

- Rapport de projet de synthèse -
Réalisation d'une application
« Tous sports individuels »

Université Cergy Pontoise

Auteurs : Ralaimanantsoa Hery Mickael
Phetramphand Antoine

Table des matières

| | |
|--|----|
| I. Présentation du projet | 4 |
| 1. Contexte | 4 |
| 2. Objet | 4 |
| 3. Organisation | 4 |
| a) analyse..... | 4 |
| b) spécification | 4 |
| c) conception | 4 |
| d) programmation | 5 |
| e) tests | 5 |
| f) validation..... | 5 |
| g) maintenance..... | 5 |
| 4. Environnement | 5 |
| 5. Planification | 6 |
| II. Objectif | 6 |
| 1. Description brève du logiciel | 6 |
| 2. Le but de cette application..... | 6 |
| 3. Liste des fonctionnalités..... | 6 |
| 4. Contraintes | 6 |
| III. Livraisons attendues..... | 7 |
| IV. Conception du projet | 7 |
| 1. Diagramme UML..... | 7 |
| 2. Schéma MCD | 8 |
| 2. Schéma MLD..... | 8 |
| V. Description du schéma conceptuel..... | 9 |
| 1. Description des classes Java | 9 |
| a) classe UTILISATEUR | 9 |
| b) classe JdbcUtilisateurAccess | 9 |
| c) classe SPORT | 9 |
| d) classe POIDS | 9 |
| e) classe TEMPS | 9 |
| f) classe VITESSE | 10 |

| | |
|--|----|
| g) classe Exercice | 10 |
| h) classe IHM | 10 |
| i) classe MENU | 10 |
| j) classe UtilisateurTableModel | 10 |
| k) classe UtilisateurDialog | 10 |
| l) classe JDBCconnect | 11 |
| 2. Description des tables (Entités)..... | 11 |
| a) table UTILISATEUR..... | 11 |
| b) table SPORT | 11 |
| c) table POIDS..... | 11 |
| d) table TEMPS | 11 |
| e) table EXERCICE | 11 |
| f) table VITESSE..... | 12 |
| 3. Description des relations entre les tables..... | 12 |
| VI. Manuel d'utilisation | 13 |
| VII. Remarques sur le déroulement du projet | 18 |
| 1. Méthode de travail au sein du groupe..... | 18 |
| 2. Difficultés rencontrées | 18 |
| VIII. Conclusion | 19 |
| IX. Remerciements | 20 |
| X. Glossaire | 21 |

I. Présentation du projet

1. Contexte

Dans le cadre de notre formation à l'Université de Cergy Pontoise, les étudiants en Licence *Informatique* troisième année réalisent des projets personnels afin de valider leur année scolaire. Ce type de réalisation exige donc un certain temps, un investissement personnel et l'usage de toutes les ressources disponibles. Ici, nous allons réaliser une application intitulant « ***Tous sports individuels*** ».

2. Objet

Le but est la réalisation d'une application qui consiste à gérer les données personnelles de l'utilisateur concernant les sports qu'il exerce.

3. Organisation

Notre équipe est composée de deux membres à la charge du projet : *Ralaimanantsoa Hery Mickael* et *Phetramphand Antoine*.

Pour mieux appréhender le projet, nous allons entamer des recherches et approfondir les non acquis et mettre le problème en œuvre techniquement en utilisant le processus «*modèle en cascade*». Il s'agit du modèle le plus simple des processus logiciels en termes de complexité et de facilité de mise en œuvre et repose sur les étapes suivantes :

a) analyse

On essaye de comprendre le problème, puis d'analyser les besoins fonctionnels (contrainte technique).

b) spécification

On décrit les besoins du client, puis on traduit les besoins en fonctionnalités.

c) conception

On transforme le problème en solution.

d) programmation

On passe du résultat de la conception à un ensemble de programmes traduit en un langage de programmation (Écriture des textes des programmes).

e) tests

On recherche des erreurs dans une spécification ou programme.

f) validation

Le système répond aux exigences du client et s'assure de l'adéquation des résultats de l'analyse et de la spécification.

g) maintenance

On veille au bon fonctionnement des programmes du projet. Les programmes doivent être robustes.

4. Environnement

Les ressources que nous disposons pour le développement de notre projet sont :

- **PhpMyAdmin** : logiciel permettant de programmer en langage **SQL**.
- **Eclipse** : logiciel permettant de développer nos programmes en langage **Java**
- **Dia** : logiciel permettant de réaliser notre schéma **MCD**.
- **Wampserver** : environnement comprenant deux serveurs (**Apache** et **MySQL**), un interpréteur de script (**PHP**), ainsi que **phpMyAdmin** pour l'administration Web des bases **MySQL**.
- **JDBC (Java Data Base Connectivity)** : bibliothèque ou interface **Java** d'accès aux bases de données **SQL**.
- **JFreechart** : bibliothèque open source qui permet d'afficher des données statistiques sous la forme de graphiques.

Les langages que nous utiliserons :

- Java
- SGBD utilisé : **MySQL**

5. Planification

Délai : le projet doit se finir avant le 10/03/2015.

II. Objectif

1. Description brève du logiciel

L'objectif du projet consiste à créer une application qui permet à l'utilisateur de gérer ses données personnelles concernant les sports qu'il pratique. L'utilisateur a la possibilité d'ajouter, modifier et/ou supprimer des données. S'il le souhaite, il peut consulter ses statistiques sous forme de représentations graphiques.

2. Le but de cette application

Cela consiste à l'utilisateur à charger ses données une fois qu'il soit connecté à la base de données ou à une partie de l'application que nous avons développé ; puis à sauvegarder en saisissant ses informations qui seront stockées dans la base de données. Enfin, l'utilisateur pourra lire ses données sous forme de graphiques (courbes, camemberts et histogrammes).

3. Liste des fonctionnalités

Cette application permettra de réaliser les fonctionnalités suivantes :

- Gestion des données personnelles de l'utilisateur : ajout, suppression et/ou mise à jour de différents types de données.
- Génération des graphiques : l'utilisateur a la possibilité de lire ses données personnelles sous forme de graphiques (courbes, camemberts et histogrammes).
- Gestion multiutilisateurs : les utilisateurs peuvent partager et/ou visualiser des graphiques sur le même sport entre eux s'ils le souhaitent (y donner les droits).
- Offrir une simplicité d'utilisation : le jeu devra être facile à utiliser et à manipuler, c'est-à-dire moins de touches du clavier à utiliser.

4. Contraintes

- La base de données doit être conforme aux normes **SQL**.

- Le code d'interrogation de la base de données doit respecter les règles définies dans les normes **SQL**.

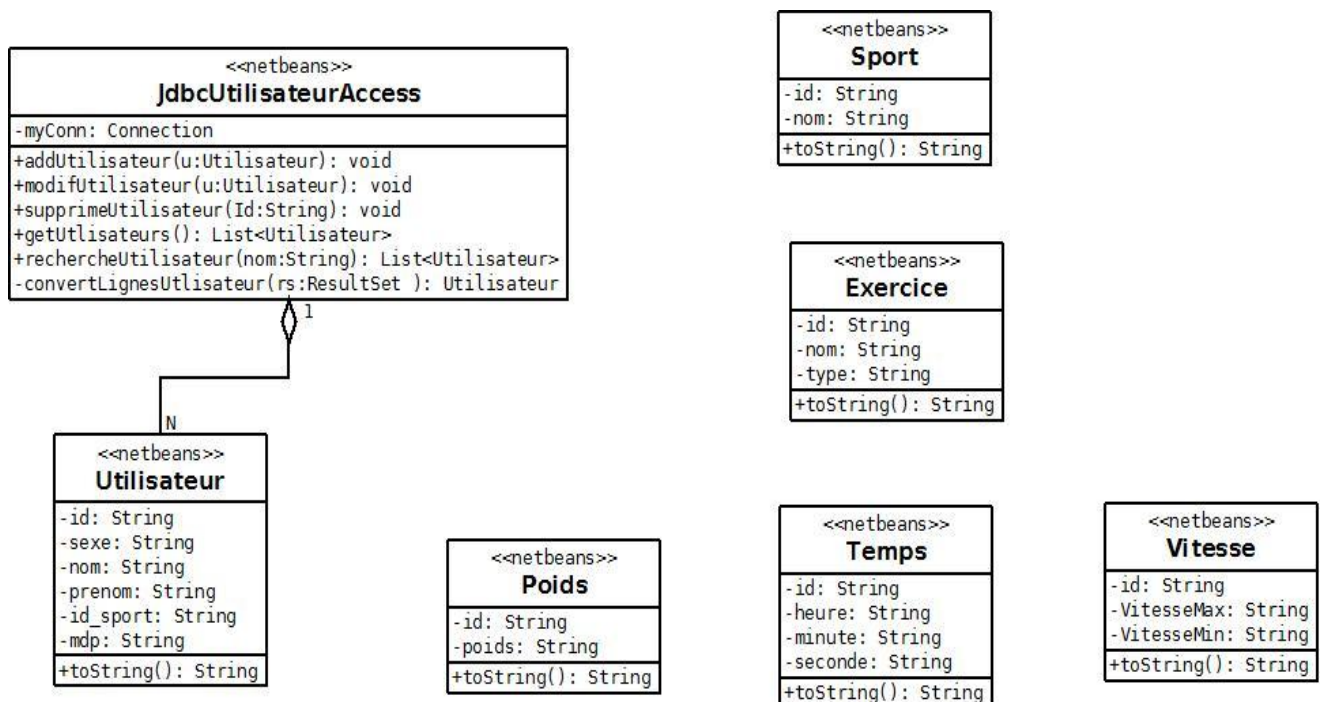
III. Livraisons attendues

À la fin de notre projet, nous pensons rendre les documents suivants :

- code source des programmes en langage Java
- un rapport rédigé en pdf
- manuel d'utilisation du logiciel
- un fichier sql

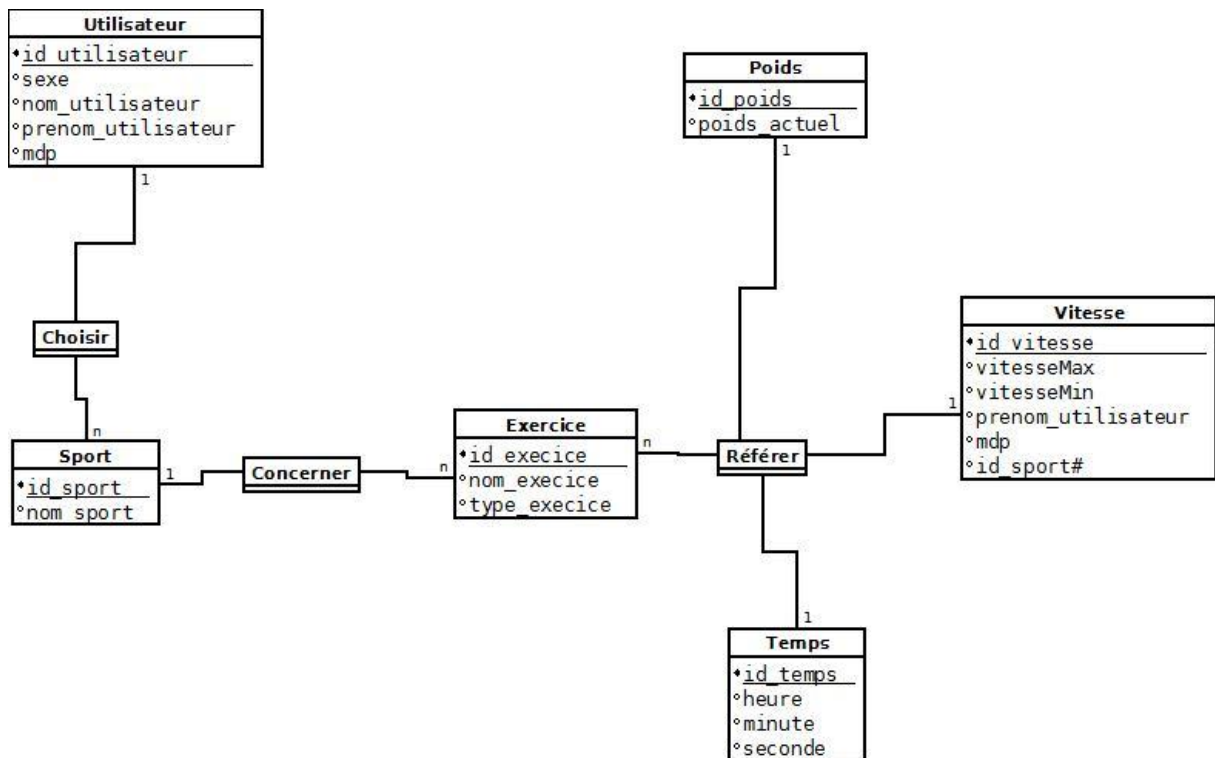
IV. Conception du projet

1. Diagramme UML



2. Schéma MCD

Voici le schéma MCD :



2. Schéma MLD

Voici le schéma MLD :

UTILISATEUR (id_utilisateur, sexe, nom_utilisateur, prenom_utilisateur, mdp, #id_sport)

SPORT (id_sport, nom_sport, #id_execice)

POIDS (id_poids, poids_actuel, #id_execice)

TEMPS (id_temps, heure, minute, seconde, #id_execice)

EXERCICE (id_execice, nom_execice, type_execice)

VITESSE (id_vitesse, vitesseMax, vitesseMin, #id_execice)

V. Description du schéma conceptuel

1. Description des classes Java

a) classe UTILISATEUR

C'est la classe représentant un utilisateur.

Elle a pour attributs les informations d'un utilisateur (son numéro id, son sexe (M pour masculin et Mme pour Madame), son prénom, son mot de passe).

Elle dispose d'une méthode toString() pour permettre l'affichage des informations d'un utilisateur.

b) classe JdbcUtilisateurAccess

C'est la classe qui contient l'ensemble des utilisateurs trouvés (stockés dans une *ArrayList*).

Elle dispose des méthodes d'ajout, de mise à jour et de suppression d'un utilisateur.

c) classe SPORT

C'est la classe représentant un sport.

Elle a pour attributs les informations du sport (son numéro id, le nom du sport).

Elle dispose d'une méthode toString() pour permettre l'affichage des informations d'un sport.

d) classe POIDS

C'est la classe représentant un poids donné.

Elle a pour attributs les informations d'un poids exercé par l'utilisateur (son numéro id, son poids).

Elle dispose d'une méthode toString() pour permettre l'affichage des informations d'un poids.

e) classe TEMPS

C'est la classe représentant un temps réalisé par l'utilisateur.

Elle a pour attributs les informations d'un utilisateur (son numéro id, les heures, les minutes et les secondes passées).

Elle dispose d'une méthode toString() pour permettre l'affichage des informations d'un temps.

f) classe VITESSE

C'est la classe représentant une vitesse réalisée par l'utilisateur.

Elle a pour attributs les informations d'une vitesse (son numéro id, sa vitesse maximale et sa vitesse minimale).

Elle dispose d'une méthode toString() pour permettre l'affichage des informations d'une vitesse.

g) classe Exercice

C'est la classe représentant un exercice.

Elle a pour attributs les informations d'un utilisateur (son numéro id, son nom, son type).

Elle dispose d'une méthode toString() pour permettre l'affichage des informations d'une vitesse.

h) classe IHM

C'est la classe qui représente l'interface graphique principale.

i) classe MENU

C'est la classe qui représente une partie de l'application, c'est-à-dire, elle représente une interface graphique permettant à l'utilisateur de se connecter en saisissant son nom et son mot de passe.

j) classe UtilisateurTableModel

C'est la classe qui gère le tableau (JTable) qui affiche les informations sur les utilisateurs.

k) classe UtilisateurDialog

C'est la classe qui ouvre une fenêtre de type JDialog qui demande de saisir les informations sur les utilisateurs soit pour ajouter, modifier ou supprimer les données.

La méthode setData() permet de rafraîchir le tableau en utilisant la nouvelle l'ArrayList fournie.

I) classe JDBCconnect

C'est la classe qui servira une partie de l'IHM pour se connecter à la base de données.

2. Description des tables (Entités)

a) table UTILISATEUR

- id utilisateur : numéro id de l'utilisateur
- sexe : le sexe de l'utilisateur
- nom utilisateur : le nom de l'utilisateur
- prenom utilisateur : le prénom de l'utilisateur
- mdp : mot de passe
- id sport : numéro id

b) table SPORT

- id sport : numéro id du sport
- nom sport : nom du sport

c) table POIDS

- id poids : numéro id du poids
- poids actuel : le poids actuel

d) table TEMPS

- id temps : numéro id
- heure : les heures passées en exercice
- miniute : les minutes passées en exercice
- seconde : les secondes passées en exercice

e) table EXERCICE

- id execice : numéro id de l'exercice
- nom execice : le nom de l'exercice

- type_execice : le type d'exercice

f) table VITESSE

- id_vitesse : numéro id de la vitesse
- vitesseMax : la vitesse maximale passée en exercice
- vitesseMin : la vitesse minimale passée en exercice

3. Description des relations entre les tables

Choisir

Un utilisateur peut choisir un ou plusieurs sports

Concerner

Chaque sport concerne un ou plusieurs exercices

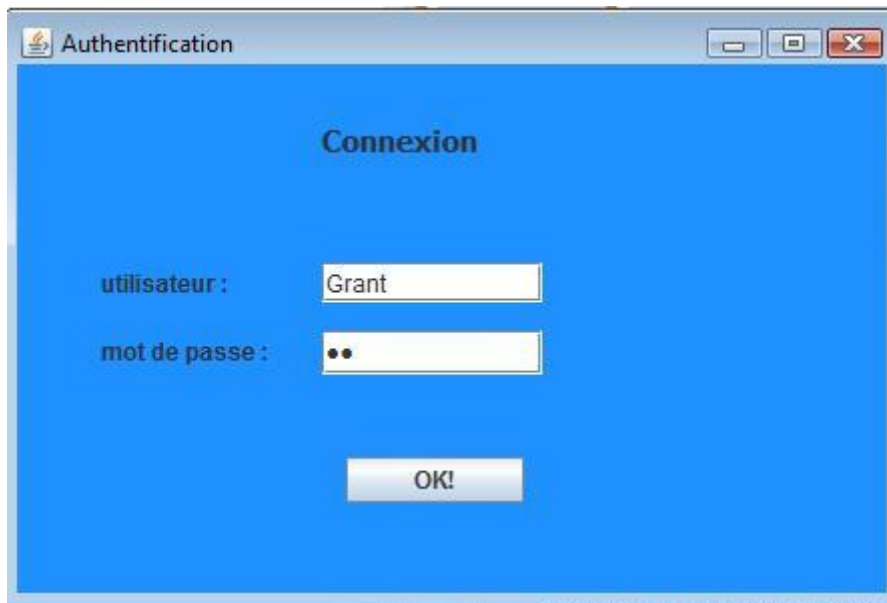
Référent

Chaque exercice réfère au moins à une information sur la vitesse, le poids et le temps.

VI. Manuel d'utilisation

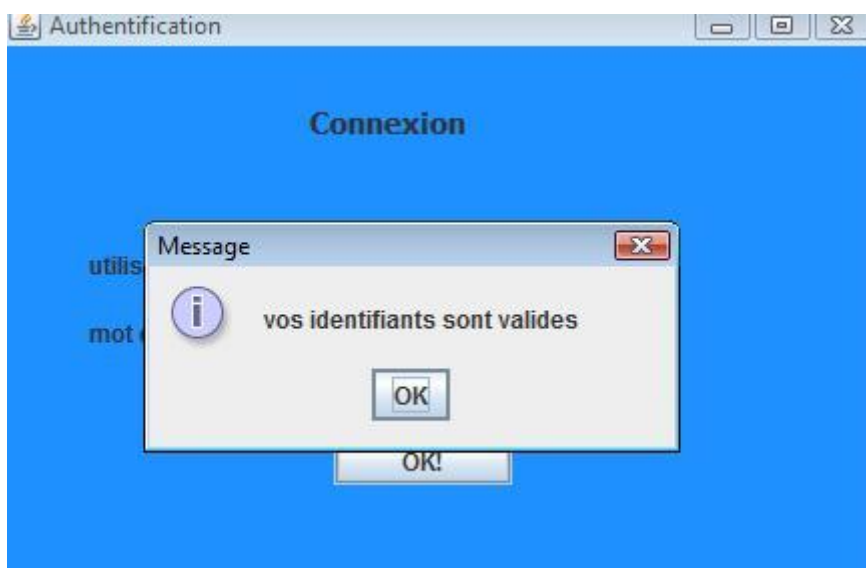
Voici un petit manuel d'utilisation qui vous aidera à utiliser ce logiciel

Lors du lancement du logiciel, l'interface suivant apparaît :



1) On vous demandera de saisir les champs ci-dessus pour vous connecter, ici donc le nom de l'utilisateur et le mot de passe.

NB : les mots de passe pour chaque utilisateur est le même. Donc c'est « **bd** » le mot de passe à saisir si vous voulez tester le programme.



2) Si les champs saisis sont bien correctement valides, alors votre connexion est bien établi sinon on vous redemandera de saisir à nouveau les identifiants.

| Sexe | Nom | Prenom |
|------|------------|-----------|
| Mme | Chanteloup | Sandrine |
| M | Dos Santos | Alexandre |
| Mme | Dumas | Catherine |
| M | Grant | Alain |
| M | Miller | Richard |

| id vitesse | 01 | 02 | 03 | 04 |
|-------------|------|------|------|------|
| Vitesse max | 54.8 | 99.1 | 76.6 | 23.8 |
| Vitesse min | 34.1 | 55.2 | 22.5 | 7.5 |

3) L'image ci-dessus représente la partie principale de l'application

En tapant sur « **Recherche** » en laissant le champ du texte vide, cela va afficher par défaut toutes les données concernant les utilisateurs (*son sexe, son nom et son prénom*). En revanche, si on cherche une donnée précise comme son nom (ou une partie de son nom si elle existe), le tableau affichera les informations de cette personne, et non les autres utilisateurs.

Exemple : en tapant « **Chanteloup** » ou « **Chant** », cela affichera en résultat « **Mme Chanteloup Sandrine** ».

Pour quitter l'application, appuyez sur « **Quitter** ».

The screenshot shows a web application interface with a blue header. A modal dialog titled 'Ajout utilisateur' is open, allowing the addition of a new user. The dialog contains four input fields: 'ID' with the value '26', 'Sexe' with 'M', 'Nom' with 'toto', and 'Prenom' with 'titi'. At the bottom of the dialog are two buttons: 'Enregistré' and 'Annulez'. In the background, a table is visible with the following data:

| Sexe | Nom | Prenom |
|------|------------|-----------|
| Mme | Chanteloup | Sandrine |
| M | Dos Santos | Alexandre |
| Mme | Dumas | Catherine |
| M | Grant | Alain |
| M | Miller | Richard |
| M | toto | titi |

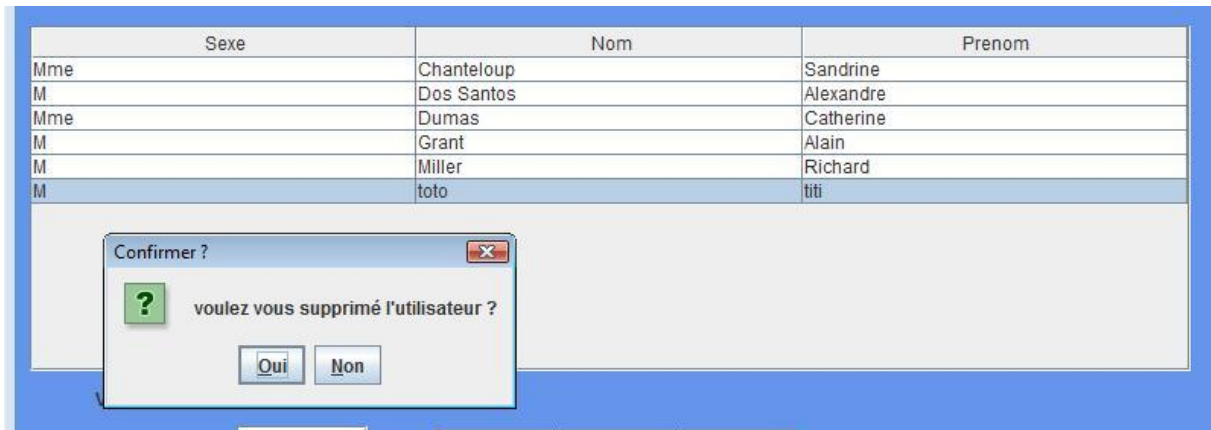
4) Si on clique sur « **Ajouter** », l'application ouvrira une fenêtre qui invite à l'utilisateur d'ajouter un autre utilisateur dans le tableau. Dans notre exemple, on a choisi de saisir « **26** » son ID, « **M** » son sexe, « **toto** », « **titi** » son prénom.

NB : Si on clique sur « **Modifier** » au lieu d'appuyer sur « **Ajouter** », l'application ouvrira ce même type fenêtre pour entrer des données. Cependant, cette fonction a pour but de modifier ces données, non de les ajouter.

The screenshot shows the same web application after a successful registration. A modal dialog titled 'Utilisateur enregistré' is open, displaying an information icon and the message 'Sauvegarde réussie.' with an 'OK' button. The background table now includes the newly added user:

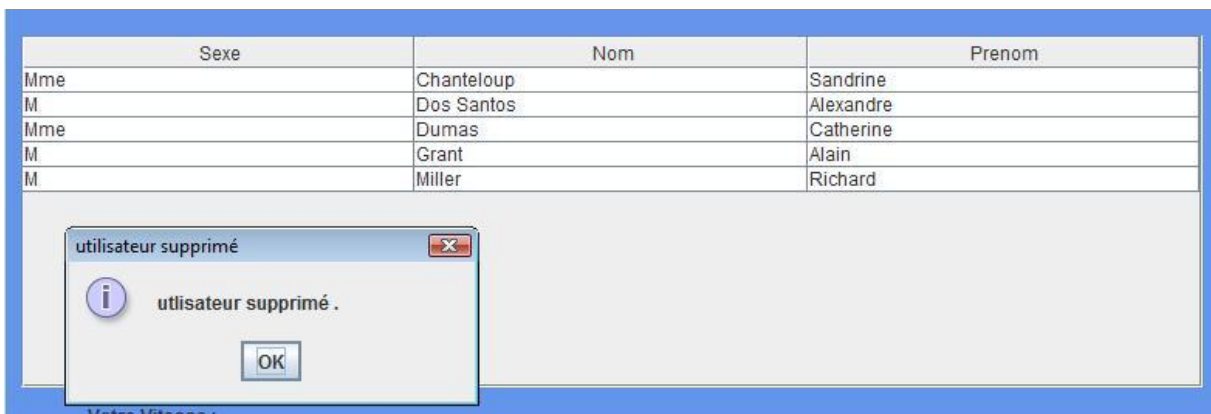
| Sexe | Nom | Prenom |
|------|------------|-----------|
| Mme | Chanteloup | Sandrine |
| M | Dos Santos | Alexandre |
| Mme | Dumas | Catherine |
| M | Grant | Alain |
| M | Miller | Richard |
| M | toto | titi |

5) Si ces champs sont bien saisis et validés, alors ces données sont bien sauvegardées et ajoutées dans le tableau.



6) Si on clique sur « **Supprimer** », l'application demandera à l'utilisateur s'il souhaite de supprimer ou non ses données sélectionnées.

NB : avant de cliquer sur « **Supprimer** » pour pouvoir supprimer, veuillez sélectionner une ligne du tableau sinon cela affichera une erreur.



7) Si oui, ces données seront totalement supprimées (*son ID, son sexe, son nom, ainsi que son prénom*) du tableau.

The screenshot shows a web application interface with two dropdown menus. The first menu is labeled 'sport :' and has 'Aikido' selected. The second menu is labeled 'exercice :' and has 'aérobie' selected.

8) Cette partie du produit permet de choisir son sport et le type d'exercice.

Votre Vitesse :

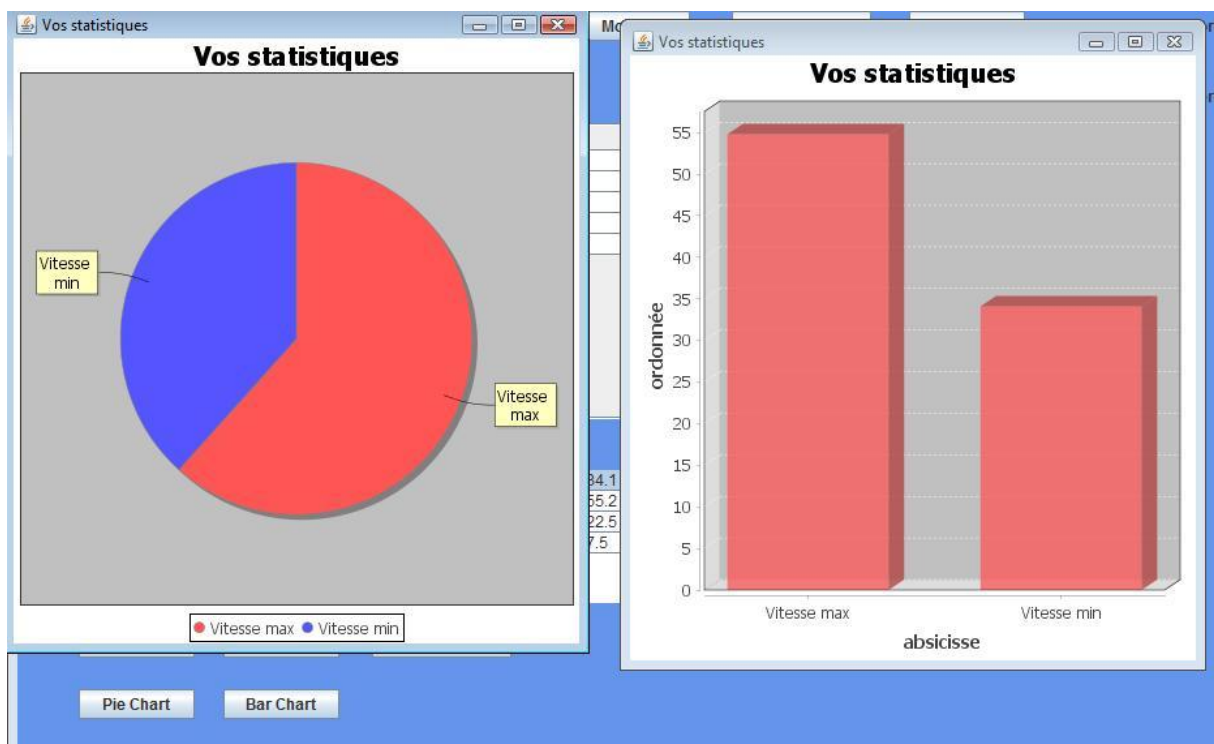
id vitesse

Vitesse max

Vitesse min :

| | | |
|----|------|------|
| 01 | 54.8 | 34.1 |
| 02 | 99.1 | 55.2 |
| 03 | 76.6 | 22.5 |
| 04 | 23.8 | 7.5 |

9) Cette partie de l'application concerne les données (son ID, sa vitesse maximale et la vitesse minimale) de vitesse saisies par l'utilisateur.
Elle comprend plusieurs fonctions : ajouter, modifier et supprimer les données.
Ces fonctions fonctionnent de la même manière que pour les données des utilisateurs



10) Lorsqu'on clique sur « **Pie Chart** » et « **Bar Chart** », cela affichera les statistiques en **camembert** et en **histogramme**.

VII. Remarques sur le déroulement du projet

1. Méthode de travail au sein du groupe

Après avoir choisi et analysé le sujet, nous nous réunissions pour mettre en commun nos idées, puis nous nous répartissons les tâches pour effectuer chacun une partie du travail. Puis, nous nous réunissions afin d'effectuer des tests de notre projet et de corriger les erreurs.

Nos travaux étaient enregistrés sur nos ordinateurs personnels pour éviter toute perte de données.

2. Difficultés rencontrées

- l'exécution des requêtes.
- la connexion de la base de données associée à nos classes Java
- l'authentification des utilisateurs
-

VIII. Conclusion

Par conséquent, nous avons donc créé notre programme construit avec des fonctionnalités grâce au logiciel *PhpMyAdmin* et *Eclipse*.

En réalisant ce projet, nous nous sommes rendus compte de la difficulté du métier d'informaticien.

Il ne suffit pas seulement de savoir créer des algorithmes mais il faut également parvenir à se renseigner lorsqu'on ne maîtrise pas un sujet.

De plus, au niveau informatique nous avons découvert et maîtrisé de nouvelles techniques de programmation (*JDBC*) et perfectionné notre programmation *SQL*.

Ceci nous aura été bénéfique pour des projets futurs que ce soit dans les études ou dans la vie professionnelle.

Si nous devons refaire ce projet, nous déciderons d'optimiser avec plus de données, soit plus de classes Java et l'authentification de l'utilisateur pour améliorer les parties manquantes du projet.

Pour finir, bien que nous n'ayons pas réussi à achever ce projet, nous sommes assez contents du résultat obtenu.

IX. Remerciements

Nous tenons particulièrement à remercier notre professeur qui nous a encadrés durant tout ce semestre.

En effet, LIU Tianxiao, enseignant pédagogique et très compétent, nous a aidés à mener à bien notre projet de parcours.

X. Glossaire

Eclipse : un environnement de développement (IDE) Open Source principalement destiné au langage JAVA et fonctionnant à base d'extensions (plugins).

Java : langage de programmation orienté objet.

Jdbc (Java Data Base Connectivity) : bibliothèque ou interface *Java* d'accès aux bases de données **SQL**.

JFreechart : bibliothèque open source qui permet d'afficher des données statistiques sous la forme de graphiques.

MySQL : système de gestion de bases de données relationnelles

SGBD (système de gestion de base de données) : logiciel système destiné à stocker et à partager des informations dans une base de données, en garantissant la qualité, la pérennité et la confidentialité des informations, tout en cachant la complexité des opérations. Il permet d'inscrire, de retrouver, de modifier, de trier, de transformer ou d'imprimer les informations de la base de données

Wampserver : plateforme de développement *Web* de type **WAMP**, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts *PHP*. *WampServer* n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (**Apache** et **MySQL**), un interpréteur de script (*PHP*), ainsi que **phpMyAdmin** pour l'administration *Web* des bases **MySQL**.