

Cahier des charges techniques

Sommaire

1. Contexte du projet

1.1. Présentation du projet

1.2. Date de rendu du projet

2. Besoins fonctionnels

3. Ressources nécessaires à la réalisation du projet

3.1. Ressources matérielles

3.2. Ressources logicielles

4. Gestion du projet

5. Conception du projet

5.1. Le front-end

5.1.1. Wireframes

5.1.2. Maquettes

5.1.3. Arborescences

5.2. Le back-end

5.2.1. Diagramme de cas d'utilisation

5.2.2. Diagramme d'activités

5.2.3. Modèles Conceptuel de Données (MCD)

5.2.4. Modèle Logique de Données (MLD)

5.2.5. Modèle Physique de Données (MPD)

6. Technologies utilisées

6.1. Langages de développement Web

6.2. Base de données

7. Sécurité

7.1. Login et protection des pages administrateurs

7.2. Cryptage des mots de passe avec Bcrypt

7.3. Protection contre les attaques XSS (Cross-Site Scripting)

7.4. Protection contre les injections SQL

1. Contexte du projet

1.1. Présentation du projet

Votre agence web a été sélectionnée par le comité d'organisation des jeux olympiques de Paris 2024 pour développer une application web permettant aux organisateurs, aux médias et aux spectateurs de consulter des informations sur les sports, les calendriers des épreuves et les résultats des JO 2024.

Votre équipe et vous-même avez pour mission de proposer une solution qui répondra à la demande du client.

1.2. Date de rendu du projet

Le projet doit être rendu au plus tard le 7 novembre 2024.

2. Besoins fonctionnels

Le site web devra avoir une partie accessible au public et une partie privée permettant de gérer les données.

Les données seront stockées dans une base de données relationnelle pour faciliter la gestion et la mise à jour des informations. Ces données peuvent être gérées directement via le site web à travers un espace administrateur.

3. Ressources nécessaires à la réalisation du projet

REPRENDRE RÉPONSES MISSION 1

3.1. Ressources matérielles

- PC Portable
- PC fixe
- Clavier
- Sourie
- Ecran
- Unité centrale

3.2. Ressources logicielles

- IDE = Environnement de développement (VS CODE)
- GITHUB = Plateforme de développement collaborative
- APACHE = Serveur web contenue dans MAMP
- SGBDR Relationel = MYSQL contenue dans MAMP
- TRELLO = Outil de gestions de projet
- UML et arboressence = Visuale paradingm online
- Figma = Maquette (4)
- Mokodo = Conception BDD

3.3. Langages utilisé

- HTML 5
- CSS 3
- Javascript
- Back end : PHP 8
- Langage interrogation BDD : SQL
- Stack overflow : PHP

4. Gestion du projet

Pour réaliser le projet, nous utiliserons la méthode Agile Kanban. Nous utiliserons également l'outil de gestion de projet en ligne Trello.

INSÉREZ UNE CAPTURE D'ÉCRAN DE VOTRE TRELLO

Nous travaillons également sur GitHub, plateforme de développement collaboratif.

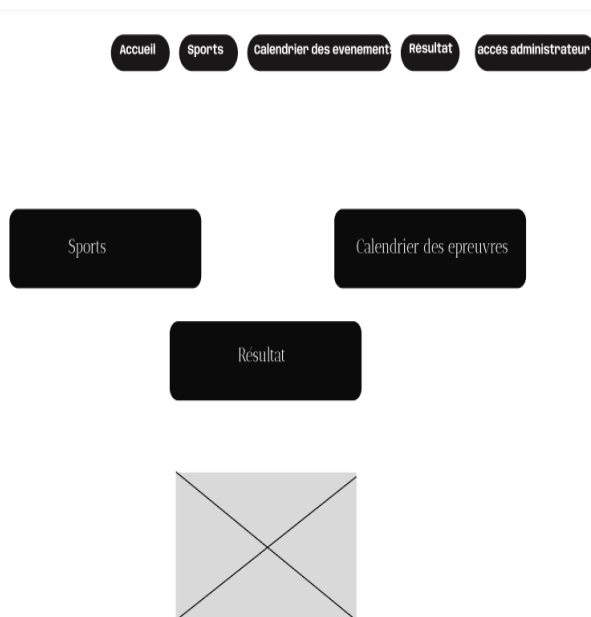
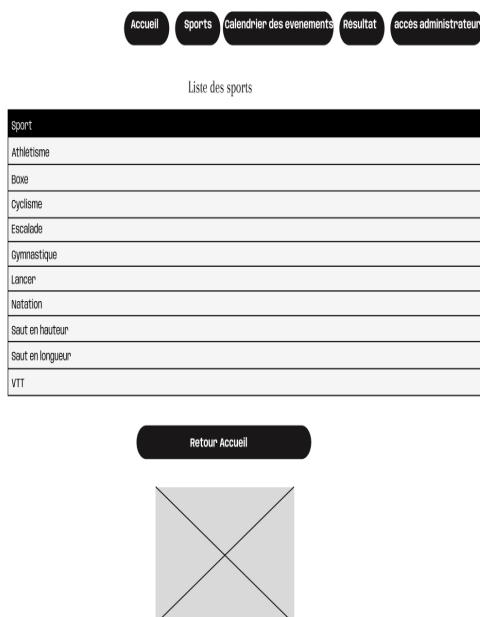
5. Conception du projet

5.1. Le front-end

REPRENDRE RÉPONSES MISSION 4

5.1.1. Wireframes

Les wireframes façons pc :

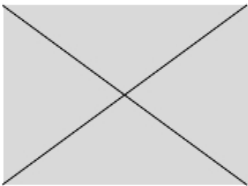


Connexion

Login :

Mot De Passe :

Se connecter



Versions tablettes :

Accueil

Sports

Calendrier des evenements

Résultat

accès administrateur

Liste des Sports

Sport

Athlétisme

Badm

Cyclisme

Escalade

Gymnastique

Lancer

Natation

Shot en l'air

Shot en l'air junior

VTT

Retour Accueil

Accueil

Sports

Calendrier des evenements

Résultat

accès administrateur

Calendrier des épreuves

Sports

Résultat

Accueil

Sports

Calendrier des evenements

Résultat

accès administrateur

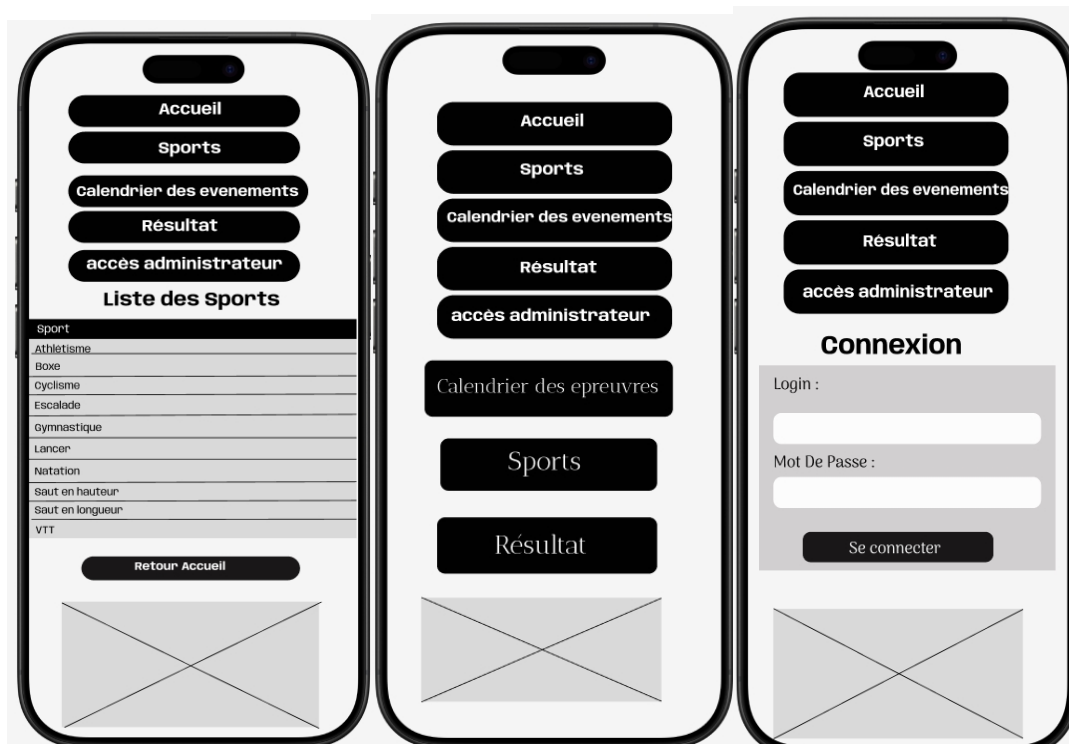
Connexion

Login :

Mot De Passe :

Se connecter

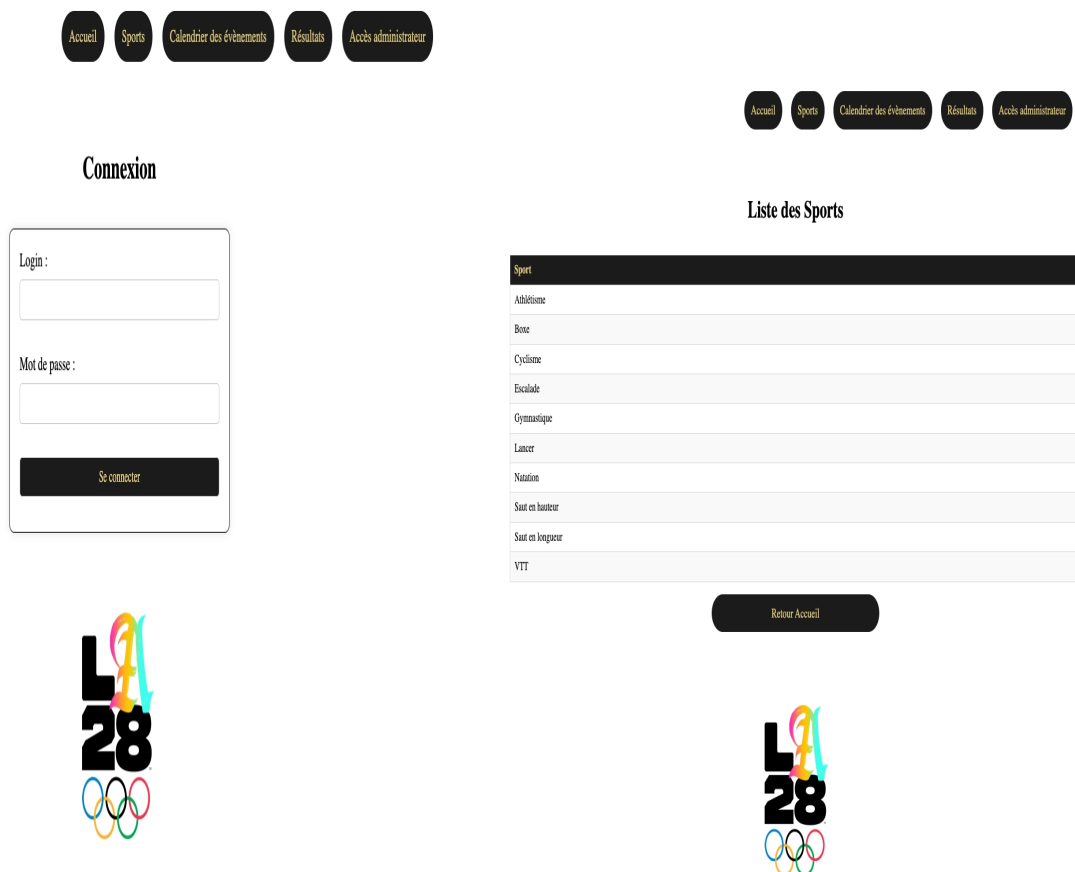
Versions Telephones :



5.1.2. Maquettes

Les maquettes utilisé pour le projet sont :

Les maquettes verion pc :





Les maquettes versions Tablettes / Telephones :

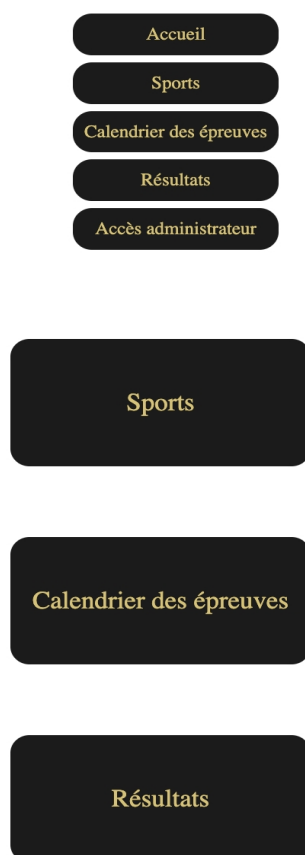


Connexion

Login :

Mot de passe :

Se connecter

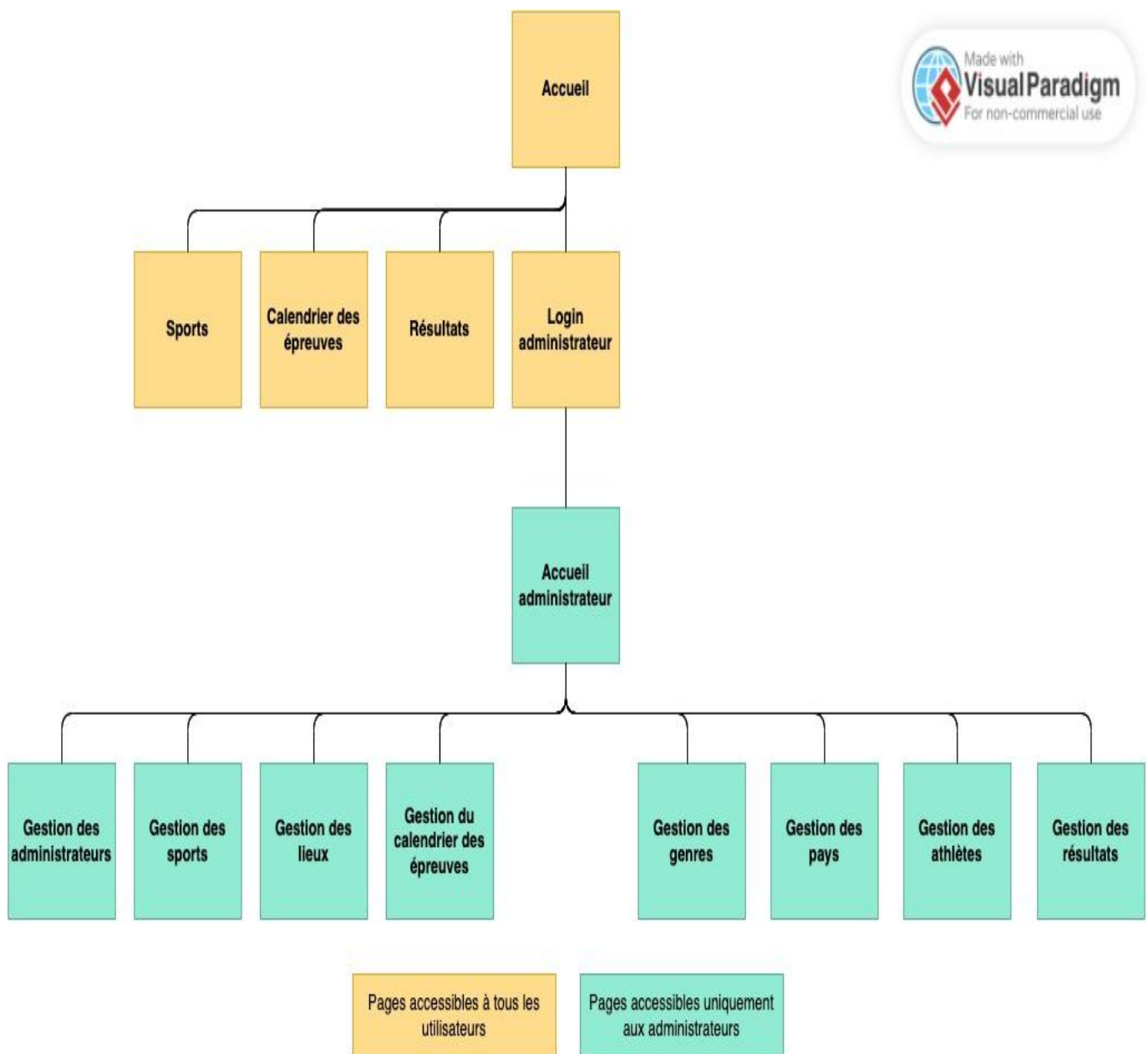


Liste des Sports

Sport
Athlétisme
Boxe
Cyclisme
Escalade
Gymnastique
Lancer
Natation
Saut en hauteur
Saut en longueur
VTT



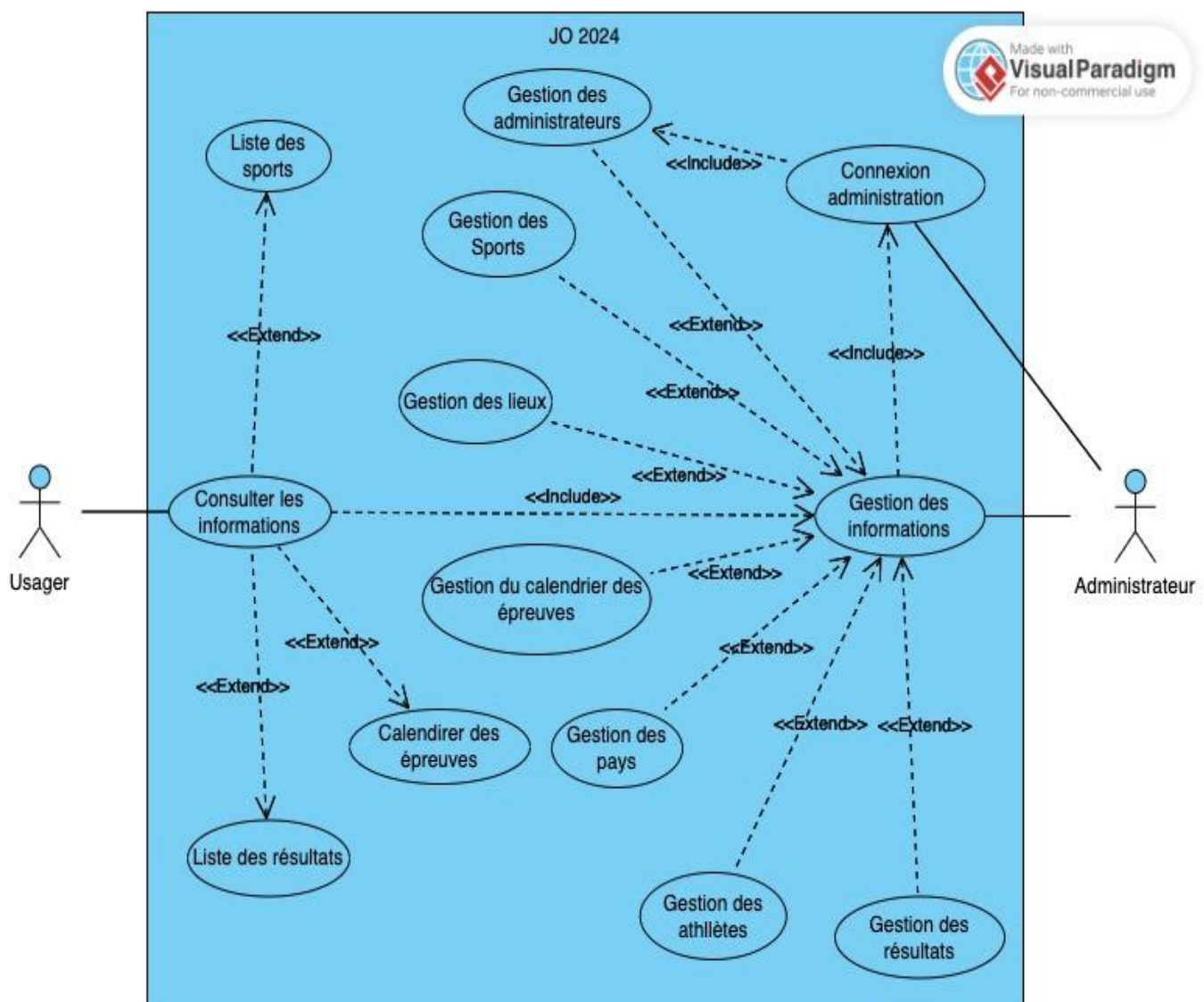
5.1.3. Arborescences



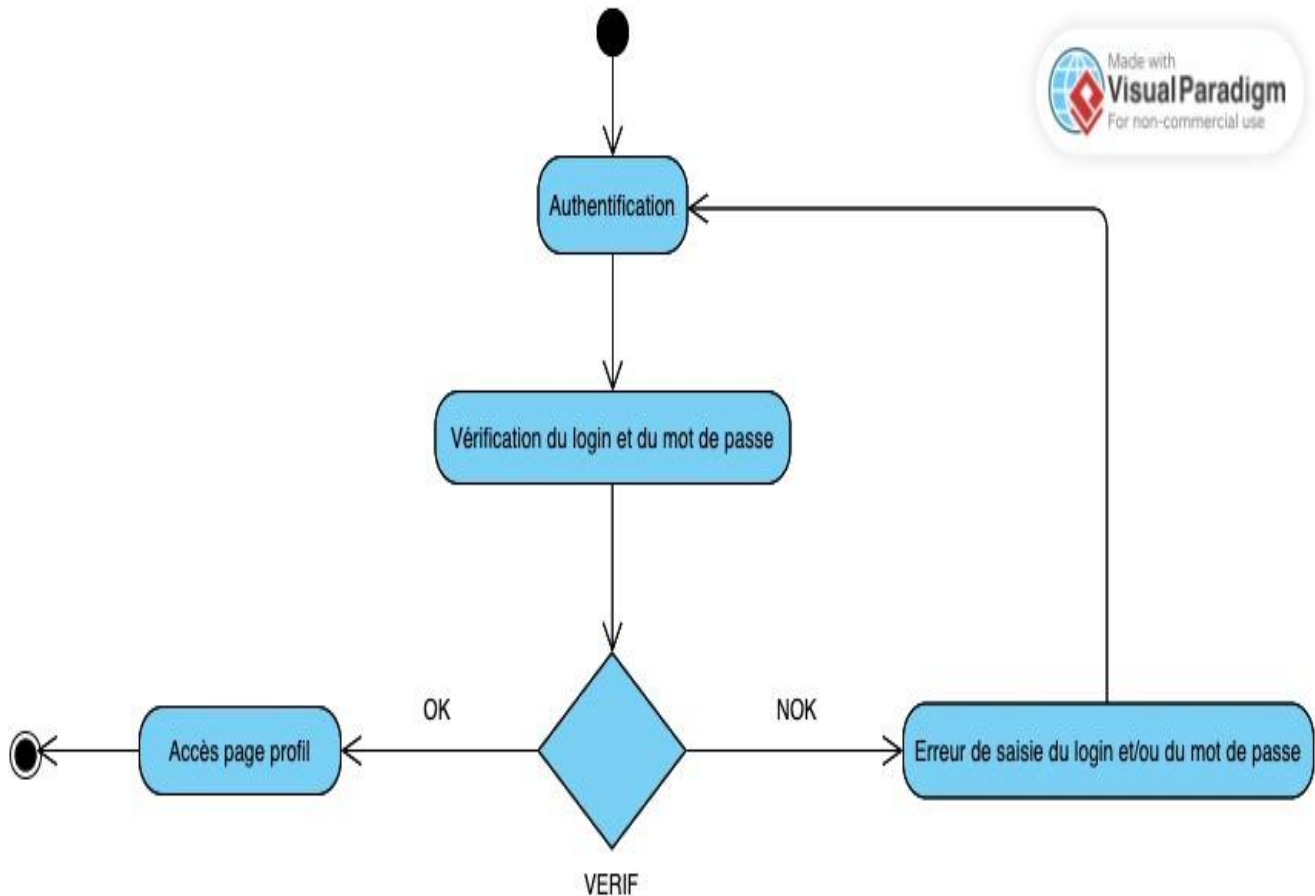
5.2. Le back-end

REPRENDRE RÉPONSES MISSIONS 2 ET 3

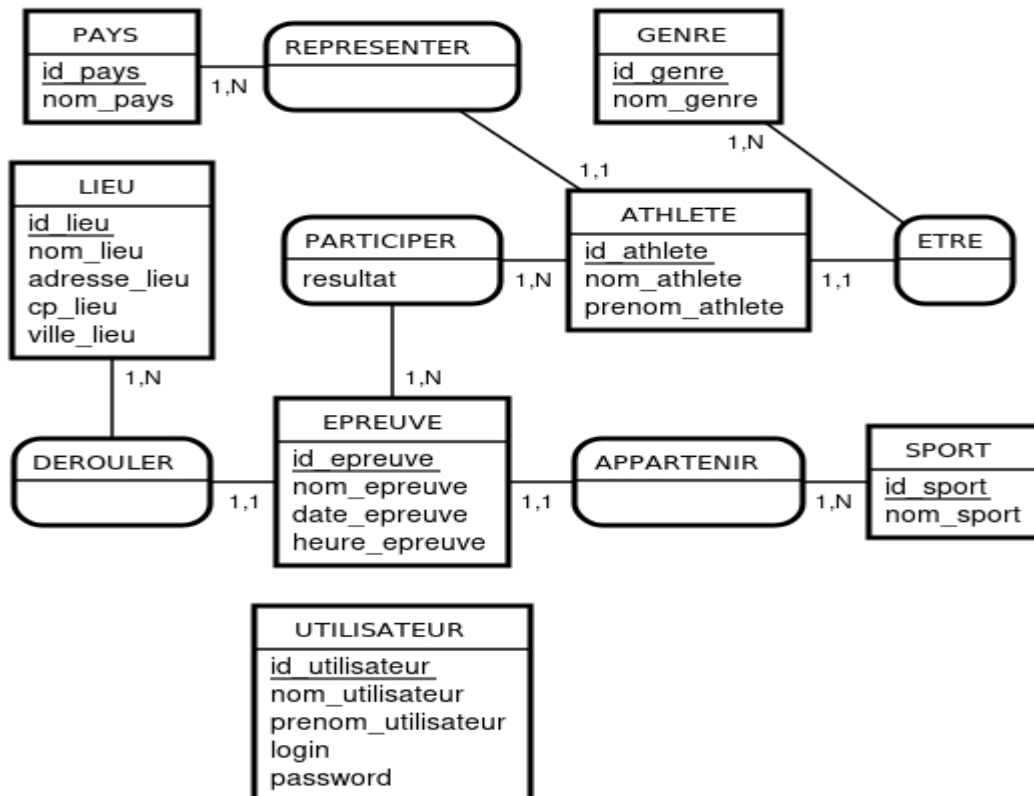
5.2.1. Diagramme de cas d'utilisation



5.2.2. Diagramme d'activités



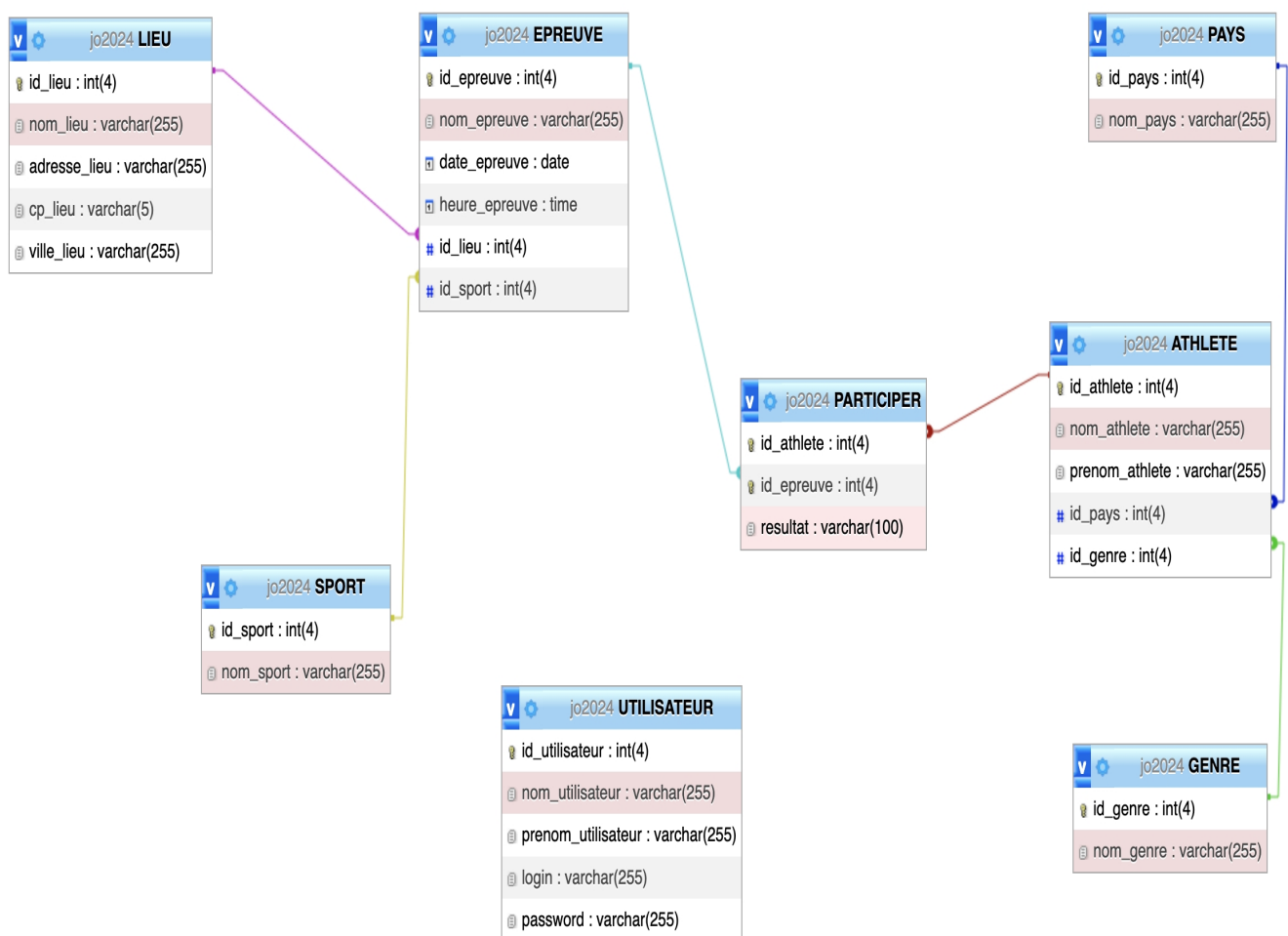
5.2.3. Modèles Conceptuel de Données (MCD):



5.2.4. Modèle Logique de Données (MLD)

- ATHLETE (id_athlete, nom_athlete, prenom_athlete, #id_pays, #id_genre)
- EPREUVE (id_epreuve, nom_epreuve, date_epreuve, heure_epreuve, #id_lieu, #id_sport)
- GENRE (id_genre, nom_genre)
- LIEU (id_lieu, nom_lieu, adresse_lieu, cp_lieu, ville_lieu)
- PARTICIPER (#id_athlete, #id_epreuve, resultat)
- PAYS (id_pays, nom_pays)
- SPORT (id_sport, nom_sport)
- UTILISATEUR (id_utilisateur, nom_utilisateur, prenom_utilisateur, login, password)

5.2.5. Modèle Physique de Données (MPD)



6. Technologies utilisées

REPRENDRE RÉPONSES MISSION 1

6.1. Langages de développement Web

Les langages de développement Web sont :

1. HTML5
2. CSS3
3. PHP8
4. JS
5. SQL pour bdd

6.2. Base de données

Pour la BDD nous allons utilisé du SQL

7. Sécurité

DÉFINISSEZ (SI POSSIBLE) ET EXPLIQUEZ BRIÈVEMENT VOTRE SOLUTION.

AJOUTEZ SI VOUS LE SOUHAITEZ DES COURS EXTRAITS DE CODE.

7.1. Login et protection des pages administrateurs

Une page Login est basé généralement sur un nom et un mdp a entré sur une page de connexion alors nous pourrions :

1. Utiliser un password_verify() pour comparer le mot de passe haché stocké avec celui entré par l'utilisateur.
2. bind_param() pour éviter les injections SQL
3. htmlspecialchars() pour empêcher des certains scripts
4. Utilisation de HTTPS pour chiffré les données
5. avoir un fichier logs pour savoir qui c'est connecté

7.2. Cryptage des mots de passe avec Bcrypt

Bcrypt est un algorithme de hachage pour protéger les mots de passe en les rendant difficiles à déchiffrer même en cas de fuite de base de données

```
<?php
require 'vendor/autoload.php'; // Inclure l'autoload de Composer

use Bcrypt;

// Hachage stocké (Exemple)
$hachage_stocke = '$2y$10$8j8jF8gfK.Xf'; // Exemple d'un hachage

// Mot de passe saisi par l'utilisateur
$mot_de_passe_saisi = 'MonMotDePasseSecurise123';

// Vérification du mot de passe

if (Bcrypt::check($mot_de_passe_saisi, $hachage_stocke)) {
    echo "Le mot de passe est correct, accès autorisé.";
} else {
    echo "Mot de passe incorrect, accès refusé.";
}
?>
```

7.3. Protection contre les attaques XSS (Cross-Site Scripting)

Les attaques XSS sont des failles de sécurité qui permettent à un attaquant d'injecter dans un site web un code client malveillant pour avoir accès et usurper généralement le compte des utilisateurs

1. Valider ou Échapper toutes les entrées des utilisateurs
2. Utiliser des politiques de Content Security Policy (CSP) exemple (interdire l'exécution de scripts inline.)
3. Utiliser des bibliothèques de sécurité comme htmlspecialchars()

7.4. Protection contre les injections SQL

L'injection SQL est une technique d'attaque qui sert à insérer des requêtes SQL malveillantes dans des champs d'entrée (Pages de connexion...) d'une application

1. On peut utiliser des instructions SQL comme les PDO
2. Valider et filtrer les entrées des utilisateurs
3. Code ci-dessous :

```

<?php
// Informations de connexion à la base de données
$host = 'localhost'; // ou l'adresse de votre serveur de base de données
$db = 'nom_de_la_base_de_donnees';
$user = 'utilisateur';
$pass = 'mot_de_passe';

try {
    // Création d'une nouvelle connexion PDO
    $pdo = new PDO("mysql:host=$host;dbname=$db;charset=utf8", $user, $pass);
    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION); // Gérer les
erreurs
} catch (PDOException $e) {
    echo "Erreur de connexion : " . $e->getMessage();
}
?>

```

Ici grace au PDO on va pouvoir gerer les erreurs et détecter des problèmes comme des violations sur les contraintes imposé