



Application JOly

Rapport SAE1256 DOO

Ben Ouirane Taycir
Bernard Adrian
R S Sukumar Chandra
Youssef Ralil

Groupe 2E



Table des matières

1.	Introduction	3
2.	Déroulement du projet	3
1.	Changements principaux par rapport au rendu UML	3
2.	Répartition du travail.....	4
3.	Modèle MVC	4
4.	Implémentation des Fonctionnalités	4
	Exécution	5
3.	Modification UML / IHM	9
4.	Conclusion	9

1. Introduction

Ce rapport porte sur la réalisation de notre application JOly, dédiée à la gestion des équipes, des sessions et des épreuves des Jeux Olympiques. L'objectif de ce projet est de fournir une solution complète et intuitive pour répondre aux besoins des organisateurs, des athlètes et des spectateurs.

Notre démarche a été structurée autour du modèle MVC (Modèle-Vue-Contrôleur), garantissant une séparation claire des responsabilités entre la gestion des données, l'interface utilisateur et la logique de contrôle. Cette approche permet de créer une application modulaire, maintenable et évolutive.

L'application JOly intègre principalement **les fonctionnalités** essentielles suivantes :

- Gestion des équipes : création, ajout, suppression d'équipes.
- Gestion des sessions et des épreuves.
- Affichage du planning des épreuves à venir.

En nous appuyant sur les exigences spécifiques du projet et en respectant les principes de la programmation orientée objet, nous avons développé une interface conviviale et intuitive, facilitant la gestion des Jeux Olympiques. Ce rapport détaille les différentes étapes de la réalisation de ce projet, les choix technologiques et les résultats obtenus, offrant une vision complète de notre démarche et des solutions mises en œuvre pour atteindre nos objectifs.

2. Déroulement du projet

1. Changements principaux par rapport au rendu UML

Avant de commencer les autres étapes de la mise en œuvre du modèle MVC, nous avons révisé ensemble notre diagramme UML initial. Cette révision collective a permis de détecter les ajustements nécessaires pour mieux aligner le modèle avec les exigences du projet.

Les changements ont été identifiés pour simplifier l'architecture et améliorer la modularité de l'application. **Par exemple**, nous avons décidé d'intégrer directement certaines méthodes de gestion des équipes dans les classes de modèle pour éviter la complexité excessive d'une classe de gestion intermédiaire.

2. Répartition du travail

Après avoir révisé l'UML et apporté les modifications nécessaires, nous avons réparti le travail de manière équilibrée entre les membres de l'équipe. Chaque membre a pris en charge des parties spécifiques de l'application, assurant une répartition claire des responsabilités :

- Adrian : Ajout des procédures événementielles d'équipe/athlète, IHM de sessions/planning. (+Rapport)
- Chandra : Ajout des procédures événementielles de session/plannings, IHM de menu et retouche IHM d'équipe. (+Rapport)
- Taycir : Ajout des procédures événementielles du menu, IHM d'équipe et retouche IHM de menu. (+Rapport)
- Ralil : Mise en forme du rapport.

3. Modèle MVC

Nous avons structuré notre application selon le modèle MVC (Modèle-Vue-Contrôleur). Cette architecture a permis une séparation claire des responsabilités entre la gestion des données, la logique métier et l'interface utilisateur.

- ❖ **Modèle (Model)** : Les classes de modèle ont été conçues pour gérer les données et la logique métier. Par exemple, la classe Equipe gère les informations sur les équipes et leurs membres.
- ❖ **Vue (View)** : Les classes de vue ont été développées pour créer l'interface graphique. Par exemple, les classes PanelEquipe, ButtonEquipe, et PanelEquipe gèrent l'affichage des données et les interactions utilisateur.
- ❖ **Contrôleur (Controller)** : Les contrôleurs, comme MenuButtonListener, gèrent les actions des utilisateurs et mettent à jour les modèles et les vues en conséquence.

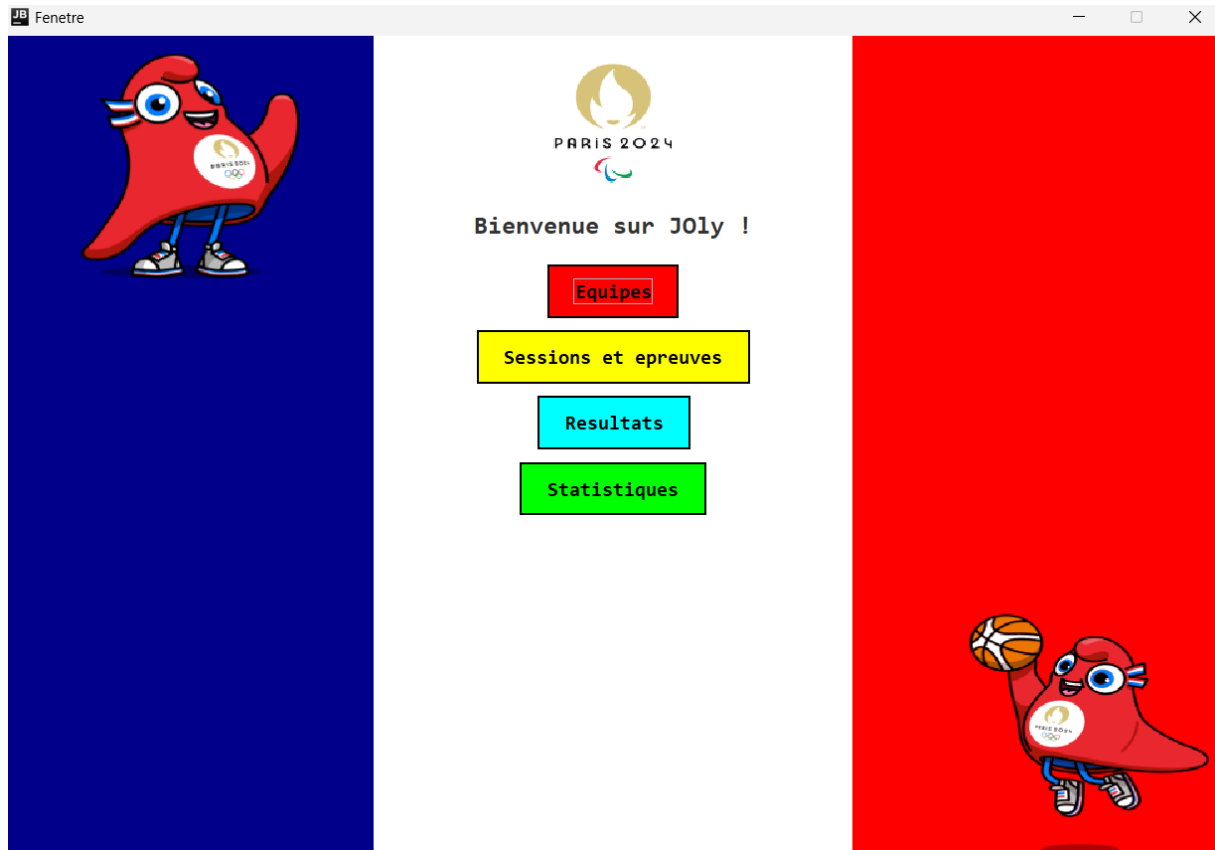
Afin d'avoir un meilleur regroupement des fichiers, le MVC a été découpé avec des sous paquetages comme equipe, menu, et session.

4. Implémentation des Fonctionnalités

- **Gestion des Équipes** : Création, ajout et suppression des équipes. Ces fonctionnalités permettent aux utilisateurs de gérer les équipes participant aux épreuves.
- **Gestion des Sessions et des Épreuves** : Planification et gestion des sessions et des épreuves. Cela inclut la création de nouvelles sessions et l'affichage des informations sur les épreuves. Par manque de temps les sessions ont été simplifier et ne tiennent pas compte des épreuves.
- **Affichage du Planning des Épreuves** : Nous avons utilisé JTable pour afficher le planning des épreuves dans une interface tabulaire, permettant aux utilisateurs de visualiser facilement les sessions à venir.


3. Exécution

Voici quelques captures d'écran de l'exécution de l'exécution :



Fenetre

Equipes






Nos Equipes

Nom de l'equipe:

Nom du pays:


Ajoutez une Equipe !

Equipe1	Pays1		
---------	-------	-----------------------------------------------------------------------------------	-----------------------------------------------------------------------------------



Fenetre

Athletes



Nos Athletes

Nom de l'athlete:

Prenom de l'athlete:


Age de l'athlete:


Email de l'athlete:

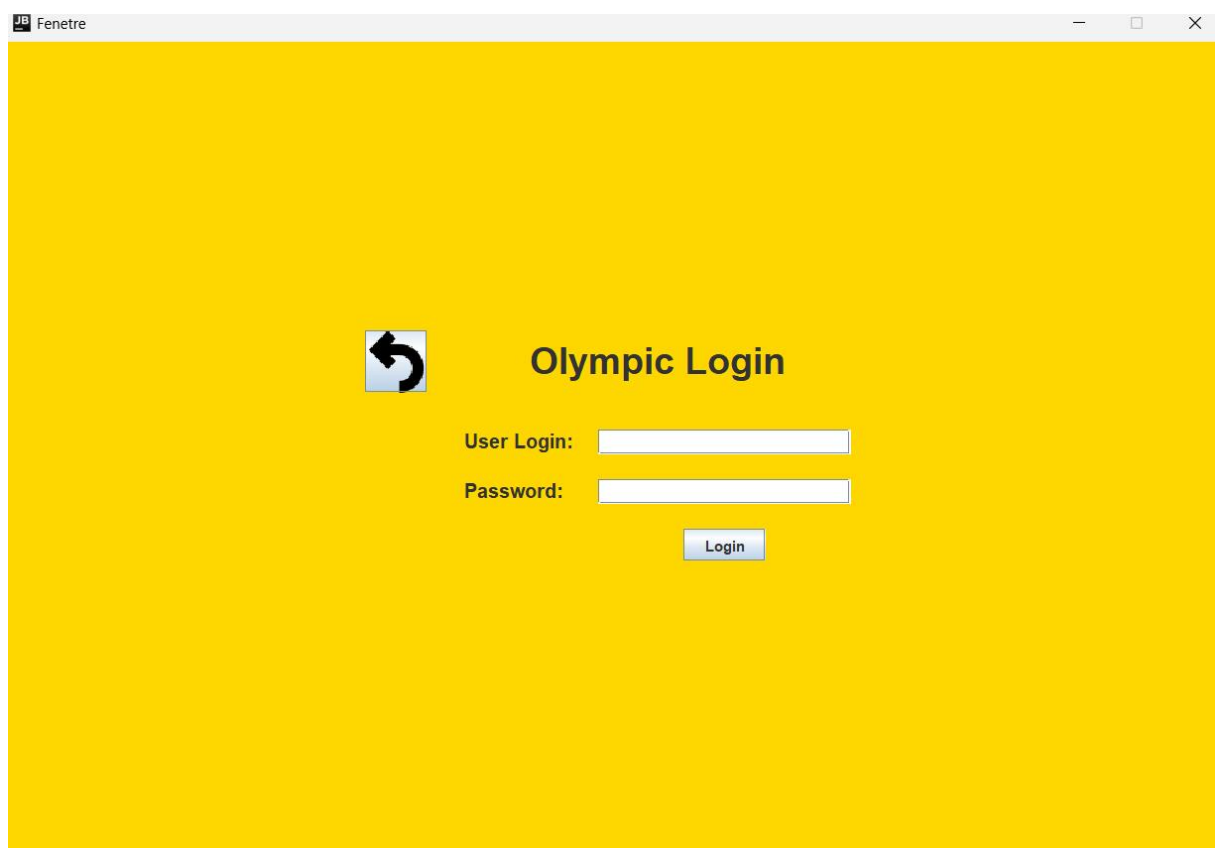
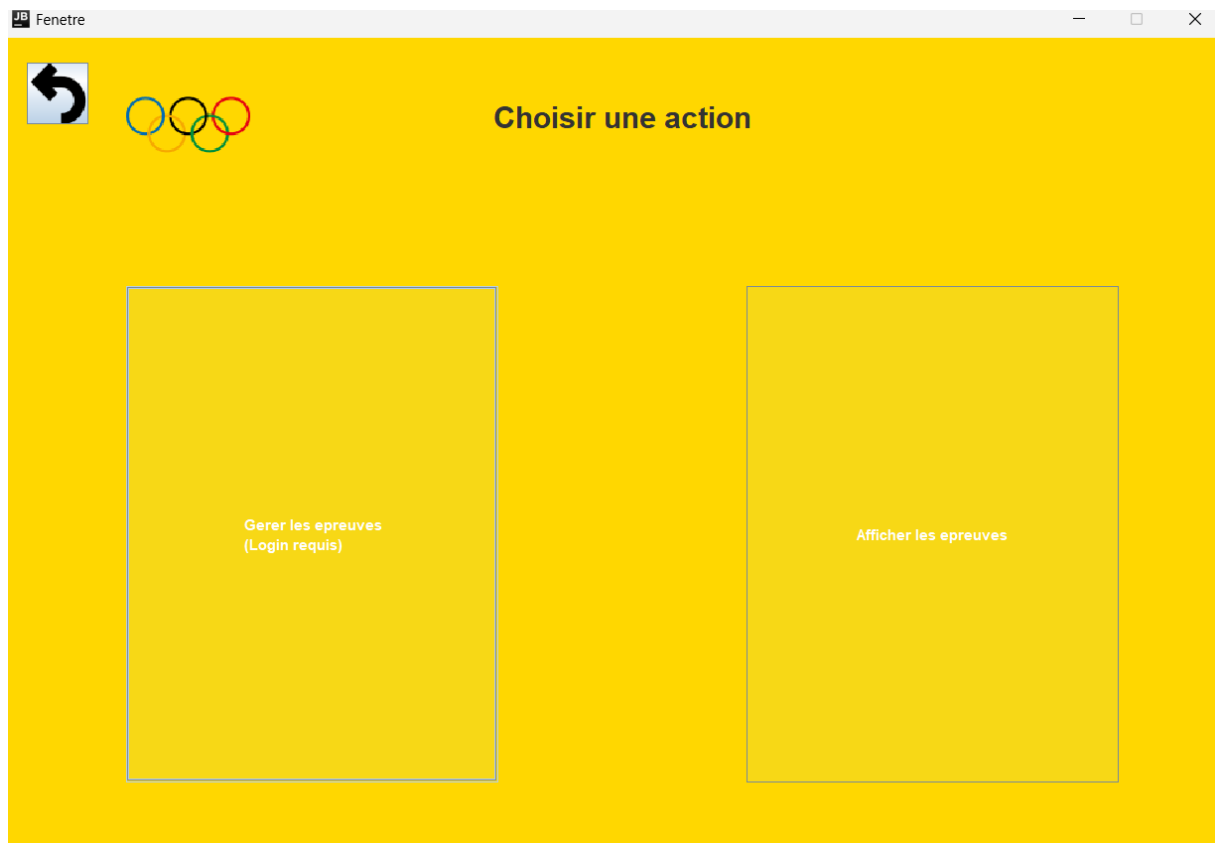
Poids athlete (kg):

Taille athlete (cm):

Ajoutez un Athlete !


Nom : a	Prenom : a	Age : 1
Mail : a@gmail.com	Taille (cm) : 182	
Poids (kg) : 75		





Fenetre

Gerer les Sessions



Sport:

Date de debut:


Duree (HH:MM):

Description:

Ajouter Session

Fenetre

Planning des sessions



Epreuve	Debut	Duree	Type
Sport1	08/06/2024	01:30	Demi-finale

Chaque équipe peut être modifiée séparément, en modifiant une équipe vous accéderez à la page EquipeAthlete qui permet de rajouter ou supprimer des athlètes.

4.Modification UML / IHM

Afin de respecter les contraintes de temps et de simplifier le programme de manière général, des modifications ont été effectuées sur l'idée initiale de l'interface graphique et des classes du projet.

Pour l'interface graphique on retrouve quelque chose de plus simple tout en gardant les couleurs du rendu IHM.

Pour les classes les on retrouve des classes allégées, notamment pour athlète ou équipe et des moins de vérification qu'initialement prévu (vérif doublon, athlète existant etc...).

Ces changements ont été nécessaires afin de rendre une application fonctionnelle dans les délais demandés.

4.Conclusion

Le projet JOly, structuré selon les principes du Développement Orienté Objet (DOO) et le modèle MVC, a abouti à une application robuste, modulaire et intuitive pour la gestion des équipes, des sessions et des épreuves des Jeux Olympiques. Grâce à une méthodologie agile et une collaboration efficace, nous avons développé une interface conviviale répondant aux besoins des utilisateurs. La prochaine étape inclue des tests utilisateurs pour identifier les améliorations nécessaires, garantissant ainsi une expérience optimale et renforçant le succès de l'application.