	15
3.1 Maquette	16
3.2 Interface graphique (IHM) réalisée	19
4. Implémentation	21
5. Organisation du travail de groupe	23
6. Bilan du groupe et bilans individuels	24
6.1 Conclusion	24
6.2 Bilans personnel	25
7. Mentions légales :	28

Lien dépôt GitHub :

<https://github.com/Emir-de-la-moquette/SAE-2.01-2.07-java.git>

1. Introduction

Lors de cette première année au BUT, dans le cadre de la SAE 2.01, il nous a été demandé de réaliser une application en Java qui prend en charge les résultats d'épreuves sportives variées de jeux Olympiques.

Celle-ci doit permettre à ces utilisateurs, tel que des journalistes, un accès aux résultats des JO. Ces résultats peuvent prendre en compte différentes épreuves de différents sports, il doit également être possible d'accéder au statistique des athlètes participants.

L'application nécessitera également des solutions d'administration pour pouvoir obtenir de nouvelles données et faire participer les Athlètes au JO.

Nous sommes un groupe de 4 pour réaliser cette SAE, Clément Renaudin, Bastien Monet, Imad Assou, Shanka Clermont.

Nous nous sommes organisés pour développer les fonctionnalités de l'application ainsi que la réalisation d'une interface homme machine pour pouvoir communiquer avec celle-ci.

Nous présenterons tout d'abord, l'analyse d'UML de notre application en créant un diagramme de cas d'utilisation, un diagramme séquence et un diagramme de classes et nous établirons un schéma MCD qui décrira notre base de données utilisée.

Puis nous expliquerons notre démarche et les calculs utilisés lors de l'implémentation de notre modèle de base de données.

Ensuite il y aura une analyse détaillée sur le design et les fonctionnalités de l'IHM. Cette partie sera constituée de la maquette de l'application qui apportera des informations supplémentaires quant à l'ergonomie et la facilité de l'utilisation de l'application.

De plus, nous mettrons en avant l'organisation et le travail réalisé par séance que ce projet à demander pour voir le jour. Et enfin, chacun des membres du groupe ont rédigé leur bilan personnel qui décrit leur ressenti sur cette SAE.

2. Analyse UML (Diagrammes et cas d'utilisation, organisation BD)

Dans ce chapitre, nous allons voir en premier le diagramme de cas d'utilisation avec un résumé détaillé sur les fonctionnalités principales de notre application.

Pour donner suite à cela, la seconde partie va présenter notre diagramme de classe composé d'une représentation du diagramme de classe, une analyse détaillée des classes et de la représentation des matches.

Pour finir notre dernière partie s'occupe du diagramme de séquence, représentent les fonctionnalités de l'application en général.

2.1 Diagramme de cas d'utilisation

Un diagramme de cas d'utilisation, c'est comme une carte qui montre comment les utilisateurs vont interagir avec un logiciel. Il représente les différentes actions que les utilisateurs peuvent effectuer et comment le système réagit à ces actions. Cela aide les développeurs à comprendre ce que le logiciel doit faire pour satisfaire les besoins des utilisateurs.

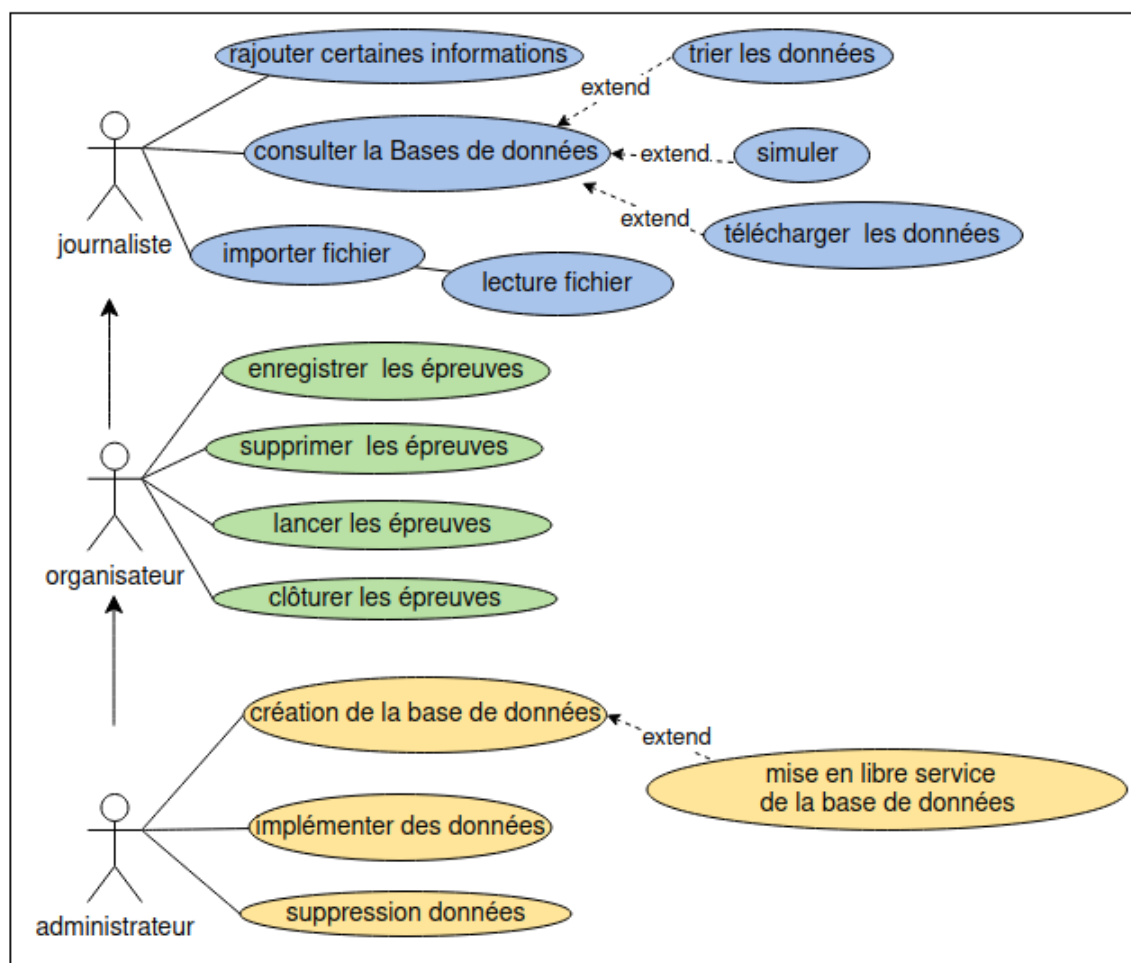


Figure 1 : Diagramme de cas d'utilisation

Le diagramme (figure 1) décrit les fonctionnalités principales de notre application

Nous représentons trois rôles dans ce diagramme :

- L'utilisateur lambda comme un journaliste qui ne peut que consulter les résultats et le classement des Jeux olympiques, ainsi que de les télécharger pour eux.

- L'Administrateur qui s'occupera de gérer les données stockées dans la base de données. L'administrateur peut donc supprimer ou ajouter des données par exemple des matchs, des équipes, des Athlètes, et même des sports. De plus, l'administrateur pourra trier les données via des requêtes de la base.

Pour finir, les Organiseurs pourront gérer les différentes épreuves ainsi que les différents sports, en ajouter ou en retirer.

2.2 Diagramme UML/ Diagramme de classe

Un diagramme UML est comme un plan pour construire une maison, mais pour les logiciels. Il utilise des dessins simples pour montrer comment les différentes parties du logiciel fonctionnent ensemble, comme les pièces d'un puzzle. Cela aide les programmeurs à comprendre et à créer des logiciels plus facilement.

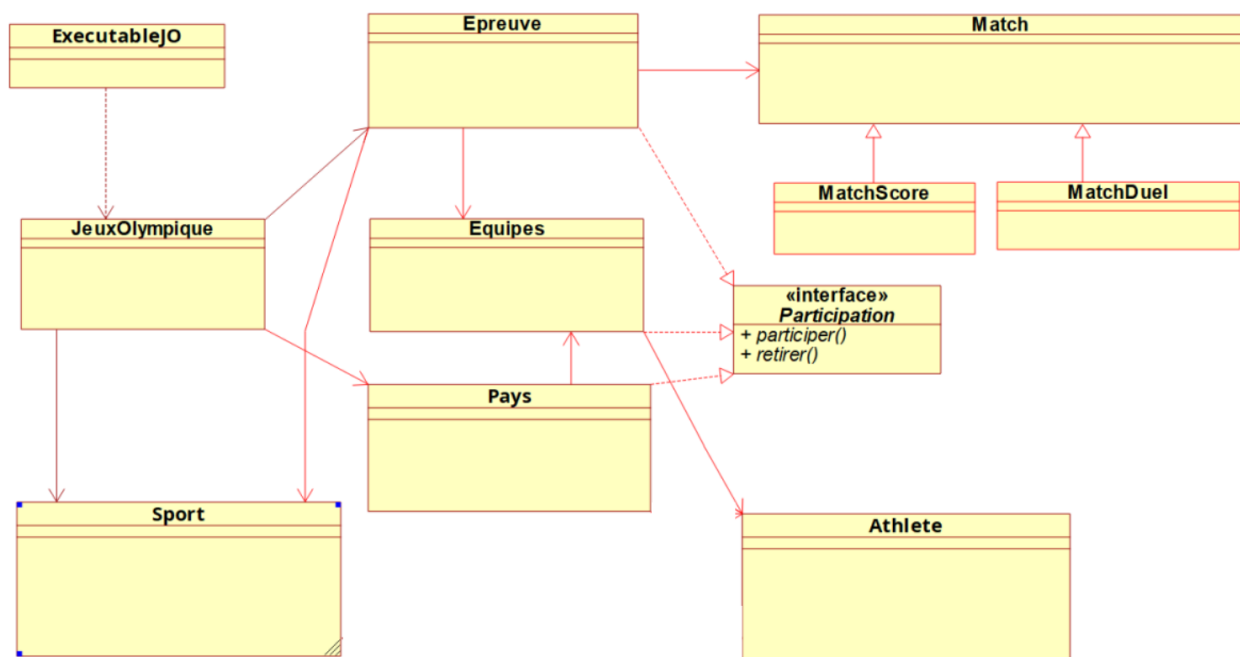


Figure 2 : Diagramme UML de classe complet

Le diagramme de classe sert à modéliser les objets qui constituent le système. Nous avons ainsi établi 9 classes, dont 2 classes associées à la classe match, ainsi que 2 interfaces : une interface pour gérer les données Json, et une interface comparator.

- Ainsi nous avons la classe JO, on précise son année et son lieu. Cette classe peut accéder aux sports. C'est une classe très importante, car celle-ci possède la méthode simulJO pour jouer l'entièreté des matchs de toutes les épreuves. De plus la méthode

classement, permettra de savoir quel pays possède les équipes ayant le plus de médaille.

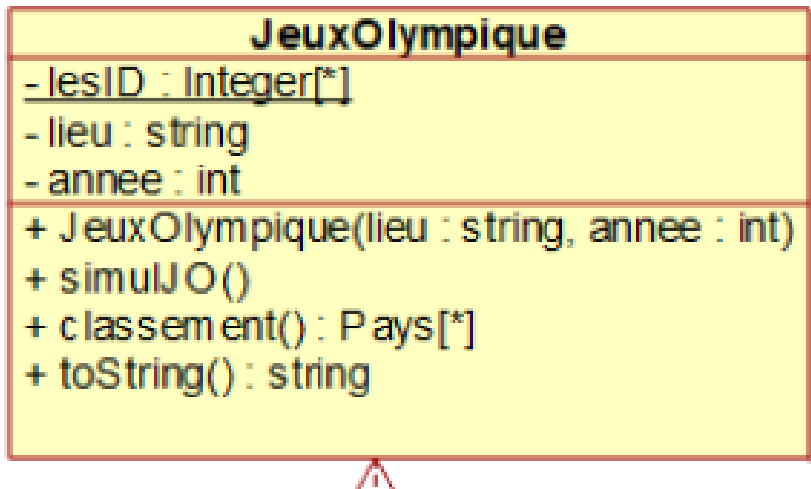


Figure 3 : Diagramme UML classe JeuxOlympique

- Un sport possède un nom, le nombre de joueurs nécessaire qu'une équipe doit posséder pour pouvoir jouer, ainsi que des coefficients évaluant les aptitudes les plus importantes à avoir pour être bon dans un tel sport. de plus pour obtenir des résultats encore plus cohérent, on rajoute les point minimum que l'on peut faire, les point max, le record du monde

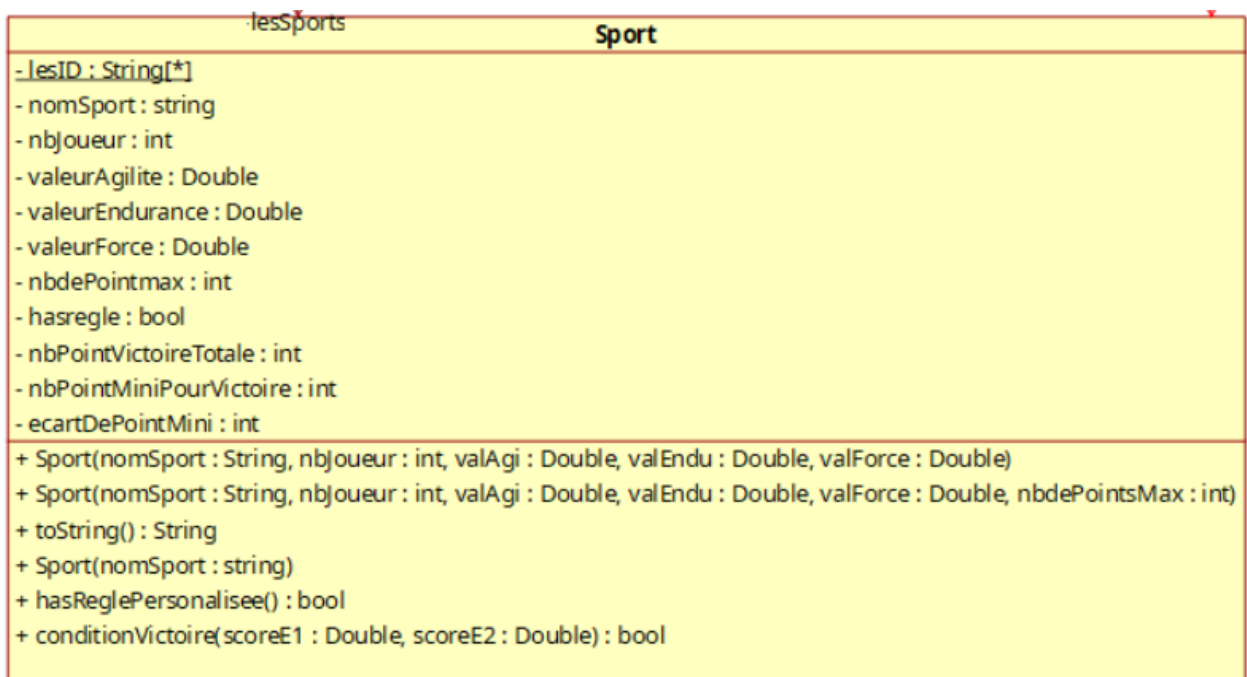


Figure 4 : Diagramme UML classe Sport

- Les athlètes ont un nom, un prénom, un sexe, ainsi que des statistiques d'aptitudes, les athlètes connaissent les sports.

-Un pays possède des athlètes, c'est dans cette classe que l'on va évaluer le nombre de médailles qu'un pays va avoir.

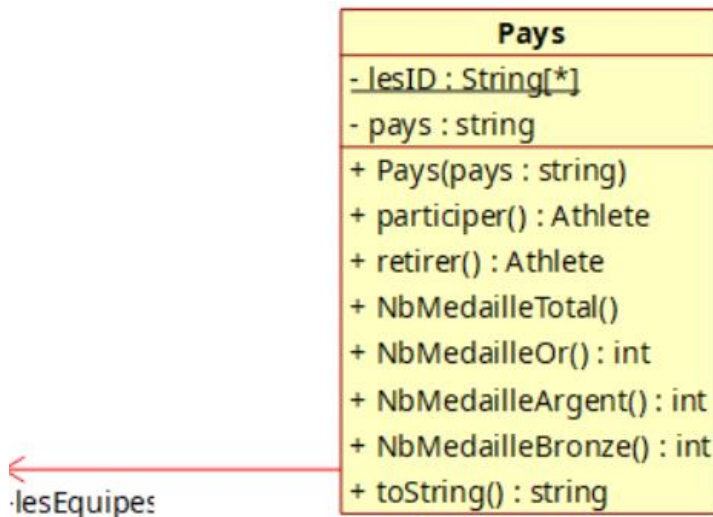


Figure 5 : Diagramme UML classe Pays

Nous avons ensuite décidé de rajouter une classe équipe qui contient des athlètes, pour simplifier et limiter la structure du diagramme, on estime qu'il est possible qu'une équipe ne possède qu'un seul athlète, et de le faire jouer en solo. C'est aussi à partir d'ici que va se faire l'ajout de médailles.

Une fois que l'on a nos équipes, celles-ci sont intégrées à l'épreuve grâce à la méthode « Participer ». Ces épreuves ont des caractéristiques : leur nom, le sexe de l'athlète admis, leur catégorie (la catégorie peut être à null si il n'y en a pas) et leur type d'épreuve (Duel ou à score (*voir page 9*)). Une fois que les athlètes se sont inscrits, la méthode "lanceEpreuve" lance la succession de matchs qui déterminera le classement.

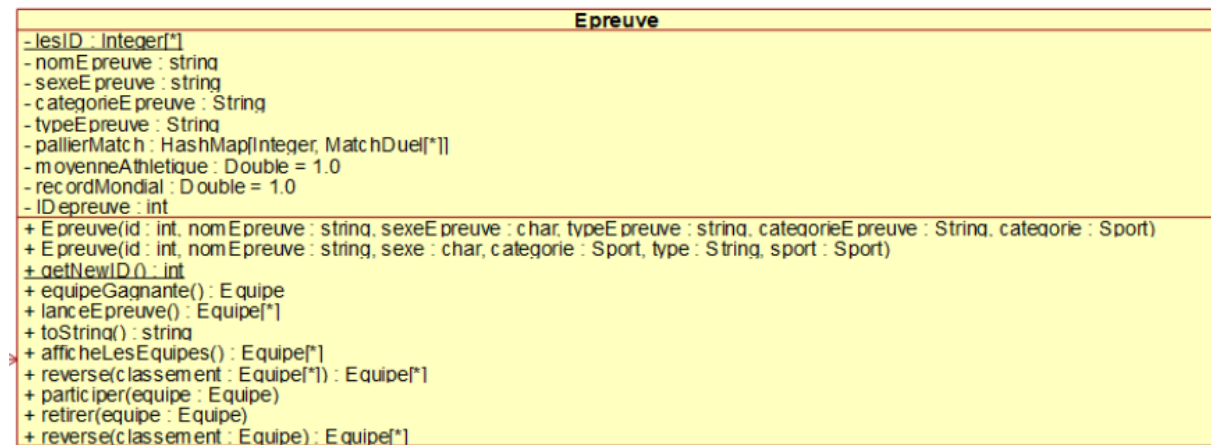


Figure 6 : Diagramme UML classe Epreuve

Nous avons implémenté 2 types d'épreuves, les épreuves de type "Score" et les épreuves de type "Duel". Les épreuves de type "Score" sont les plus basiques : on exécute une succession de match qui contient la liste de participants toujours en course, une fois les matchs effectués, on retire les 3 plus mauvaises équipes et on les intègre en bas du classement et ainsi de suite, jusqu'à obtenir un classement du premier au dernier.

Nous avons ajouté une autre méthode pour coller au réalisme des JO avec les matchs "Duel", ce qui va réaliser une succession de match entre 2 équipes et en réaliser un classement final, vous pouvez suivre la méthode d'implémentation avec la *figure 3* ci-dessous. Pour réaliser ces tournois nous avons pensé à ceci : on crée des niveaux, 1, 2, 3 etc... Au tout début chaque équipe se trouve au niveau 0, puis ils réalisent des matchs entre eux, le gagnant passe au niveau supérieur et le perdant reste au même niveau. Les équipes ne peuvent qu'affronter d'autres équipes présentes au même niveau qu'eux, cela signifie que parfois lors d'une boucle, une équipe seule dans un niveau, n'affronte personne et ne changera pas de niveau. Le classement est établi lorsqu'il ne reste qu'une équipe par niveau et que donc tous les matchs ont été joué.

Exemple épreuve MatchDuel d'une poule de 4					
Niveau					
...					
3					X
2			X	XX	X
1		XX	XX	X	X
0	XXXX	XX	X	X	X
Etape	0	1	2	3	4
<p>X : équipe sans match</p> <p>X : équipe perdante</p> <p>X : équipe gagnante</p>					

Figure 7 : Schéma du fonctionnement du type d'épreuve "Duel".

Chaque match déterminera les scores des athlètes durant des épreuves, ces scores seront les valeurs qui déterminent qui gagnera. La méthode "dérouler Match", établira les équipes participant au match avec leurs scores, puis les placera dans "scoreEquipe" de la classe Équipe, nous pourrons alors établir des vainqueurs, et continuer les matchs jusqu'à obtenir notre classement final.

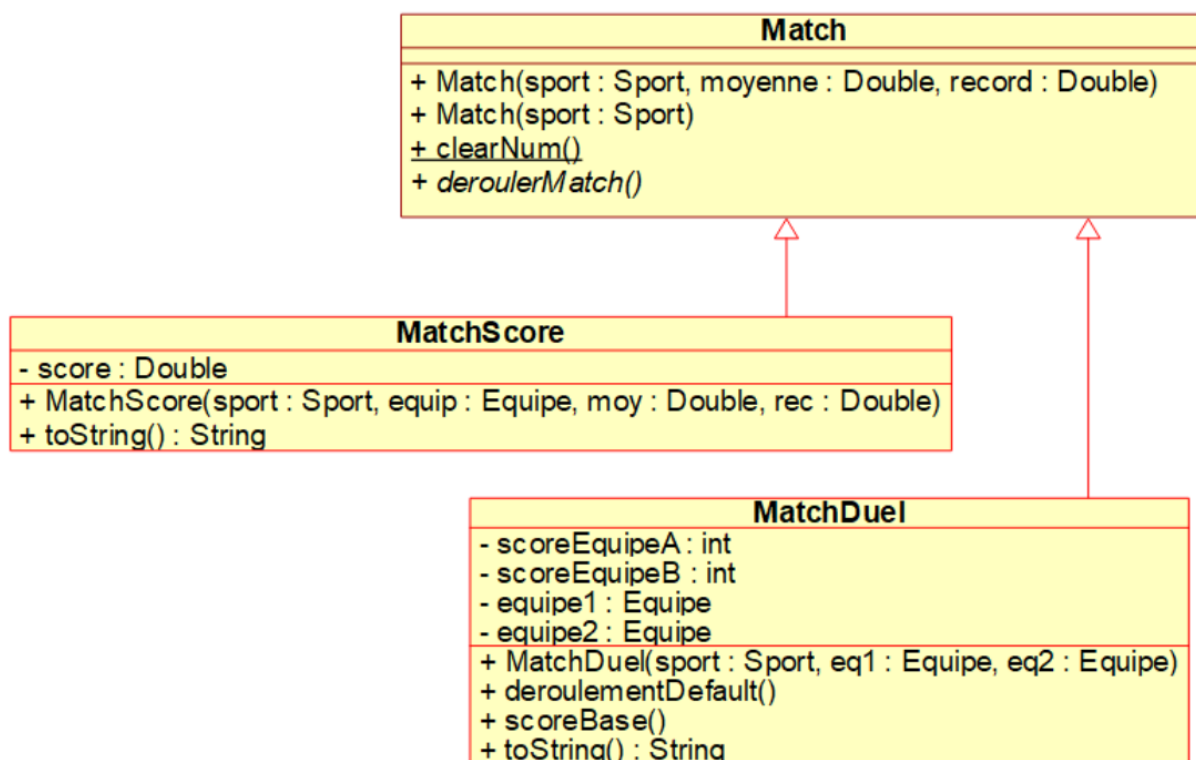


Figure 8 : Diagramme UML classes Match

Chaque match déterminera les scores des athlètes durant des épreuves, ces scores seront les valeurs qui déterminent qui gagnera. La méthode “dérouler Match”, établira les équipes participant au match avec leurs scores, puis les placera dans “scoreEquipe” de la classe Équipe, nous pourrons alors établir des vainqueurs, et continuer les matchs jusqu’à obtenir notre classement final.

Nous pouvons enfin noter l’ajout des getters, setters. Ainsi que des « ajouter », « supprimer » un élément dans les différentes listes utilisées

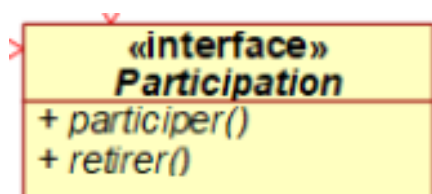


Figure 9 : Diagramme UML interface Participation

Participation est une interface qui permet d’appliquer le fait de faire participer un élément à un groupe ou le retirer, cette dernière est appliquée aux classes Equipe, Pays et Epreuve.

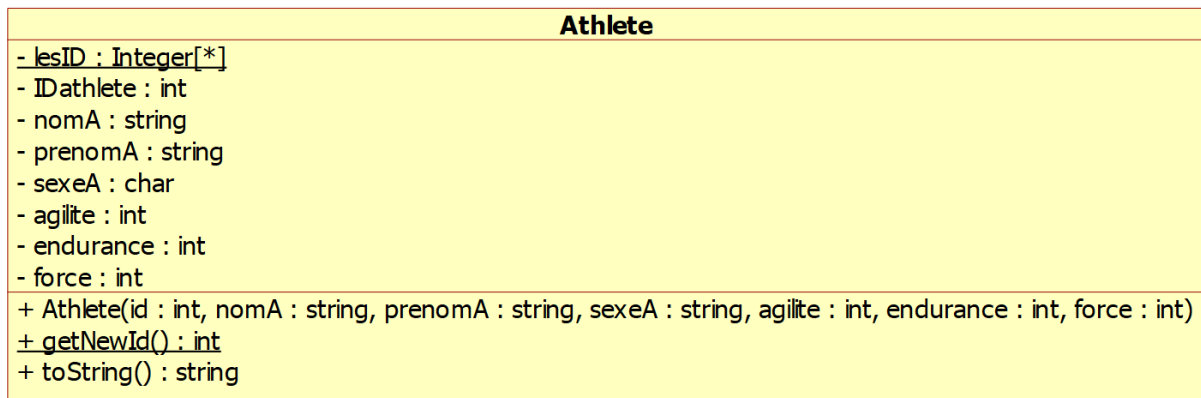


Figure 10 : Diagramme UML classe Athlete

Voici comment sont représentés les athlètes dans notre application, leurs compétences sont représentées par agilité, endurance et force, Ils ont bien évidemment un nom, un prénom et un sexe.

La plupart de ces classes comme représentez dans ces diagramme UML possèdent un Identifiant afin de faciliter leur exploitation.

2.3 Diagramme de séquence

Un diagramme de séquence est une représentation graphique qui illustre les interactions entre les différentes parties d'un logiciel au fil du temps. Chaque étape de la séquence montre quels composants communiquent entre eux et décrit les messages échangés. Ce type de diagramme est essentiel pour visualiser et analyser les flux de communication et les processus opérationnels au sein du système, offrant ainsi une compréhension approfondie de son fonctionnement dans diverses situations.

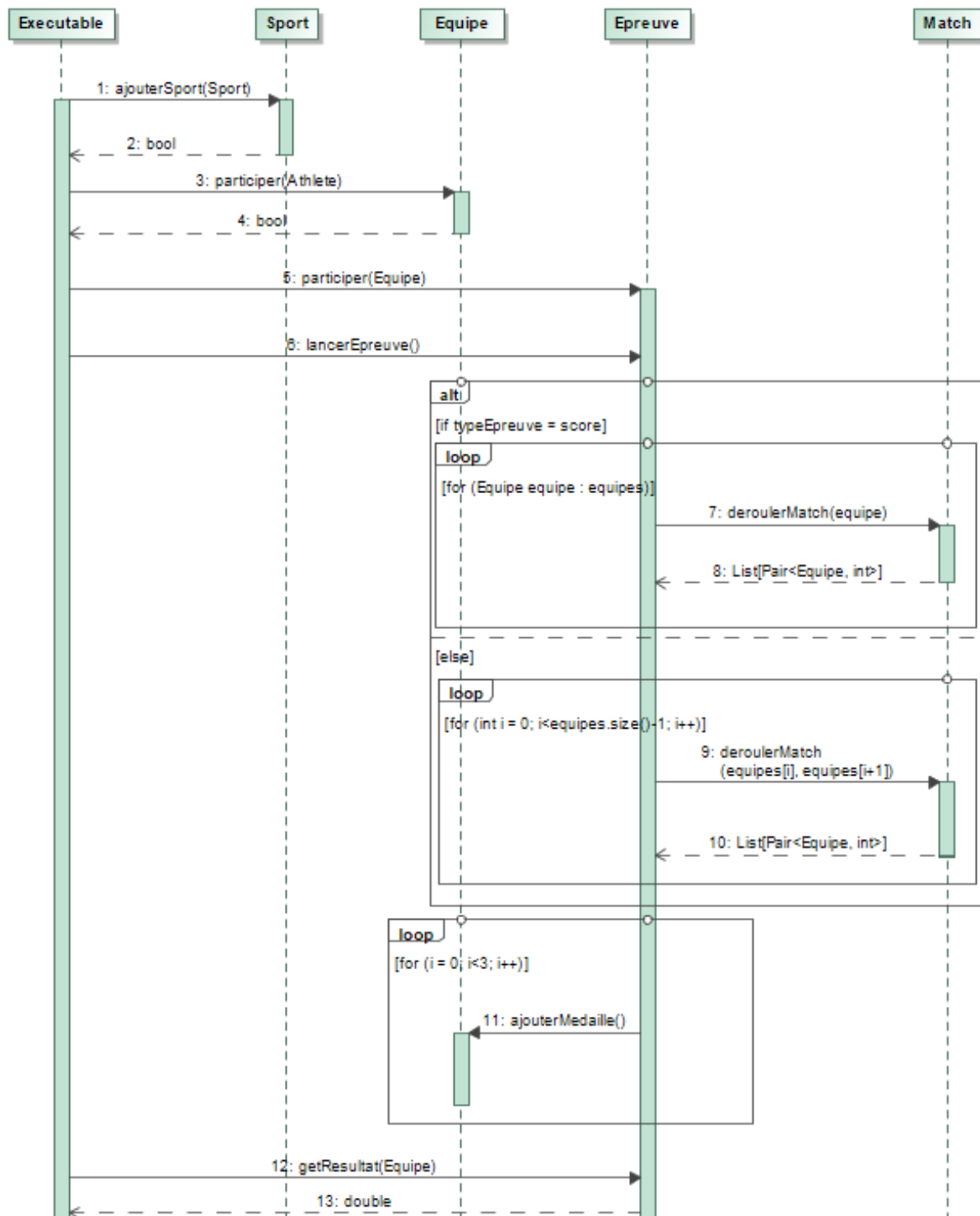


Figure 11 : Diagramme de séquence complet.

Ce diagramme de séquence va représenter les fonctionnalités de l'application en général au niveau de l'exécution des épreuves. Cela permet une meilleure compréhension technique de l'application

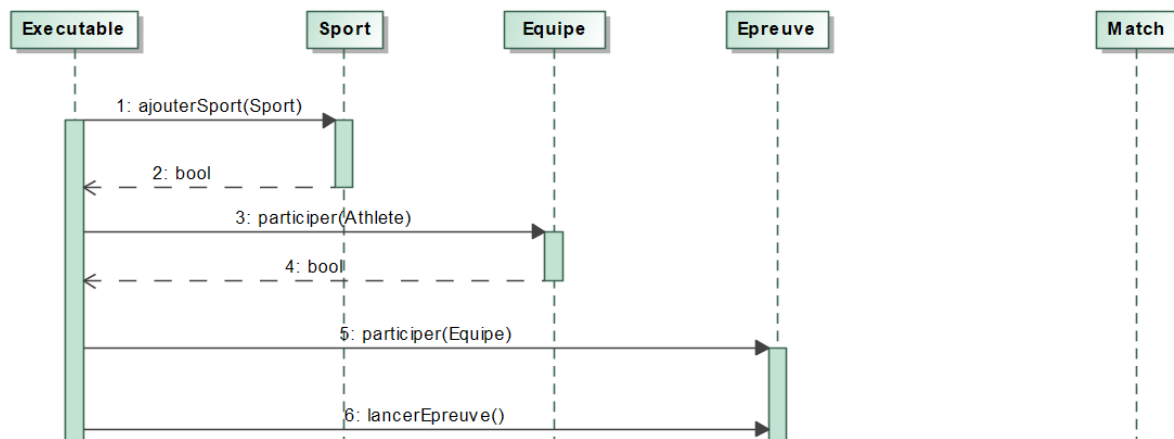


Figure 12 : Partie initialisation du diagramme de séquence.

On commence par l'initialisation de sport. Ils définissent des listes de sport pour la classe Jeux olympiques et Athlète via la méthode « ajouterSport » qui confirme notre action avec un booléen.

Pour donner suite à cela, il y a l'initialisation des classes Pays, Athlète et Équipe. Pays est relié aux athlètes et à l'équipe. Les athlètes vont être ajoutés à une équipe, définit par un pays, un sport et des athlètes par la méthode « participer ».

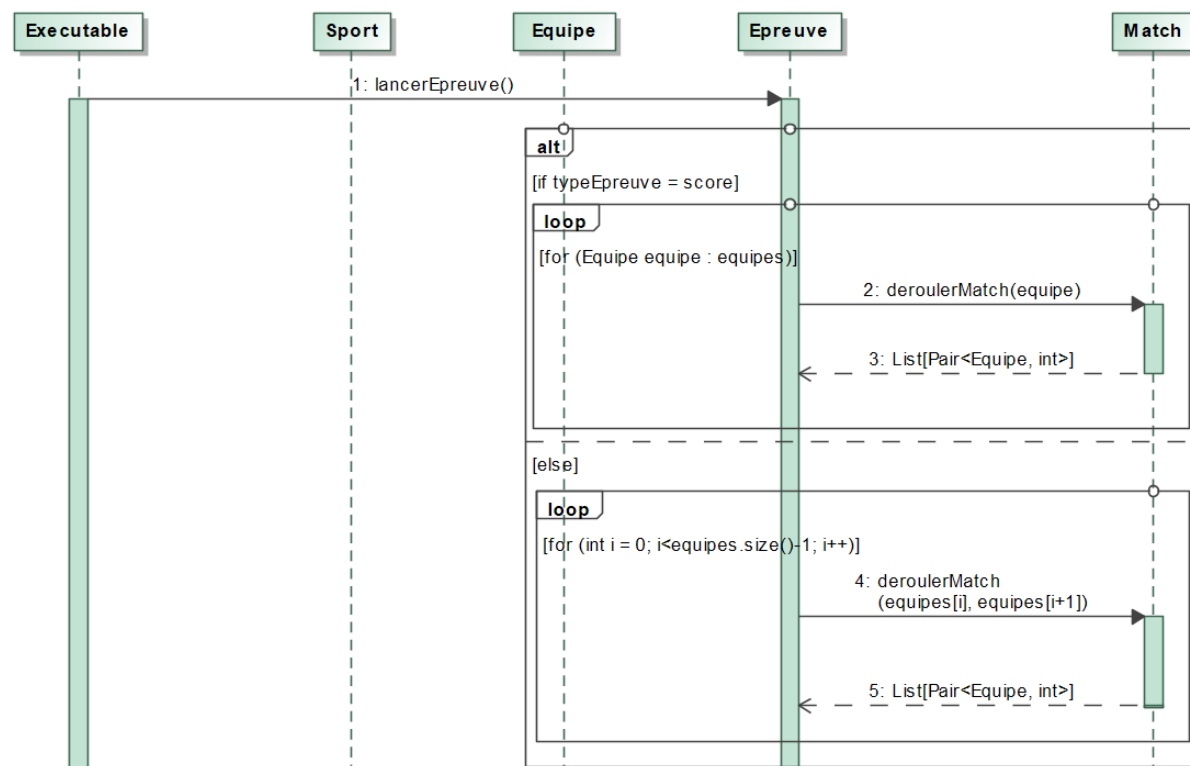


Figure 13 : Déroulement d'une épreuve sur le diagramme de séquence.

Ensuite « lancerEpreuve » va vérifier le type d'épreuve, son exécution créera des boucles qui lanceront « derouler_match » pour chaque équipe, il conservera le score de toutes les équipes participantes au match. Que ce soit un duel ou une performance à score.

« dérouler_match » durera jusqu'à ce que chaque équipe finisse de jouer pour que tous les scores soient établis.

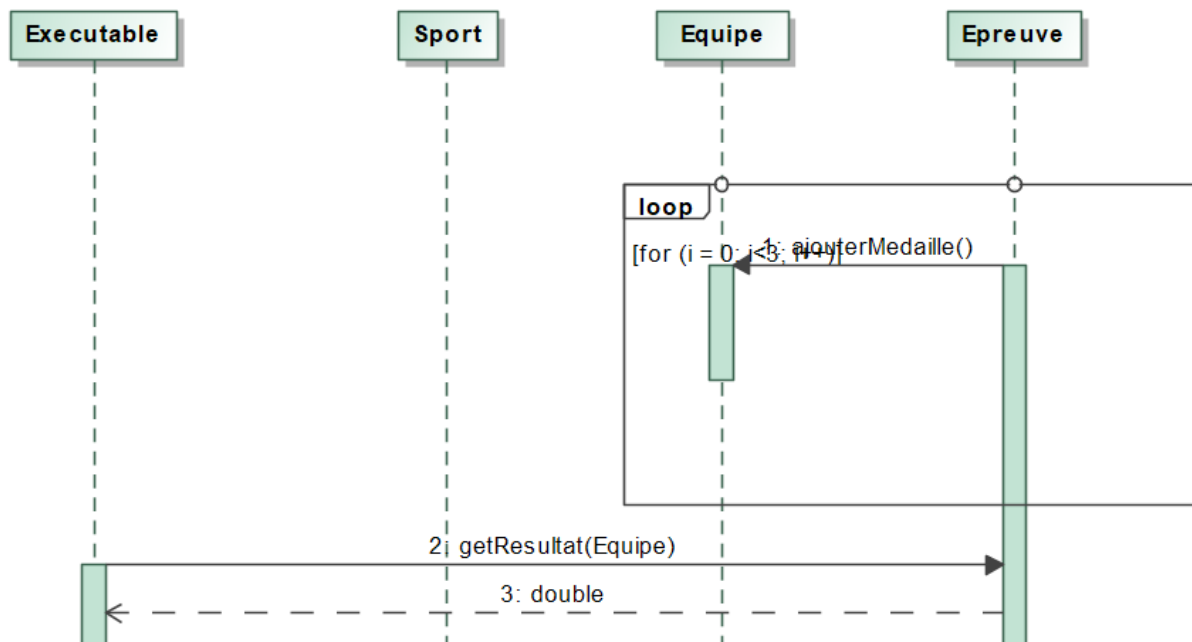


Figure 14 : Partie finale du diagramme de séquence.

Après ça, on attribue les médailles en fonction du classement grâce à « ajouterMedaille ». On pourra demander le score d'une équipe à une épreuve avec la méthode « get_resultat ».

2.4 Base de données

Cette section va représenter comment notre base de données est créée au début, notre MLD détaillé et la démonstration des dépendances relationnelles.

Notre base de données va récupérer les données des athlètes et les mettre dans un fichier CSV contenant 500 athlètes différents. Nous avons traité la partie java pour interagir avec. Quand nous aurons fini, nous stockerons ces données dans une classe annexe.

Voici le MLD qui contient les principales informations sur la représentation de la base. Certaines valeurs ont été rajoutées ou enlevées selon les besoins.

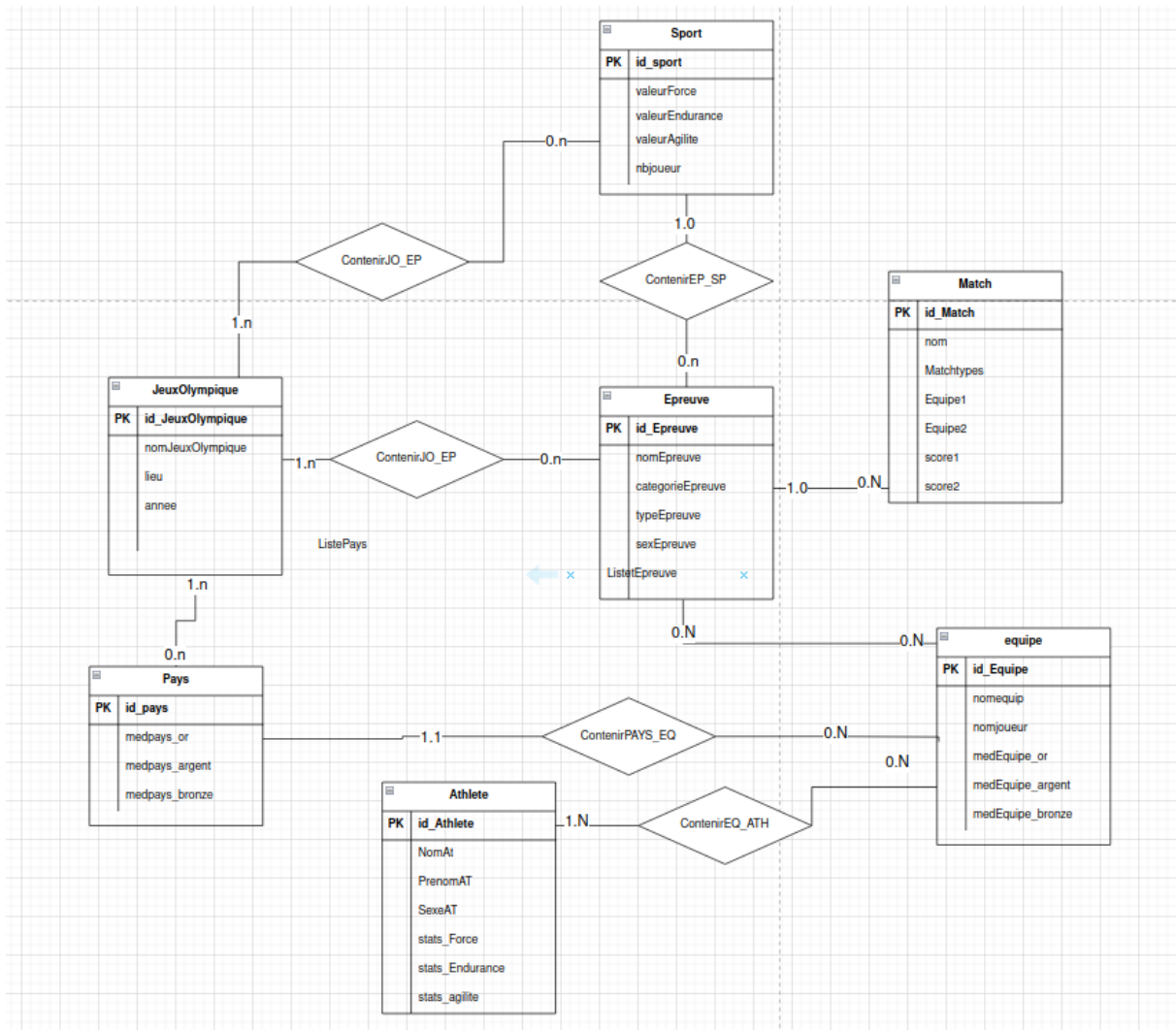


Figure 15 : Base de données.

Voici le modèle relationnel de notre base de données, on peut voir tous les éléments nécessaires à des Jeux Olympiques et quand ces derniers contiennent d'autres éléments nous avons une table contenir.

Il y a aussi qui n'est pas affiché sur ce diagramme, les tables utilisateur et activation afin de gérer la sécurité des droits et la gestion des utilisateurs.

3. Interface Homme Machine

Cette Partie sera consacrée à l'interface utilisateur de l'application, ainsi nous allons définir les emplacements des différents éléments visuels auquel l'utilisateur aura accès et à tous l'aspect ergonomique que l'application possède pour guider et s'adapter à différents utilisateurs.

3.1 Maquette

Ici contre, la maquette de l'interface accéder par les organisateurs

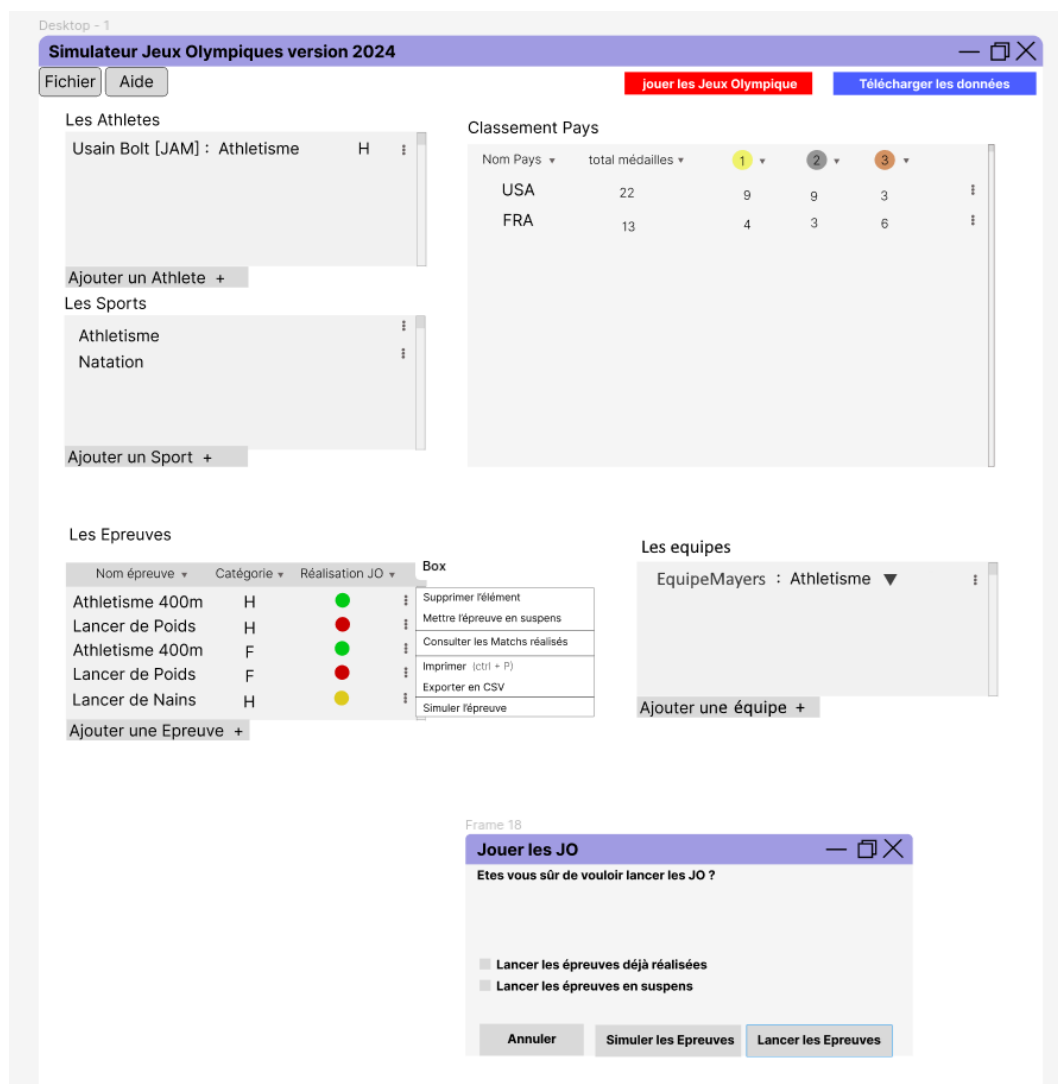


Figure 16 : Maquette de l'application

L'interface de l'application est assez simple afin d'être facilement utilisable par tout le monde, on y distingue 4 blocs principaux :

Le bloc Athlète : ce dernier permet d'afficher tous les athlètes participants à cette session des Jeux Olympique, en faisant un clic droit ou en appuyant sur les 3 points à droite sur un athlète nous pourrions obtenir des informations complémentaires le concernant.

Le bloc Sports : celui-ci permet d'afficher toutes les catégories de sport présent à cette session des Jeux Olympique, il est aussi possible d'obtenir plus d'informations comme pour les athlètes.

Le bloc Équipes : celui-ci affiche toutes les équipes d'athlètes pour les sports collectifs, en cliquant sur le triangle juxtaposé à la catégorie de l'équipe on peut dérouler cette dernière et afficher tous ces participants. Il est aussi

Le bloc épreuve : ici on peut traiter les épreuves, en ajouter, en retirer, les modifier. De plus, nous avons ajouté un aspect visuel avec des pastilles de couleur (voir figure 7 ref.1): cela définit si l'épreuve a déjà été lancée ou non étant donné qu'une option sera disponible pour réaliser une simulation d'une seule épreuve, cela n'exécutera qu'une épreuve et non les Jeux olympiques tout entier, il est possible de gérer l'épreuve selon les droits disponibles par la licence mais aussi de consulter les matchs réalisés par une simulation ou réellement (voir figure 7 ref.2)

Le bloc Classement pays: quant à lui ne sera utile qu'après la mise en route des JO via l'option rouge en haut à droite de l'écran "jouer les jeux olympiques", à partir de là, nous pourrions consulter le classement des pays lors de ces JO, trier les résultats par leur différentes médailles ou alors leur quantité totale de médailles.

Notre IHM contient aussi de petite fenêtre de confirmation pour les fonctions irréversibles comme "jouer les jeux olympiques" ou bien la suppression d'un élément

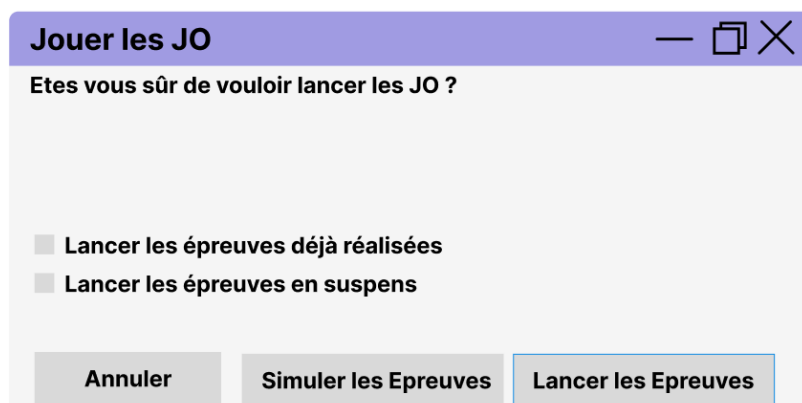


Figure 17 : Option de confirmation pour situation irréversible

Cette fenêtre s’affiche après avoir cliqué sur “Jouer les jeux olympiques”, (voir figure 7) On peut y voir de multiples éléments, premièrement sur la partie centrale de la fenêtre on retrouve des paramètres à cocher, sur cette maquette on dispose d’au moins 2 options prévues :

Lancer les épreuves réalisées, cette option permet de lancer les Jeux Olympiques en réalisant de nouveau les épreuves déjà effectuées (pastille verte, voir figure 7 ref.1)

Lancer les épreuves en suspens, cette option permet de lancer les Jeux Olympiques en réalisant les épreuves en suspens (pastille jaune, voir figure 7 ref.1), les épreuves en suspens sont des épreuves non-définitive qui ne sont pas enregistrées comme officiel, les consultant ne peuvent uniquement ajouter ce genre d’épreuve et cela leur donc permet de tester les résultats potentiels d’épreuves non-officielles

Ensuite nous voyons les 3 boutons du bas, annuler et les lancements d’épreuves :

Simuler les épreuves ne met pas à jour les pastilles de réalisation (voir figure 7 ref.1), mais met tout de même à disposition les données générées.

Lancer les épreuves n’est pas disponible pour les consultants mais permet de lancer réellement les épreuves, donc les pastilles de réalisation se mettront à jour (voir figure 7 ref.1).

Il est essentiel à rappeler qu’un bouton “aide” sera disponible, au cas où l'utilisateur ne s’y retrouve plus.

3.2 Interface graphique (IHM) réalisée

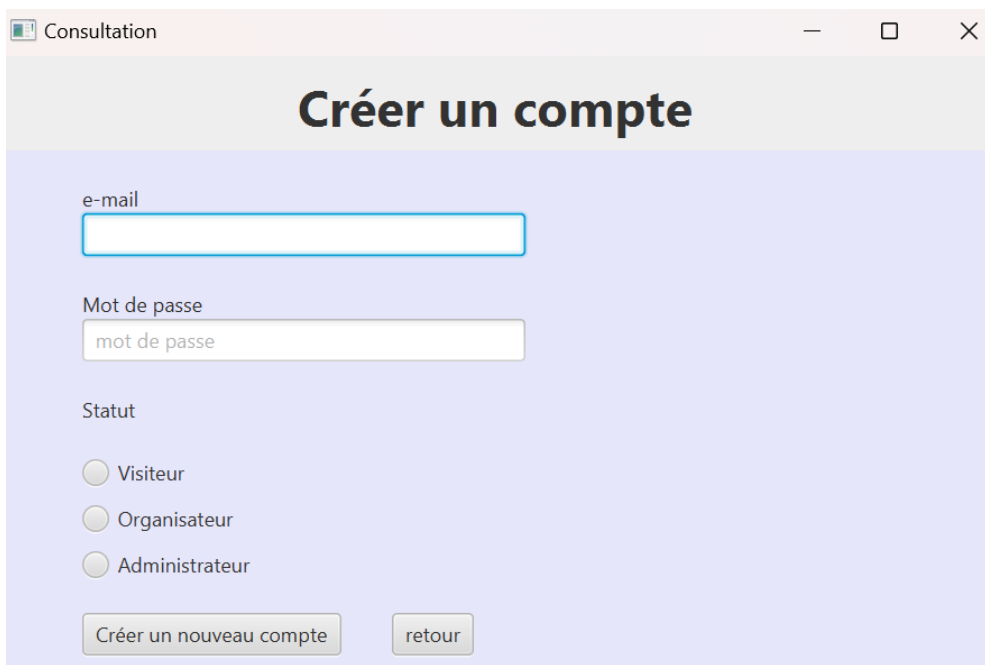
Durant cette dernière semaine nous avons dû réaliser l'interface graphique que nous avons planifié, globalement les maquettes ont été respectés, cependant le manque de temps à empêché de la finir correctement.



The screenshot shows a web application window titled "Consultation". The main heading is "Bienvenue sur Appli Gestion JO". Below it is a section titled "Connexion". There are two input fields: "Identifiant" and "Mot de passe". The "Mot de passe" field has a placeholder text "mot de passe". Below the input fields are two buttons: "Se connecter" and "Pas de compte?".

Figure 19 : Page de connexion

Cette fenêtre est la première vue que l'on voit quand on lance l'application, il suffit de rentrer ses identifiants pour se connecter, si l'on n'a pas de compte nous pouvons en créer un avec le bouton « Pas de compte » qui nous redirige.



The screenshot shows a web application window titled "Consultation". The main heading is "Créer un compte". There are two input fields: "e-mail" and "Mot de passe". The "Mot de passe" field has a placeholder text "mot de passe". Below the input fields is a section titled "Statut" with three radio button options: "Visiteur", "Organisateur", and "Administrateur". At the bottom are two buttons: "Créer un nouveau compte" and "retour".

Figure 20 : Page de création de compte

Si l'on ne dispose pas de compte il est possible d'en créer un, il suffit de rentrer des identifiants et si jamais on souhaite créer un compte Organisateur ou Administrateur il faudra rentrer une clé d'activation (non implémenté en front end).

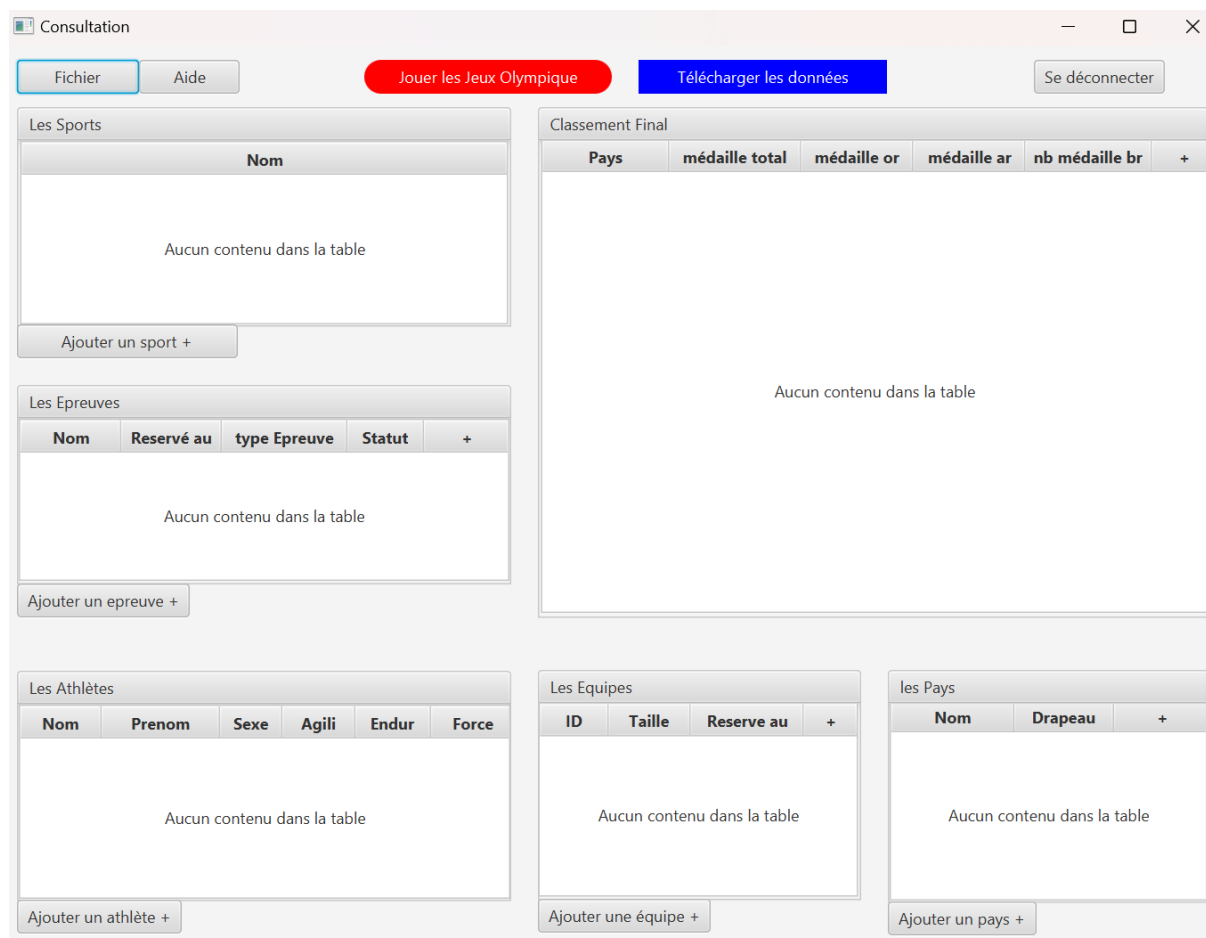


Figure 21 : Page de consultation (Accueil)

Lorsque l'on est connecté on arrive sur cette page, où l'on peut consulter les données des Jeux Olympiques, il est possible :

- de se déconnecter,
- d'ouvrir un fichier CSV
- télécharger les données
- consulter le manuel d'aide de l'application
- de jouer/simuler les jeux olympiques

Et évidemment pour chaque panel il est possible d'ajouter un élément, tel un athlète.

Consultation

Ajoute Athlète X Ajoute Sport Ajoute Equipe Ajoute Epreuve Ajoute Pays

Créer un Athlète

Compétence de l'athlète

Nom

Prenom

Sexe

Homme Femme

Agilité

Endurance

Force

Créer l'athlète retour

Figure 22 : Page d'ajout des éléments

Quand on clique sur un bouton pour ajouter un élément (voir fig. 21) on arrive sur cette page (dépendant du bouton auquel on a cliqué comme l'on peut le voir aux menus).

Sur cette vue on peut indiquer tous les attributs nécessaires à l'élément que l'on veut ajouter.

Pour des Pays, des Equipes ou encore des Epreuves il y a un menu d'ajout rapide pour les éléments qui sont inclus dedans.

4. Implémentation

Pour l'implémentation, nous avons suivi le modèle MVC permettant une séparation entre les éléments coder spécifiquement pour la vue, le modèle et les contrôleurs. Ensuite nous avons mis en place une démarche de connexion local vers nos ordinateurs personnels accédant directement à notre Mysql Mariadb.

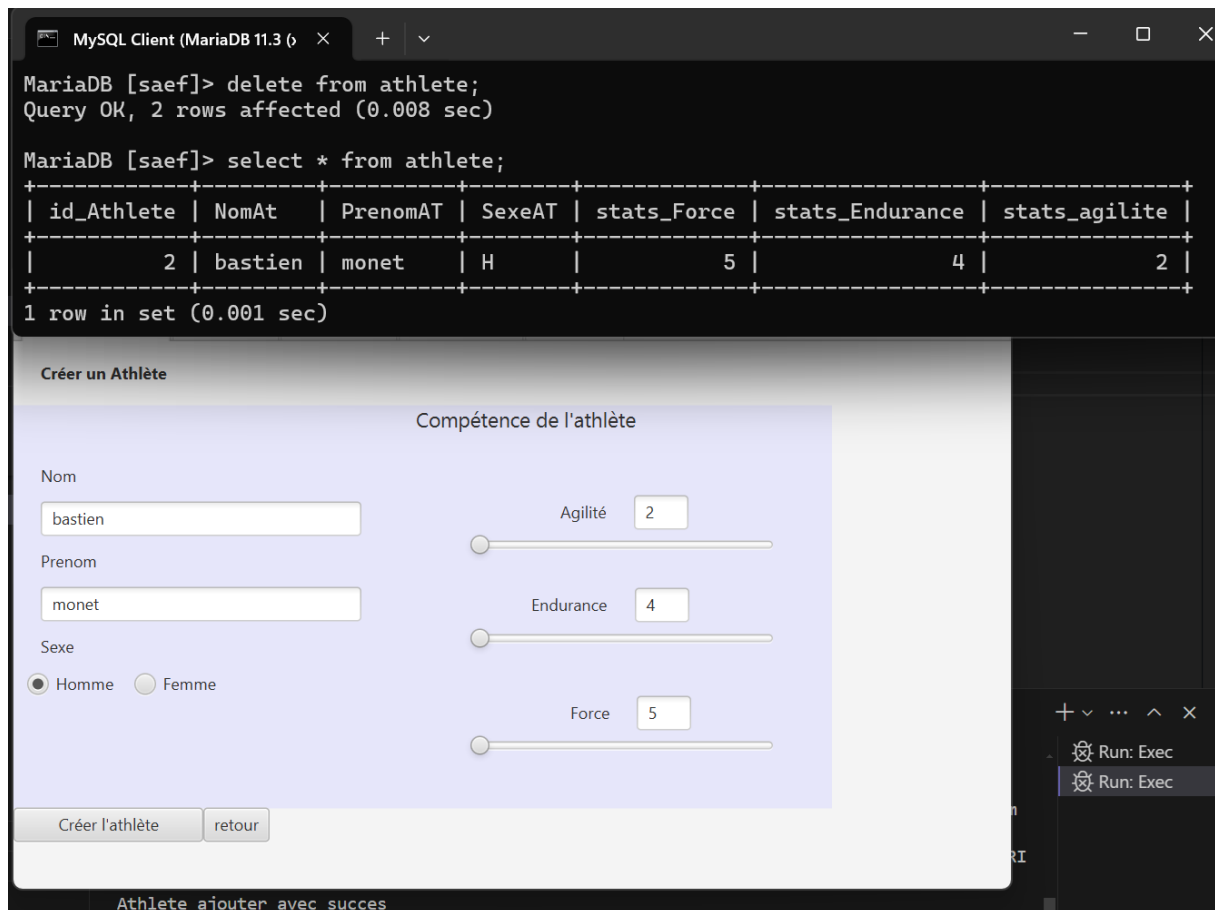


Figure 18 : Démonstration d'un Athlète placé dans la base de données

Voici une démonstration de la création d'un athlète dans notre vue « ajouter » (voir figure 18) , comme vous pouvez le constater, il suffit de rentrer manuellement les données et de sélectionner le bouton crée l'athlète pour directement l'ajouter à notre base de donnée (local)

Précédemment, nous avons implémenter notre base de données sur nos ordinateurs personnels, cela signifie que sans cela, celle-ci n'est pas fonctionnel l'application est exécutable depuis une distribution Windows, mais aucun test n'a été réaliser concernant la distribution linux

Lorsque n'importe élément est créé celui est stocké dans un cache en plus d'être placé dans la base de données si l'utilisateur à les bons droits.

Grâce à cet emplacement garder en mémoire, il nous est théoriquement possible d'exécuter les épreuves afin d'en avoir un classement final, pour cela on prendrait juste

toute les équipes des pays, que l'on affiche grâce à un fichier requetes JDBC mysql contenant nos requêtes permettant d'afficher les tableaux de donnée des divers instance créer.

5. Organisation du travail de groupe

Lors de cette SAE, dès la première semaine nous avons à réaliser 3 diagrammes UML, nous nous sommes réparti les tâches équitablement, malgré l'absence de Imad nous avons entamé les premiers diagrammes et établi un discord commun à la réalisation de ce projet, Shanka et Bastien s'occupait du diagramme de classe tandis que Clément établissait les cas d'utilisation.

La deuxième semaine nous avons commencé la planification des tâches (avec désormais la présence de Imad) à l'aide d'un Trello et avons aussi commencé la rédaction hebdomadaire du rapport à l'aide de Google docs, ces outils en ligne permettent une collaboration efficace entre tous les membres. Ensuite nous avons complété les deux diagrammes commencés précédemment, puis nous avons tous entamé la création du diagramme de séquence et du modèle relationnel.

La troisième semaine, durant la séance de PPP (sans Imad), nous avons avancé le rapport, en y ajoutant la partie Analyse UML, en ajoutant les diagrammes des semaines passées. Bastien s'est occupé de la retranscription textuelle des diagrammes puis à compléter la 3ème semaine de l'organisation du travail de groupe, pendant que shanka réalisait le design du rendu final, et enfin clément à améliorer le schéma expliquant la différence entre matchScore et matchDuel.

Puis en séance SAE, nous avons commencé la programmation de l'application, chacun de nous s'occupait d'une classe précédemment définie, l'objectif étant d'obtenir un code qui compile. Bastien c'est occuper la classe Epreuve, Shanka, celle de match, Clément a réalisé la classe Pays ainsi que l'interface comparator, et enfin Imad à réaliser les classes Sport et Athlete.

La quatrième semaine, pendant que Clément et Imad modifiait la rédaction du rapport, notamment en refaisant le diagramme de cas d'utilisation précédemment remis en question, Shanka et Bastien ont commencé à réaliser la maquette de la future IHM de l'application sur figma.

Ensuite lors de la SAE, Shanka et Bastien ont finalisé respectivement les classes matchs et épreuve, Clément quant à lui, c'est occupé de la classe équipe

La cinquième semaine, nous avons avancé le rapport, notamment en remettant à jour notre UML, c'était le travail de Shanka. Bastien a revu quelque partie inachevée du rapport et Clément et Imad se sont occuper du diaporama que nous utilisons pour présenter notre application lors de la soutenance

En SAE, la mission du jour a été de réaliser un exécutable dans le but de montrer à un professeur notre travail durant la SAE, c'est Bastien et Shanka qui s'en sont occupés, Clément quant à lui à essayer de rendre le code plus lisible en y ajoutant des docstrings.

La sixième semaine nous avons finalisé le rapport, Shanka à rajouter des indications sur la maquette de L'IHM en expliquant en détail les différentes fonctionnalités. Bastien quant à lui à achever la conclusion du rapport, Imad à choisi les ressources à utiliser dans la soutenance avec Clément qui à participer au design de la maquette. Clément a corrigé les cas d'utilisation.

Cette semaine finale nous avons travaillé d'arrache-pied pour essayer de finaliser ce projet (même si cependant nos efforts ont été vain), nous avons aussi eu l'ajout d'un nouveau membre à partir de Mercredi midi qui nous a facilité l'avancement de la base de donnée, Vendredi nous avons essayé de faire tous les documents.

6. Bilan du groupe et bilans individuels

6.1 Conclusion

Nous avons dans un premier temps réaliser différents diagrammes utiles au développement du code, le diagramme de classe, le diagramme de cas d'utilisation, le diagramme de séquence, ces diagrammes donnent différentes indications qui permettent à une personne lambda de comprendre notre projet.

Au final l'application contient 14 classes différentes en backend, cela nous permet de créer des jeux olympiques qui contiennent des sports et des Pays, de plus nous pouvons créer des équipes à partir d'athlète que nous pouvons également personnaliser selon leur capacité et leur pays. Ceci sera placés dans des épreuves et joueront des matchs jusqu'à obtenir un classement final.

Ces résultats donneront alors des médailles au gagnant, ce qui nous permettra après avoir effectué toutes les épreuves, d'obtenir le classement des médailles des pays.

Tout cela sera traitée dans une base de données afin de récupérer et traiter les différentes informations créer par les utilisateurs

Puis notre IHM permettra aux utilisateurs de réaliser ces différentes actions, y accéder par import de fichier et les télécharger. 3 rôles sont définis, ainsi le journaliste ne pourra que consulter les résultats et simuler les épreuves, l'administrateur pourra traiter la base de données en ajoutant ou retirant des éléments et seul l'organisateur pourra exécuter les jeux olympiques, télécharger les résultats, et les donner au journaliste.

Pour cette semaine finale, le manque de compétence, d'effectifs et de temps a fait qu'il nous a été impossible de tout finir correctement, ce qui fait que l'application n'est clairement pas fonctionnelle.

6.2 Bilans personnel

Clermont Shanka

Dans ce projet j'ai contribué à plusieurs choses, premièrement j'ai été un contributeur majeur sur l'architecture du programme en ayant énormément aidé sur la structure du diagramme de classes et ai mené sa re-réalisation (voir figure 2), de plus j'ai entièrement réalisé le diagramme de séquence de la gestion autour de la fonction "participer" qui permet de mieux comprendre l'aspect intérieur du logiciel ce qui a donc aidé le groupe (voir Figure 11).

Sur la réalisation du rapport j'ai créé le style et la forme du rapport et se basant sur des éléments officiels des Jeux Olympiques de Paris 2024, mais j'ai aussi rédigé une partie sur l'organisation du travail de groupe, rédigé la partie IHM/maquettage et conseillé mes camarades sur certains choix d'écriture.

La partie sur les maquettes a été réalisée en collaboration principalement avec Bastien Monet et j'ai principalement dirigé la charte graphique (voir Figure 14).

Si l'on regarde la réalisation du projet en termes de programmation, j'ai réalisé toute la partie concernant les matchs et à améliorer les méthodes produites sur les autres classes du logiciel, mais aussi participé grandement au questionnement sur l'algorithme de réalisation des épreuves.

Pour conclure, ma participation à ce projet a été importante et bien équilibrée sur tous les aspects du projet malgré quelques étapes qui m'ont ralenti sur les objectifs qui m'ont été fixés.

Cette dernière semaine j'ai réalisé et remodifié de multiples classes notamment sur la gestion des données, j'ai aussi ajouté des éléments sur la base de données.

Cependant, par manque de temps la quasi-totalité de mon travail n'a pas été implémenté avec le reste.

ASSOU Imad

Dans ce projet j'ai écrit plusieurs classes et ai contribué à l'écriture du rendu (explications des diagrammes)

Monet Bastien

Lors de cette SAE, je me suis proposé comme chef de projet. Ce rôle implique de devoir gérer l'équipe, orienter les différents membres du groupe sur leurs tâches respectives afin d'obtenir un travail structuré et cohérent.

En tant que chef de projet, j'ai veillé à ce que chaque membre de l'équipe comprenne bien ses tâches et ses responsabilités.

En plus de mes responsabilités de gestion, j'ai participé au développement du projet. J'ai réalisé la totalité de la classe Épreuve. J'ai également contribué au développement d'autres classes, comme Équipe et Pays, en apportant mon aide pour la conception et la programmation. Clément et Imad modifiaient la rédaction du rapport, notamment en refaisant le diagramme de cas d'utilisation

Concernant la rédaction du rapport, j'ai adapté et rédigé les paragraphes expliquant les différents diagrammes, le diagramme de classes et le diagramme de cas d'utilisation afin de faciliter le développement.

Mon seul regret est d'avoir mal utilisé GitHub au début du projet. En effet, une mauvaise gestion des branches et des commits a entraîné des pertes de temps et des difficultés dans le suivi des versions. Cependant, cette expérience m'a permis d'apprendre de mes erreurs et de m'améliorer dans l'utilisation des outils de gestion de version.

En conclusion, cette expérience en tant que chef de projet m'a permis de développer des compétences en gestion de projet, en communication et en développement logiciel. Malgré quelques difficultés.

Clément Renaudin

Dans ce projet, j'ai contribué à plusieurs choses que ce soit en programmation, l'analyse et la mise en compréhension du code, premièrement, je me suis attaqué à la partie analytique du projet avec les diagrammes de cas d'utilisation, j'ai aidé mes membres d'équipes avec des petites fonctions pour les aider.

J'ai principalement réalisé des modifications de certaines CLASS et de diagrammes où je me référais à mon groupe pour valider ou non ces modifications principalement avec Bastien Monet et Shanka

En plus de voir avec eux pour la réalisation de l'UML, je réalisais des documents qui me permettaient de donner une compréhension simple du projet avec par exemple la représentation de la base de données et de certains schémas le cas d'utilisation encore la réalisation de plusieurs prototypes de maquette permettant de comprendre comment fonctionne un match.

Bien sûr, j'ai programmé des fonctions diverses la plupart ont été modifier pour une meilleure optimisation de l'application, j'ai fait par exemple la classe Pays, la classe Équipe et quelques-uns des comparateurs en plus de rajouter des petites fonctions, j'ai ensuite essayé de réaliser deux class polymorphiques avec sport collectif et sport individuel.

On peut dire que j'ai principalement fait la partie documentation de la SAE avec la docstring des classes et la mise en place de la présentation pour la SAE et un support pour les autres codes.

Pour la partie du rapport, j'ai créé et récupéré les images importantes du projet, je faisais en sorte de faire une mise à jour régulier dû au changement fréquent de l'UML. J'ai complété la partie de cas d'utilisation et de la base de données

Pour conclure, ma participation lors de ce projet était régulier mais moins marquante dans l'ensemble cependant j'ai essayé au mieux de suivre le rythme et de me mettre à jour sur tous les aspects du projet pour aider mon groupe malgré quelques incompréhensions au début j'ai fait de mon mieux pour aider malgré mes difficultés en java pour ne pas pénaliser mon groupe.

Pour cette deuxième partie de la SAE, j'ai été un réel rôle dans l'équipe, j'ai créé les bases de la base de données et j'ai mis en place les rôles des utilisateurs. Puis quand Bastien a fini sa partie pour le modèle XML, j'ai récupéré les informations de ces nodes et je les ai instanciés, j'ai ensuite fait en sorte qu'on pouvait passer de scènes en scènes dans l'application en reprenant ce que j'avais appris avec le pendu ainsi la mise en place de mode pour gérer l'affichage des scènes avec la même fenêtre ce qui nous permet de changer les scènes et pages avec différents modules.

Mais avant la conversion de l'XML, j'ai réalisé un graphe scène pour avoir une idée claire de la scène que je voulais afficher, j'ai bien sûr utilisé scène Builder pour la réalisation de quelque scène et pour les alertes une modification de plusieurs types bouton pour pouvoir lancer les JO ou la simulation. Le XML a pris beaucoup de temps tout comme la mise en place des contrôleurs et des alertes, pour la réalisation de mes tâches, j'ai principalement travaillé avec Bastien lors de cette SAE pour la partie XML et correction de certaines erreurs.

Au cours de cette SAE les fondements de la base de données ont été changer, mais j'ai réussi à refaire le modèle relationnel de la base de données pour qu'on comprenne mieux la base. Pour finir, j'ai réalisé les diaporamas pour la présentation de même pour la réalisation du manuel d'utilisation...

Malo Dantec

Pour cette deuxième partie de la SAÉ, j'ai intégré ce groupe au milieu de la SAÉ en introduisant toute la première partie des requêtes jdbc qui consistait à faire des insertions et des suppressions dans toutes les tables. La modification est faite en parallèle si un élément (sa clé primaire) existe déjà et qu'on veut l'ajouter. J'ai également participé à la modification de la base de données pour bien faire le lien entre certaines tables (les clés étrangères) et en modifiant le contenu de certaines tables. J'ai fondé les bases de la connexion à mySQL reprise par mes collègues pour des améliorations.

7. Mentions légales :

1. Cette Situation d'Apprentissage et d'évaluation (SAÉ) ainsi que l'équipe du projet n'est en aucunement lié à la société PARIS 2024 COMITE D'ORGANISATION DES JEUX OLYMPIQUES ET PARALYMPIQUES, ce qui fait que toute production réalisée par les membres de cette équipe ne doit pas être utilisée comme élément officiel des Jeux Olympiques de PARIS 2024.
2. Ce rapport n'est en aucun cas un document officiel de la société PARIS 2024 COMITE D'ORGANISATION DES JEUX OLYMPIQUES ET PARALYMPIQUES, et par conséquent ne doit être interprété comme tel.
3. Ce document a été réalisé à des fins éducatives au sein de L'IUT d'Orléans, toute reproduction ou publication externe à l'université doit se faire sous l'autorisation de l'équipe du projet ou de l'équipe éducative.
4. Tous les noms liés aux Jeux Olympiques de PARIS 2024 appartiennent à la société PARIS 2024 COMITE D'ORGANISATION DES JEUX OLYMPIQUES ET PARALYMPIQUES et ne doivent donc pas être réutilisés à fins commerciales sans autorisation de cette société.

Cette Application a été réalisée avec le langage de programmation java



Rapport de l'application JAVA pour les Jeux Olympiques

