# Documentation technique pour Esportify.

## Réflexions initiale technologique sur le sujet.

### Analyse des besoins :

Les fonctionnalités nécessaires ont été définies en fonction des différents types d’utilisateurs et des exigences de la plateforme Esportify. Voici les fonctionnalités principales :

1. **Page d'accueil :**

Présentation de l’entreprise et galerie d’images sous forme de slide.

Affichage des évènements e-sport à venir et en cours, validés par un employé.

1. **Navigation (menu de l’application) :**

Accès rapide aux sections essentielles (accueil, évènements, connexion, contact).

Faciliter la navigation entre les différentes parties de la plateforme.

1. **Gestion des évènements :**

Vue globale de tous les évènements (titre, description, nombre de joueur, date et heure de début ainsi que date et heure de fin), avec filtres (nombre de joueurs, date, heure, pseudo).

Vue détaillée pour chaque évènement (titre, description, nombre de joueur, autorisé ou non en visibilité, date et heure de début ainsi que date et heure de fin, pseudo de l’utilisateur qui a proposé l’événement, une ou plusieurs images) avec possibilité de mettre en favori.

1. **Espace utilisateur :**

Système de connexion et d'inscription sécurisé (pseudo unique, email, mot de passe sécurisé avec une majuscule, une minuscule, un chiffre et un caractère spécial minimum), avec validation des critères de mot de passe.

Accès à un espace personnel avec :

Gestion des favoris et bouton de « rejoindre » un évènement.

Option pour créer un nouvel évènement via un formulaire et suivre son statut (validé, en attente de validation, non validé).

Historique des évènements proposés et de leurs statuts, et historique des scores obtenus.

1. **Espace employé :**

Interface pour examiner et approuver ou refuser les évènements soumis par les utilisateurs. Avec la possibilité de suspendre un évènement si nécessaire.

1. **Espace administrateur :**

Gestion des comptes employés et utilisateurs (création, modification, blocage, suppression). Outil de visualisation avec un graphique des évènements par jour. Accès aux mêmes fonctionnalités de gestion des évènements que l’employé.

1. **Chat en temps réel :**

Disponible pour les participants lors des évènements. Utilisation d’une base de données NoSQL pour gérer les échanges en temps réel.

1. Déploiement :

Possibilité de déployer l’application sur une plateforme d'hébergement pour tester les fonctionnalités dans un environnement de production simulé.

### Technologies choisies :

*Front-end :*

**HTML5 :** J'ai choisi HTML5 pour structurer les pages et assurer un rendu sémantique et accessible de l'application. HTML5 permet également de mieux organiser les balises pour un bon référencement, ce qui est essentiel pour une application en ligne.

**CSS :** CSS est utilisé pour styliser l’interface et la rendre visuellement attrayante et responsive, en s’adaptant aux différentes tailles d’écran. J’utilise CSS pour optimiser l’expérience utilisateur et garantir une cohérence visuelle sur toutes les pages.

**JavaScript :** JavaScript est nécessaire pour ajouter des éléments interactifs et dynamiques, comme les animations et les filtres. JavaScript permet également de mettre en place un filtre avancé sur les évènements, en fonction du nombre de joueurs, de la date et du pseudo.

*Back-end :*

**MySQL :** J'ai opté pour MySQL comme base de données relationnelle pour stocker les informations structurées, telles que les utilisateurs, les évènements et les favoris. MySQL est une solution fiable, bien documentée, et offre une compatibilité facile avec PHP, ce qui simplifie les interactions entre la base de données et l'application.

*Outils de développement :*

**Visual Studio Code :** J’utilise Visual Studio Code pour son interface intuitive, sa large gamme d’extensions (qui facilite le débogage et l’auto-complétions du code), et son terminal intégré qui centralise le développement.

**XAMPP (Apache et MySQL) :** XAMPP est choisi pour simuler un environnement serveur en local. Il comprend MySQL et PhpMyAdmin, qui simplifient la gestion de la base de données, ainsi qu’Apache pour héberger l’application en développement.

*Contrôle de version et gestion de projet :*

**Outil Kanban (Miro) :** Un tableau Kanban est utilisé pour structurer les tâches, suivre l’avancement des fonctionnalités et prioriser les développements, ce qui est essentiel pour un projet de cette envergure. J’utilise Miro, une plateforme visuelle collaborative, car elle offre une grande maniabilité et une interface intuitive qui facilite l’organisation des tâches. Miro permet également de collaborer en temps réel avec d’autres membres du projet et de visualiser facilement les dépendances entre tâches, ce qui améliore l'efficacité du travail en équipe et le suivi de l’avancement du projet.

Ces choix technologiques ont été faits pour répondre aux besoins fonctionnels spécifiques du projet tout en assurant performance, facilité d’utilisation et sécurité.

### Contraintes et justifications :

**Performance et réactivité**

* **Contrainte** : L’application doit offrir une expérience fluide, notamment pour des fonctionnalités comme le chat en temps réel et la navigation entre les pages.

**Sécurité des données**

* **Contrainte** : Les informations sensibles, comme les identifiants et les mots de passe des utilisateurs, doivent être stockées et traitées de manière sécurisée.

**Flexibilité de l’interface utilisateur**

* **Contrainte** : L’application doit être utilisable sur différentes tailles d’écran, notamment mobile et desktop, pour s’adapter aux besoins des utilisateurs.
* **Justification** : J’utilise **HTML5 et CSS** pour la structure et le style de l’interface, en appliquant des techniques de design responsive. CSS facilite l’adaptation de la mise en page en fonction de la taille de l’écran, garantissant ainsi une expérience utilisateur uniforme.

**Facilité de maintenance et évolutivité**

* **Contrainte** : L’application doit être facile à maintenir et à faire évoluer, car elle pourrait nécessiter des ajouts de fonctionnalités (par exemple, nouveaux types d’évènements ou rôles utilisateurs).
* **Justification** : J’ai choisi **MySQL** pour la base de données relationnelle, car sa structure tabulaire et ses clés étrangères permettent une bonne organisation et gestion des relations entre les données, facilitant les modifications ultérieures.

**Déploiement rapide et test en production simulée**

* **Contrainte** : Il est important de pouvoir déployer l’application en ligne pour effectuer des tests dans un environnement de production simulée.

**Collaboration et gestion de projet**

* **Contrainte** : Le projet doit être bien structuré et permettre un suivi clair de l'avancement des fonctionnalités.
* **Justification** : L'utilisation de **Miro** pour le Kanban facilite la gestion des tâches et offre une vue d'ensemble des fonctionnalités en cours, planifiées, et terminées. Miro est particulièrement apprécié pour son interface visuelle et sa maniabilité, ce qui aide à maintenir une organisation fluide et collaborative du projet.

## Configuration de votre environnement de travail.

### Outils et Logiciels :

En premier, j’ai choisi de me concentrer sur la gestion de projet et l’organisation des tâches, j’ai utilisé Miro pour créer le Kanban. Miro est une plateforme visuelle collaborative, parfaite pour le suivi de projets. Elle propose une interface fluide et intuitive, qui rend l’organisation des tâches très accessible. J’ai pu structurer les tâches en colonnes, suivre l’avancement de chaque fonctionnalité, et ajouter des notes ou des commentaires pour clarifier certaines étapes. La possibilité de collaborer en temps réel permet aussi de faciliter les retours et les ajustements dans un environnement professionnel en groupe, ce qui est essentiel pour un projet de cette envergure. En choisissant ce site, j’ai optimisé la gestion de projet en ayant une vue d’ensemble et en favorisant un flux de travail clair et efficace.

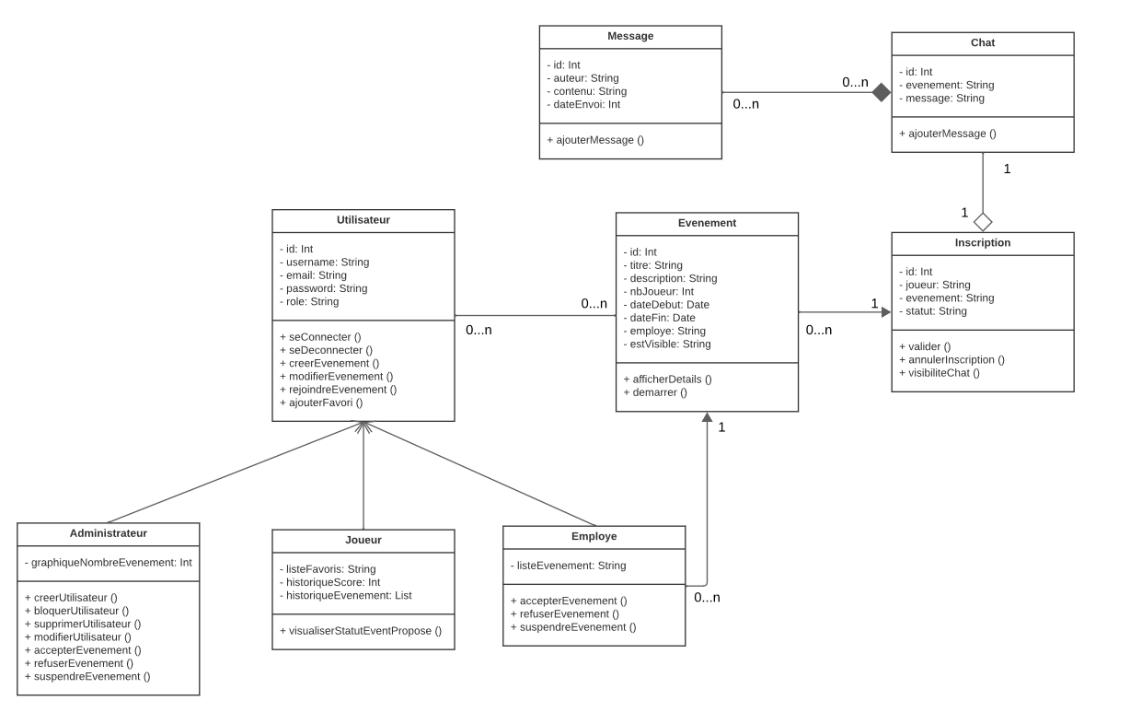
En second, l’élaboration des diagrammes, j’ai opté pour Lucidchart. Ce choix s’est imposé pour plusieurs raisons : Lucidchart est spécialisé dans la création de diagrammes UML, ce qui est idéal pour les diagrammes de classe, de séquence, et de flux de travail requis par le projet. Son interface glisser-déposer et sa bibliothèque de formes facilitent la construction de diagrammes précis et professionnels. De plus, Lucidchart propose des fonctionnalités de collaboration, ce qui permet d’échanger des commentaires et d’effectuer des modifications en équipe. Grâce à Lucidchart, j’ai pu créer des diagrammes clairs et bien structurés, essentiels pour une bonne compréhension de l’architecture technique et des interactions dans le projet.

En troisième, la création des maquettes, j’ai choisi le site Figma. Cet outil offre une interface intuitive et des fonctionnalités puissantes pour le design d’interfaces utilisateur. Figma permet de travailler de manière collaborative, ce qui facilite les retours et les ajustements en temps réel si nécessaire dans un environnement professionnel. Son système de composants réutilisables est aussi très efficace pour assurer la cohérence visuelle de l’application sur plusieurs pages et pour des itérations rapides. En utilisant Figma, j’ai pu concevoir des maquettes interactives, ce qui donne une représentation fidèle de l’expérience utilisateur attendue.

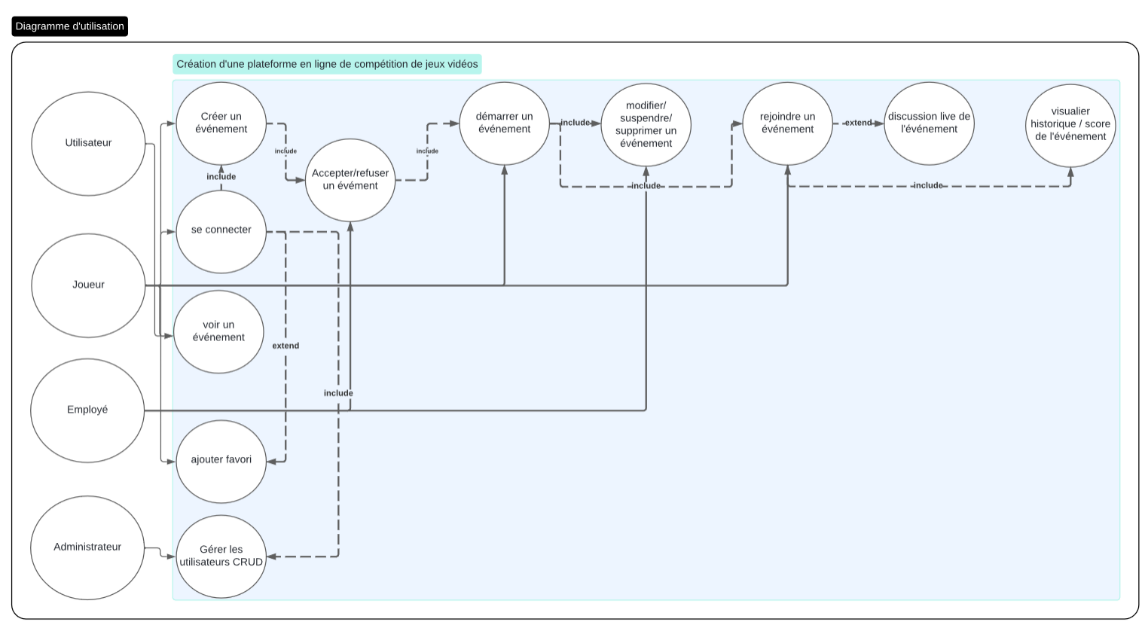
Enfin, pour la Base de Données Relationnelle, J'ai installé XAMPP pour créer un environnement de serveur Apache et MySQL en local. XAMPP permet de gérer facilement les services nécessaires, comme Apache pour l'hébergement de l’application web et MySQL pour la base de données, le tout en un seul package. J’utilise MySQL Workbench et PhpMyAdmin pour la gestion de la base de données. MySQL Workbench permet une gestion graphique avancée des bases de données, tandis que PhpMyAdmin, accessible via XAMPP, offre une interface simple et rapide pour des opérations courantes.

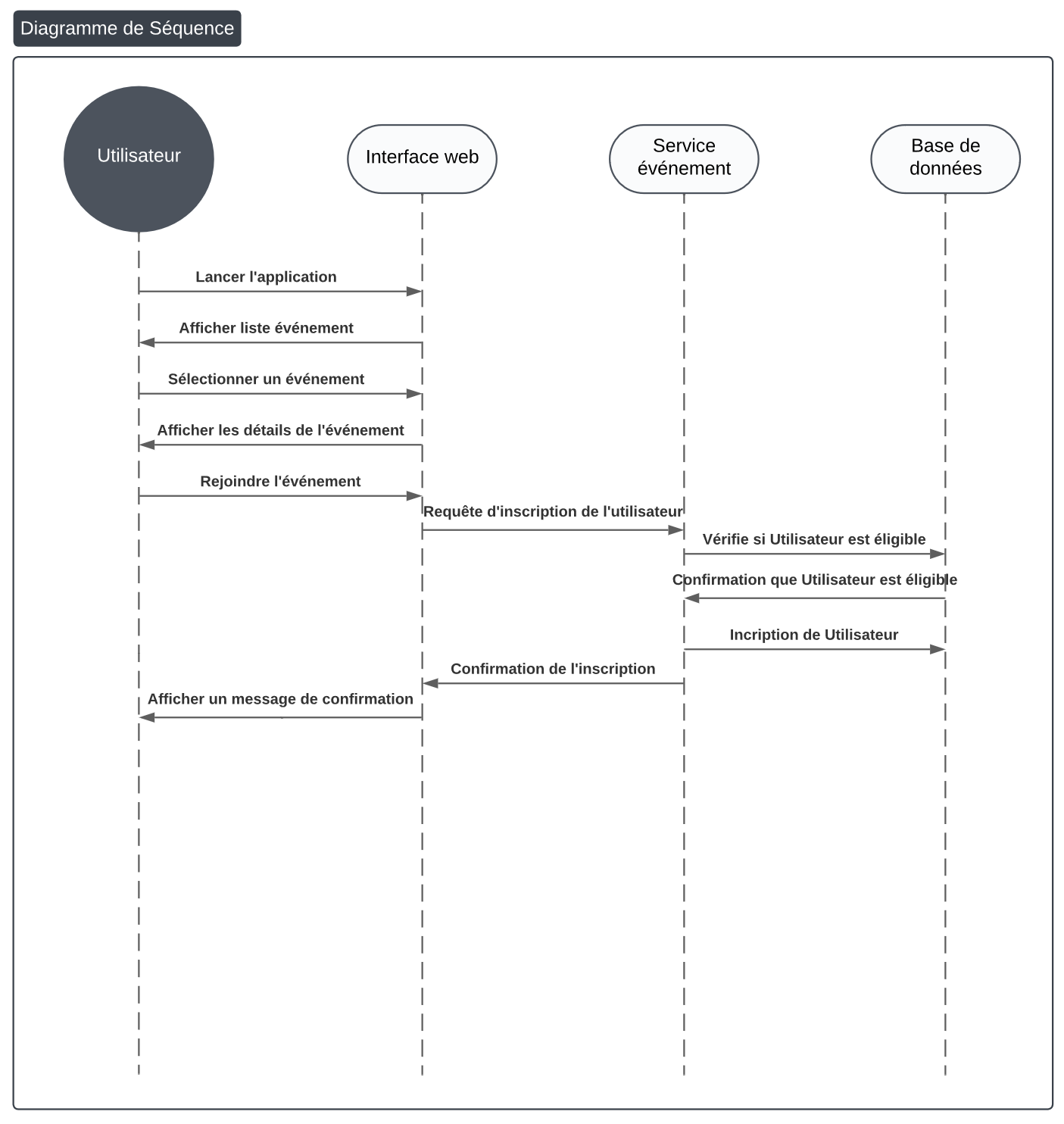
J'utilise Visual Studio Code pour le développement du front-end avec les langages HTML5, CSS et JavaScript. Visual Studio Code est un éditeur puissant et flexible, offrant une interface conviviale ainsi que de nombreuses extensions qui facilitent l'écriture de code. Par exemple, il propose des extensions pour l'autocomplétions et la validation syntaxique, ce qui aide à réduire les erreurs et à améliorer l'efficacité du développement. Il offre également des outils intégrés de débogage et un terminal, ce qui permet de tester et d'optimiser le code directement dans l'éditeur. Ce choix me permet donc de centraliser tout le travail de développement dans un seul outil performant. Petite mention spéciale pour Microsoft Edge Tools, qui permet d’ouvrir directement un visuel sur Microsoft Edge.

## Modèle conceptuel de données (ou diagramme de classe).



## Diagramme d’utilisation ainsi que le diagramme de séquence.





## Explication du déploiement de l’application.

## Liens.

<https://miro.com/welcomeonboard/dXFPQXc4QUE1QmlNQ1RDYUkvMFR5MTZXQ2E5SkV1RG5NbFhFMWtJMUhaUWpPUy9FRTJVcmE2bjU0OVVBUllhQUtVOVFnc1p6Yi9xVGFzQ3pmTk5sL0hxQ1FxZkJ2L1JRMk1MdXJrT3l2S3BnbFNGQ2FHUzkxMjdTZEhFdzZrT0UhZQ==?share_link_id=541466977410>

Lien du Kanban partagé afin d’avoir une vue globale de ma réalisation.