|  |
| --- |
|  |
| Paris 2K24 |
| Documentation technique |

|  |
| --- |
| Cheick TANGARA |

Table des matières

[**1.** **Description du projet** 2](#_Toc167266880)

[**2.** **Diagramme de cas d’utilisation / fonctionnalités du site** 2](#_Toc167266881)

[**3.** **Architecture logicielle et choix technologiques** 4](#_Toc167266882)

[**4.** **Diagramme de séquence de la fonctionnalité de réservation** 6](#_Toc167266918)

[**5.** **Modèle conceptuel de données** 7](#_Toc167266919)

[**6.** **Sécurité de l’application** 8](#_Toc167266920)

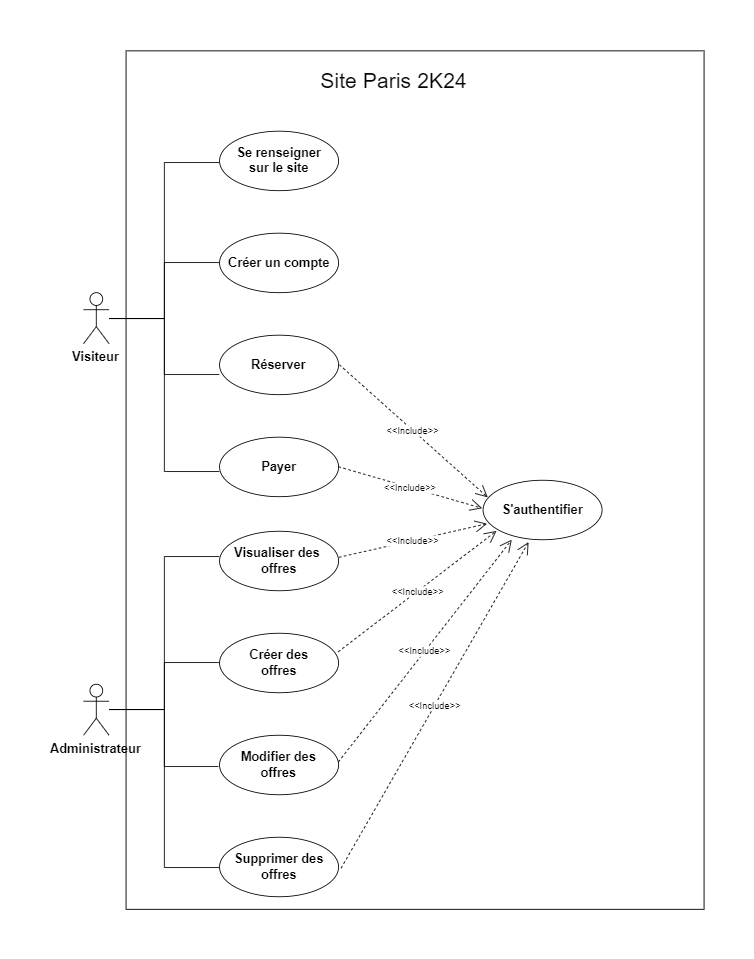
# **Description du projet**

Ce présent manuel est la documentation technique des Jeux Olympiques de Paris 2024. Le projet a été implémenté dans le but de mettre en place un site de réservation de e-ticket.

Le nom du site est Paris 2K24 et il contient plusieurs fonctionnalités. Un visiteur peut s’inscrire sur le site, se connecter ou se déconnecter, faire une réservation de tickets ou se renseigner sur différents sports. Le site est consultable sur ordinateur, tablette ou sur mobile.

# **Diagramme de cas d’utilisation / fonctionnalités du site**

La figure 1 présente le diagramme de cas d’utilisation du site et résume les concepts clés du besoin client. D’une part, on a les acteurs (clients et administrateurs) qui interagissent avec le site et, d’autre part, les cas d’utilisation qui correspondent aux fonctionnalités du site. De ce fait, un visiteur peut se renseigner sur le site (visualisation des offres, des sports etc.), créer un compte ou réserver un ticket. Quant à l’administrateur du site, un back office lui est dédié où il peut créer ou modifier des offres, en supprimer, visualiser le nombre de ventes par offre.

****

**Figure 1 :** Diagramme de cas d’utilisation du site Paris 2K24 montrant les interactions entre utilisateurs (visiteurs et administrateur du site) et les fonctionnalités.

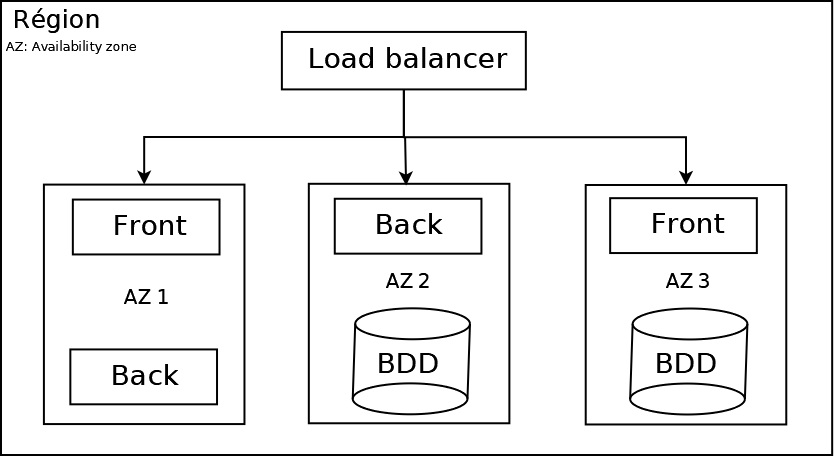
# **Architecture logicielle et choix technologiques**

La figure 2 présente l’architecture logicielle du site des Jeux Olympiques de Paris. C’est une architecture composée de deux **front-end**, de deux **back-end** et de deux **bases de données**.

Ces couches ou composants sont réparties entre trois **zones de disponibilité (Availability Zone** ou **AZ**) qui sont des data center situés dans une zone géographique ou région. Une région peut avoir au moins deux AZ.

Pour des raisons de sécurité, on n’a pas deux fois la même couche (2 front-end par exemple) dans une même zone de disponibilité ou AZ. Cela permettra d’assurer la continuité du service en cas de sinistre d’une AZ (les deux autres AZ assureront le service).

Enfin, le trafic pouvant s’intensifier lors des Jeux Olympiques, les serveurs vont être extrêmement sollicités. De ce fait, on peut utiliser un load balancer pour répartir la charge de travail entre les différents serveurs.



**Figure 2 :** Architecture du projet des Jeux Olympiques de Paris

Le tableau 1 résume les choix technologiques du projet ainsi que leurs pertinences :

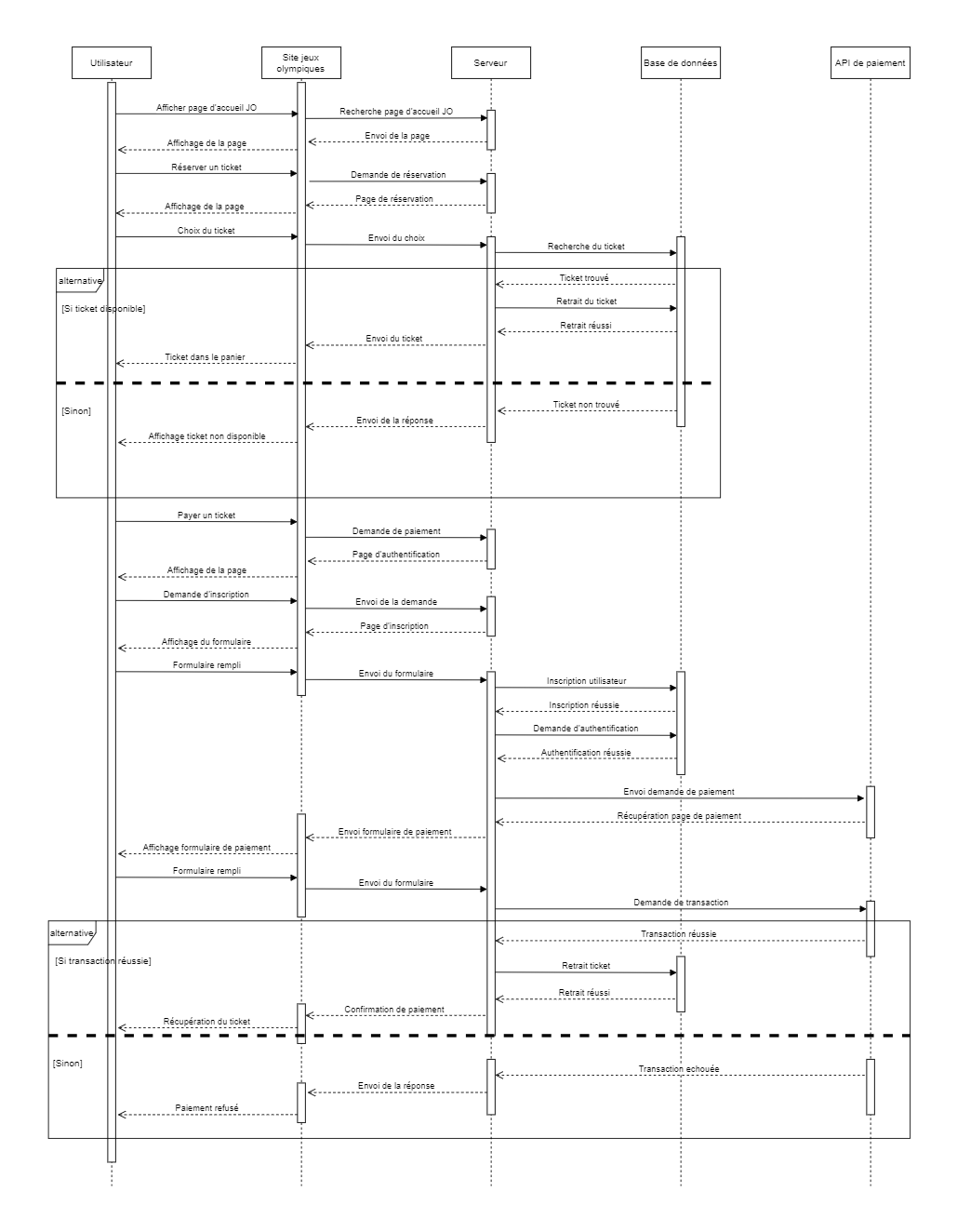
|  |  |  |
| --- | --- | --- |
| **Catégories d’outils** | **Choix technologiques** | **Pertinence** |
| **Interface de développement Intégré (IDE)** | Pycharm | * Automatisation des tâches. * Contient plusieurs modules et packages * Amélioration de la productivité * Compatible Windows, Linux, MacOS |
| **Framework** | Django | * Framework open source et gratuit * Développement de sites et d’application de façon simple, rapide et sécurisée * Inclut un back office * ORM puissant permettant de se passer des requêtes SQL * La sécurité est un point fort * Couvre à la fois le front-end (html, javascript et css) et le back-end avec python |
| **Système de gestion de base de données (SGBD)** | MySQL | * SGBD relationnel car nos données sont structurées * Populaire et fiable * Evolutif, rapide et facile d’utilisation |
| **Hébergement** | AWS d’Amazon | * Un des plus grands fournisseurs de service cloud * Garantit la sécurité des ressources * Bonne performance * Limite l’accès aux utilisateurs légitimes |
| **Solution de paiement** | * Paypal * Stripe * Carte bancaire | * Simplicité du paiement * Gain de temps * Sécurité * Flexibilité |

**Tableau 1 :** Choix technologiques du projet des Jeux Olympiques de Paris 2K24.

# **Diagramme de séquence de la fonctionnalité de réservation**

Le diagramme de séquence (figure 3) présente l’interaction entre l’utilisateur et les autres composants du site à savoir l’interface, le serveur, la base de données et l’application de paiement. Le scénario de réservation peut être décrit comme suit :

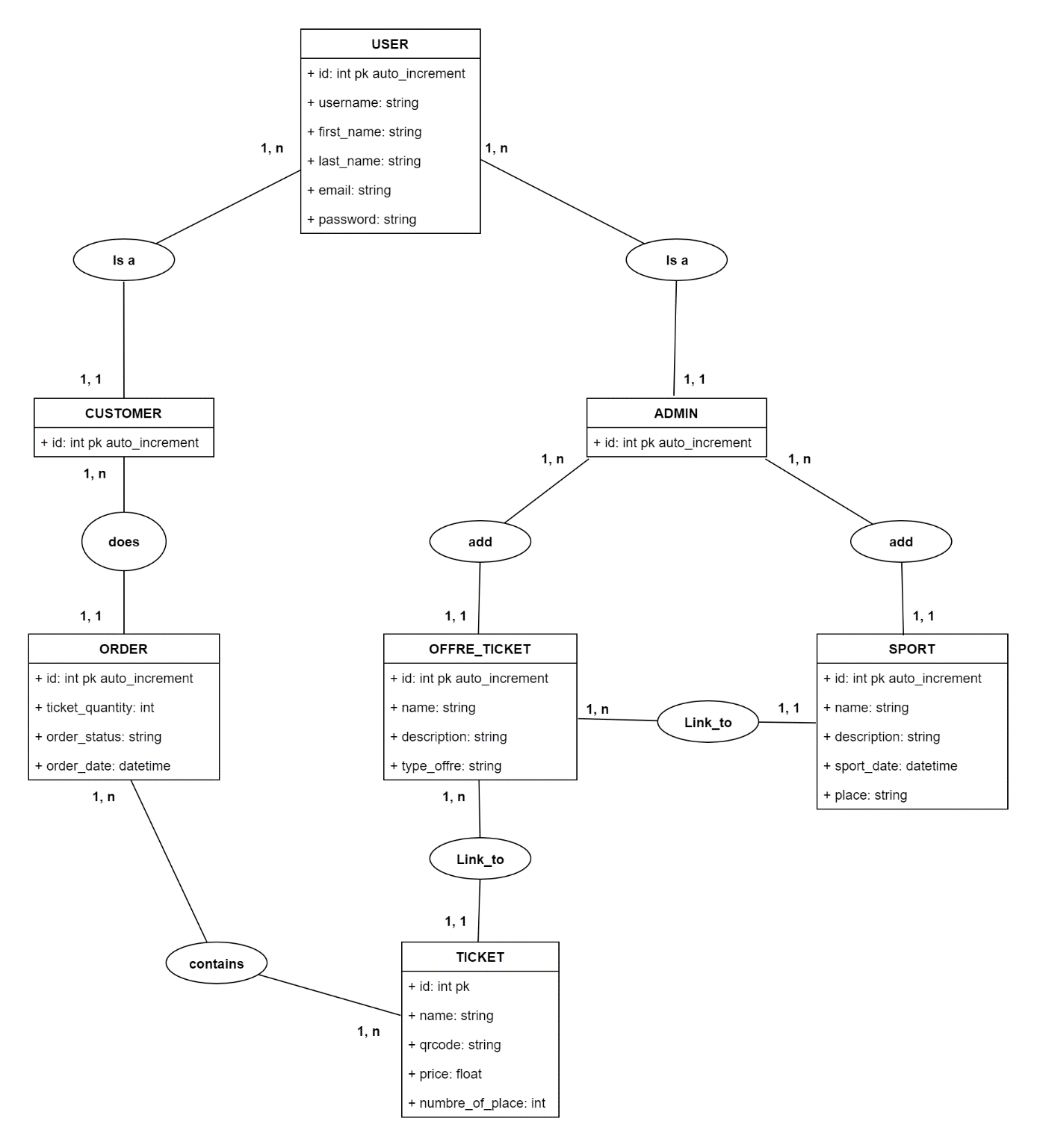
* L’utilisateur demande la page d’accueil du site et elle lui est affichée.
* Il choisit un ticket et met celui-ci dans son panier.
* Pour terminer sa réservation, l’utilisateur doit d’abord s’inscrire sur le site puis s’authentifier.
* Après inscription et authentification, l’utilisateur peut procéder au paiement de son billet.

****

**Figure 3 :** Diagramme de séquence sur la fonctionnalité de réservation.

# **Modèle conceptuel de données**

La figure 4 présente le modèle conceptuel de données. L’entité user a deux rôles : customer et admin. L’entité admin peut ajouter des offres de tickets ou des événements de sport dans la base de données tandis que le customer peut faire ses commandes de tickets.



**Figure 4 :** Modèle conceptuel de données du projet des Jeux Olympiques de Paris.

# Sécurité de l’application

Le tableau 2 présente les éléments à sécuriser dans l’application. Nous allons également mettre en place un plan de reprise après sinistre (DR, disaster recovery) pour restaurer l’accès à l’infrastructure informatique et ses fonctionnalités après un sinistre (naturel ou humain).

|  |  |
| --- | --- |
| **Eléments à sécuriser** | **Moyens utilisés** |
| **Compte utilisateur (identifiants, mot de passe, données bancaires etc.)** | * Authentification 2FA * Renouvellement fréquent du mot de passe avec caractères spéciaux, chiffres, lettres etc. * Blocage du compte après quelques tentatives d’authentification * Hachage du mot de passe * Sécurisation des sessions utilisateur * Contrôler/restreindre l’accès à une ressource ou une fonctionnalité. |
| **Le site web et ses infrastructures** | * Mise en place d’un pare-feu * Utilisation des protocoles HSTS, HTTPS * Eviter ou contrôler les composants tiers * Configurer des sauvegardes * Faire des pentest. |

**Tableau 2 :** Éléments à sécuriser dans l’application.