CAHIER DES CHARGE

Pour l’entreprise StreamIO

ALLAN BIROLINI

Cahier des charges (StreamIO – Web)

Sommaire :

Présentation Générale

Nom du projet :

Le nom de notre projet est Book IO.

Contexte :

L'entreprise StreamIO dispose de plusieurs salles équipées mais actuellement sous-utilisées. Afin de maximiser leur usage et d'optimiser la gestion de ces espaces, elle fait appel à nous pour concevoir une solution digitale. L'objectif est de mettre en place une application qui facilite la réservation, la communication entre les utilisateurs, et la gestion en temps réel de la disponibilité des salles. Ce projet s'inscrit dans une volonté de modernisation et de valorisation des infrastructures existantes de l'entreprise.

Objectif :

L'application a pour but de permettre la réservation des salles de l'entreprise, ainsi que du matériel mis à disposition dans ces espaces (vidéoprojecteurs, micros, etc.). Elle sera utilisable par deux types d'utilisateurs :

- Les utilisateurs classiques, qui pourront consulter les disponibilités, réserver une salle et sélectionner le matériel nécessaire à leurs besoins.
- Les administrateurs, qui auront un accès plus complet leur permettant de créer, modifier ou supprimer des salles, gérer les réservations, et ajouter ou retirer du matériel selon les besoins de l'entreprise.

L'application vise à centraliser la gestion des espaces et des ressources matérielles pour éviter les conflits de réservation et les pertes de matériel. Elle permettra également de suivre l'historique des réservations et d'obtenir une vision globale de l'occupation des salles à tout moment. À terme, elle contribuera à une meilleure organisation interne, en fluidifiant les échanges entre les collaborateurs et en optimisant l'utilisation des infrastructures disponibles.

Cible utilisateur

Public visée :

L'application s'adresse principalement aux employés de l'entreprise StreamIO, qu'ils soient membres de l'équipe technique, administrative ou commerciale. Elle est pensée

pour être utilisée aussi bien par des utilisateurs non techniques, souhaitant simplement réserver une salle pour une réunion ou un événement, que par des responsables ou administrateurs, chargés de gérer les espaces et le matériel mis à disposition. L'interface devra donc rester simple, intuitive et accessible à tous, quel que soit le niveau de compétence informatique de l'utilisateur.

Fonctionnalités principales

L'application propose un ensemble de fonctionnalités destinées à simplifier la gestion des salles et du matériel dans l'entreprise. Chaque utilisateur pourra s'authentifier via un système sécurisé (inscription, connexion, déconnexion) afin d'accéder à l'interface de réservation.

Les utilisateurs classiques auront la possibilité de :

- Consulter les disponibilités des salles en temps réel
- Effectuer une réservation en sélectionnant une plage horaire et le matériel nécessaire
- Modifier ou annuler leurs réservations
- Accéder à leur profil, qui regroupe leurs informations personnelles et leur historique de réservations

Les administrateurs, quant à eux, disposeront de fonctionnalités supplémentaires, notamment :

- Créer, modifier ou supprimer une salle
- Gérer le matériel disponible dans chaque salle
- Consulter et gérer l'ensemble des réservations
- Attribuer ou retirer des droits d'accès

Maquette / Design

Ambiance visuelle :

L'application adopte un design en dark mode, avec un fond noir profond, apportant une esthétique moderne, sobre et reposante pour les yeux, notamment lors d'utilisations prolongées. Les boutons et éléments interactifs sont orange, ce qui crée un contraste marqué, attire naturellement l'attention de l'utilisateur, et améliore la lisibilité globale de

l'interface. Ce choix de couleurs, à la fois simple et percutant, permet aussi de construire une identité visuelle forte et reconnaissable, tout en garantissant une bonne accessibilité visuelle.

L'interface est pensée pour être intuitive et fluide, avec une hiérarchie claire des éléments et des transitions agréables. Elle s'adapte à tous les types de supports (ordinateur, tablette, smartphone), pour offrir une expérience utilisateur optimale sur tous les appareils, grâce à un design 100% responsive. Chaque composant est conçu pour faciliter l'interaction, limiter la friction, et encourager une prise en main rapide, même pour les utilisateurs peu expérimentés.

Inspiration /référence :

Le design s'inspire d'interfaces modernes en dark mode comme Discord, Spotify, ou encore certaines vues épurées de Notion, qui ont su allier simplicité, efficacité et esthétique dans leur conception. Ces plateformes montrent l'importance d'un design pensé pour l'utilisateur final, avec une navigation fluide, des espaces bien organisés et une expérience sans surcharge visuelle.

L'application adopte cette philosophie en mettant l'accent sur une ergonomie claire, une navigation intuitive et des interactions légères mais élégantes. Des animations douces, comme des fondus ou des survols subtils, pourront être intégrées pour renforcer le confort d'utilisation et dynamiser l'interface sans la complexifier.

Enfin, les références graphiques choisies permettent aussi de guider les décisions sur la typographie, les espacements, ou encore l'organisation des blocs d'information, pour garantir une cohérence visuelle et fonctionnelle sur l'ensemble de l'application.

Technologie utilisée

L'application repose sur un socle technique moderne, robuste et scalable, adapté aux besoins d'une application web professionnelle. Le framework principal utilisé est Next.js (App Router), qui offre un système de routage efficace, une gestion automatique du rendu côté serveur (SSR) et une structure claire pour le développement en composants.

Le langage utilisé est TypeScript, qui ajoute une couche de typage statique au JavaScript, permettant de réduire les erreurs en amont, de faciliter la lisibilité du code, et d'améliorer l'expérience développeur grâce à une autocomplétions plus précise.

Pour la couche visuelle, l'interface est construite avec Tailwind CSS, un framework de classes utilitaires très flexible, permettant de créer des interfaces modernes sans avoir à écrire du CSS classique. À cela s'ajoute shadcn/ui, une bibliothèque de composants déjà stylisés et faciles à personnaliser, qui aide à maintenir une cohérence graphique dans tout le projet tout en gagnant du temps de développement.

L'authentification des utilisateurs est gérée via BetterAuth, une solution moderne et flexible qui permet une personnalisation avancée de la logique d'authentification. Pour la sécurité, les mots de passe sont chiffrés avec scrypt, un algorithme de dérivation de clé robuste conçu pour résister aux attaques par force brute, assurant la confidentialité des données utilisateurs.

Côté base de données, l'application repose sur SQLite, une base de données légère, rapide et facile à mettre en place, particulièrement adaptée aux projets en phase de développement ou aux applications avec un volume de données modéré. Elle ne nécessite pas de serveur dédié, ce qui simplifie le déploiement et réduit les dépendances.

Pour l'interaction avec la base, l'application utilise Drizzle ORM, un ORM moderne et typé qui s'intègre parfaitement avec TypeScript. Il permet de construire des requêtes SQL de manière déclarative, tout en bénéficiant d'une excellente sécurité type-safe et d'une très bonne lisibilité du code. Drizzle offre également une bonne expérience développeur avec des migrations claires et une gestion fine des modèles de données.

Le projet utilise pnpm pour la gestion des dépendances, assurant des installations rapides et efficaces, particulièrement utiles dans les environnements complexes. Enfin, tout le projet est conteneurisé avec Docker, garantissant une portabilité maximale, une configuration reproductible, et une mise en production plus rapide et plus stable.

Contrainte

Délais :

Le développement de l'application s'est inscrit dans un cadre temporel défini par une deadline imposée, dans le cadre d'un projet encadré. Le travail a été réalisé en binôme, permettant une meilleure répartition des tâches et une progression plus fluide à chaque étape du projet.

Le projet a été découpé en plusieurs phases :

- Une première phase de conception (définition des besoins, rédaction du cahier des charges, création des maquettes)
- Une seconde phase de développement (mise en place du back-end, de la base de données, de l'authentification, puis des interfaces)
- Et une dernière phase dédiée aux tests, débogage et améliorations finales.

La gestion du temps a été essentielle pour respecter les différentes échéances, en veillant à avancer régulièrement sur les différents modules tout en gardant une cohérence d'ensemble. La collaboration à deux a permis de gagner en efficacité, notamment lors des phases critiques comme l'intégration des fonctionnalités principales.

Sécurité :

La sécurité est un aspect crucial du développement de cette application, notamment en ce qui concerne la gestion des utilisateurs et la confidentialité des données. Voici les principaux mécanismes mis en place pour garantir la sécurité de l'application :

- Authentification et gestion des sessions :

  L'application utilise BetterAuth pour gérer les flux d'authentification de manière sécurisée, avec des mécanismes de protection des sessions et de gestion des permissions. L'authentification est renforcée par l'utilisation de scrypt, un algorithme de chiffrement robuste et résistant aux attaques par force brute. Les mots de passe des utilisateurs sont donc stockés de manière sécurisée, réduisant les risques de compromission.

- Protection contre les attaques courantes :

  Des mécanismes de protection contre les attaques par injection SQL sont mis en place grâce à l'utilisation de Drizzle ORM, qui permet d'éviter les injections malveillantes en générant automatiquement des requêtes sécurisées.

L’application est également protégée contre les attaques XSS (Cross-Site Scripting), qui peuvent survenir si un attaquant parvient à injecter du code malveillant dans le contenu de la page. En utilisant des bonnes pratiques de sanitisation des données et des mécanismes de validation côté serveur, nous minimisons ces risques.

- Contrôle des accès :

  Un système de contrôle d’accès basé sur les rôles est mis en place, permettant de différencier les droits d’accès entre les utilisateurs normaux (qui peuvent réserver des salles et du matériel) et les administrateurs (qui ont la possibilité de gérer la configuration des salles, le matériel, et les réservations). Cela garantit que seules les personnes autorisées peuvent accéder à des fonctionnalités sensibles.

- Audit et traçabilité :

  Toutes les actions sensibles, telles que les tentatives de connexion, les modifications de données utilisateurs ou la gestion des réservations, sont enregistrées dans un journal d’audit. Cela permet de garder une trace de toute activité sur l’application et d’identifier rapidement toute action suspecte.

Critère de validation

Les critères de validation de l’application sont définis en fonction des exigences fonctionnelles, des performances, de la sécurité, ainsi que de l’expérience utilisateur. À chaque étape du développement, des tests rigoureux seront effectués pour garantir la conformité avec les spécifications.

- Validation fonctionnelle :
  - Réservation de salle et matériel : Les utilisateurs doivent pouvoir réserver une salle et du matériel via l’interface, en s’assurant que les créneaux sont disponibles. Les réservations doivent être correctement enregistrées dans la base de données.
  - Création, modification et suppression de salles : Les administrateurs doivent pouvoir créer, modifier et supprimer des salles, avec une gestion en temps réel des disponibilités et du matériel.
  - Authentification sécurisée : L’application doit valider que les utilisateurs peuvent se connecter avec des identifiants uniques et sécurisés. Le processus d’authentification doit être conforme aux bonnes pratiques de sécurité.
- Validation des performances :
  - L’application doit répondre de manière fluide et rapide, même avec plusieurs utilisateurs simultanés. Des tests de performance seront effectués pour valider que les requêtes, notamment celles liées aux réservations et à la gestion des utilisateurs, sont traitées efficacement.
  - Le temps de chargement des pages ne doit pas dépasser 2 secondes pour l’utilisateur moyen.

- Validation de la sécurité :
    - L’application doit être protégée contre les attaques courantes telles que les injections SQL, les attaques XSS, ainsi que les attaques par force brute.
    - Les mots de passe doivent être correctement chiffrés et les sessions doivent être sécurisées.
    - Les données sensibles (comme les informations personnelles des utilisateurs) doivent être stockées et transmises de manière sécurisée.
- Tests d’acceptation utilisateur (UAT) :
    - Un ensemble de tests sera effectué pour s’assurer que l’application est facile à utiliser, intuitive et répond aux besoins des utilisateurs finaux. Les tests d’acceptation incluront des scénarios de réservation de salle, de gestion de matériel, et de gestion des utilisateurs.
    - Les retours des utilisateurs de test seront utilisés pour améliorer l’interface et l’expérience utilisateur, afin de garantir une navigation fluide et intuitive.
- Tests de compatibilité :
    - L’application doit être compatible avec les principaux navigateurs modernes (Chrome, Firefox, Safari, Edge), ainsi que sur des appareils mobiles et tablettes. Des tests de compatibilité croisée seront réalisés pour vérifier la bonne adaptation de l’application à différentes résolutions d’écran et plateformes.