

MVP - Projet de planification des agents de sécurité

Table des Matières

1. [Introduction](#)
 - 1.1. [Problématique](#)
 - 1.2. [Objectifs du projet](#)
2. [Membres de l'équipe](#)
 - 2.1. [Léo Zmoos](#)
 - 2.2. [Dylan Oliveira Ramos](#)
3. [Présentation de l'idée](#)
4. [Secteur de marché](#)
5. [Public cible](#)
 - 5.1. [Personae](#)
6. [Questionnaires et hypothèses à valider avec le public cible](#)
 - 6.1. [Objectifs des entretiens](#)
 - 6.2. [Questions pour les planificateurs](#)
 - 6.3. [Questions pour les agents](#)
 - 6.4. [Hypothèses à valider](#)
7. [Analyse des retours du public cible](#)
 - 7.1. [Retours des planificateurs](#)
 - 7.2. [Retours des agents](#)
8. [Maquettes](#)
 - 8.1. [Wireframes](#)
 - 8.2. [Mockups](#)
9. [Choix technologiques](#)
 - 9.1. [Résumé](#)
10. [Tests utilisateurs](#)
 - 10.1. [Changement majeur](#)
11. [Pérennité du projet](#)
12. [Difficultés rencontrées et solutions](#)
 - 12.1. [Prise en main de Next.js](#)
 - 12.2. [Authentification](#)
13. [Conclusion](#)

1. Introduction

1.1. Problématique

Une société de sécurité a un planificateur responsable d'attribuer des plages horaires à des agents. Le problème est que le planificateur reçoit les plages horaires de tous les agents par message sur WhatsApp. Cela lui fait perdre un temps considérable car il doit manuellement gérer les disponibilités et contraintes des agents, notamment en utilisant un fichier Excel.

En une phrase, la problématique est la **complexité dans la planification horaire des agents de sécurité**.

1.2. Objectifs du projet

L'objectif principal du projet est de réduire l'effort du planificateur et lui faire gagner du temps tout en minimisant le risque d'erreur humaine. Pour cela, une application permettrait de centraliser les données des agents et ainsi simplifier le travail du planificateur. D'autre part, l'application permettrait aux agents de gérer leurs disponibilités de manière autonome et de les transmettre au planificateur.

2. Membres de l'équipe

2.1. Léo Zmoos

- **Mail:** leo.zmoos@heig-vd.ch
- **Rôle :** Responsable du backend, de la sécurité et de l'architecture de la base de données.
- **Compétences principales:**
 - Développement backend en TypeScript avec Next.js App Router
 - Sécurité des API (authentification avec JWT, gestion des rôles)
 - Modélisation de base de données relationnelle avec Prisma + PostgreSQL
 - Intégration d'API (Leaflet / OpenStreetMap)
 - Mise en place de la logique de validation/refus des missions par les agents
 - Tests et debug serveur

2.2. Dylan Oliveira Ramos

- **Mail:** dylan.oliveiramos@heig-vd.ch
- **Rôle :** Responsable de l'interface utilisateur (frontend) et de l'expérience utilisateur (UX).
- **Compétences principales :**
 - Développement frontend en React avec MUI (Material UI)
 - Création d'interfaces responsives et accessibles
 - Intégration des composants interactifs (boutons, dialogues, cartes)
 - Gestion du routing côté client (navigation, rafraîchissement via router)
 - Déploiement et mise en page UI/UX intuitive

3. Présentation de l'idée

Notre idée est de créer une application web qui permettrait aux agents de sécurité de gérer leurs disponibilités et de les transmettre au planificateur. Le planificateur pourrait ainsi visualiser les disponibilités des agents et attribuer les plages horaires en toute simplicité. L'application serait accessible via un navigateur web et serait responsive pour s'adapter à tous les appareils (mobile, tablette, desktop). L'application serait divisée en deux parties : une partie pour le planificateur et une partie pour les agents. Le planificateur aurait accès à un calendrier avec les disponibilités des agents, tandis que les agents pourraient gérer leurs disponibilités et consulter leur planning.

4. Secteur de marché

Le secteur de marché de cette application se situerait principalement dans **la gestion des ressources humaines et la planification du personnel**, avec un focus particulier sur les **services de sécurité privée**. Voici les principaux segments concernés :

1. **Sécurité privée et surveillance** : entreprises de gardiennage, sociétés de sécurité privée, services de protection rapprochée.
2. **Gestion des effectifs et RH** : solutions de gestion des plannings pour entreprises ayant des horaires flexibles ou des équipes en rotation.
3. **Entreprises de facility management** : gestion des agents de sécurité au sein des grandes infrastructures (hôpitaux, aéroports, centres commerciaux).
4. **Événementiel** : entreprises gérant la sécurité pour des événements temporaires.
5. **Secteur public** : municipalités et administrations ayant des services de sécurité internes.

5. Public cible

Le public cible de cette application est constitué principalement de deux groupes : les planificateurs et les agents de sécurité.

1. **Planificateurs** : responsables de la gestion des plannings et de l'attribution des plages horaires aux agents. Ils sont souvent confrontés à des défis liés à la complexité de la planification, à la gestion des disponibilités et à la communication avec les agents.
2. **Agents de sécurité** : professionnels chargés de la surveillance et de la sécurité sur le terrain. Ils doivent gérer leurs disponibilités et communiquer avec le planificateur pour s'assurer qu'ils sont affectés aux bons événements.

5.1. Personae

5.1.1. Jean Dupont — Le Planificateur

Âge: 35 ans

Poste : Responsable de la planification des agents

Niveau numérique: Moyen (à l'aise avec Excel, mais pas expert logiciel)

Contexte professionnel: Jean travaille dans une société privée de sécurité, où il gère chaque semaine les plannings de plus de 40 agents pour une multitude d'événements (festivals, foires, événements sportifs). Il utilise actuellement un tableur Excel combiné à des appels téléphoniques et des SMS/Whatsapp pour coordonner les affectations.

Frustrations actuelles:

- Trop de temps perdu à reporter manuellement les données
- Informations contradictoires ou manquantes à cause de canaux multiples (mail, WhatsApp, appels)
- Difficile de suivre les confirmations/réponses des agents
- Gestion de dernière minute stressante

Besoins fonctionnels:

- Interface centralisée pour créer un événement, visualiser les missions, et suivre les affectations
- Visualisation rapide des agents confirmés / manquants
- Pouvoir assigner automatiquement des agents selon leur disponibilité ou accréditation
- Recevoir des notifications quand un agent refuse une mission ou ne répond pas

Objectifs personnels:

- Réduire au maximum le temps de planification hebdomadaire
- Améliorer la fiabilité des plannings
- Avoir une meilleure traçabilité pour justifier les affectations ou les absences

Scénario d'usage type: Jean crée un nouvel événement dans l'interface, sélectionne les horaires et le nombre d'agents requis. Le système envoie automatiquement des invitations aux agents disponibles. Il suit en temps réel qui accepte ou refuse, et peut réassigner si besoin en quelques clics.

5.1.2. Pierre Martin — L'Agent de Sécurité

Âge : 28 ans

Poste : Agent de sécurité événementiel

Niveau numérique : Bon (utilisation régulière du smartphone, familiarisé avec les apps professionnelles)

Contexte professionnel: Pierre travaille en horaires décalés, parfois pour plusieurs entreprises. Il reçoit ses missions par SMS ou appel, parfois trop tard ou de manière désorganisée. Il note tout dans un agenda papier ou une app perso.

Frustrations actuelles:

- Ne sait pas toujours s'il est affecté ou non (principe que l'agent est toujours disponible si il ne spécifie pas le contraire)
- Reçoit des informations parfois en doublon ou contradictoires
- Ne peut pas facilement refuser ou justifier une indisponibilité
- Ne reçoit pas les instructions de mission à temps principalement en cas de remplacement

Besoins fonctionnels:

- Voir toutes les missions proposées en un seul endroit
- Accepter/refuser une mission directement depuis l'application
- Ajouter une raison en cas de refus
- Voir les instructions d'un événement et l'adresse exacte sur une carte

Objectifs personnels:

- Gérer son emploi du temps de manière proactive
- Avoir plus de clarté sur ses missions
- Réduire les oublis ou les erreurs
- Éviter les appels inutiles avec les planificateurs

Scénario d'usage type: Pierre reçoit une notification pour une nouvelle mission. Il clique sur l'événement, lit les détails, accepte la mission en un clic. S'il refuse, il indique la raison. Son planning est mis à jour

automatiquement.

6. Questionnaires et hypothèses à valider avec le public cible

Nous avons réfléchi à plusieurs questions en essayant d'être le plus exhaustif possible. Lors de nos interviews, nous avons commencé par poser des questions ouvertes pour comprendre le fonctionnement actuel de la planification et les besoins des agents. Ensuite, nous avons posé des questions plus spécifiques sur les fonctionnalités souhaitées dans l'application. Le tout en évitant de poser des questions orientées pour ne pas influencer les réponses.

6.1. Objectifs des entretiens

Objectif principal	Détail
Identifier les limitations des systèmes de planification actuels	Excel, appels, messagerie, etc.
Comprendre les besoins spécifiques de chaque rôle	Informations utiles, actions critiques, communication
Recueillir les attentes fonctionnelles pour un outil numérique	Visualisation du planning, interaction, notifications
Valider l'intérêt et la disposition à utiliser un outil numérique	Acceptabilité, valeur perçue, préférences UI/UX
Vérifier les freins potentiels à l'adoption	Complexité, login, confidentialité

6.2. Questions pour les planificateurs

- Quelle est votre méthode de travail actuelle? (workflow)
- Qu'est-ce qui fait que la planification est difficile?
- Qu'est-ce qu'il se passe si un agent annonce qu'il ne peut pas venir?
- Quelle est la fréquence (semaine, mois) des planifications?
- Quelles sont les informations nécessaires concernant les agents?
- Y-a-t'il du matériel à fournir aux agents?
- Y-a-t'il différents types d'agents?
- Si vous aviez une application pour la planification, à quoi celle-ci ressemblerait?

6.3. Questions pour les agents

- Expliquez-nous votre travail d'agent, qu'est-ce que vous faites?
- Comment transmettez-vous vos disponibilités au planificateur?
- Que devez-vous faire si vous ne pouvez pas vous présenter à l'événement?

- Trouvez-vous l'organisation au sein de l'entreprise efficace?
- Cela vous dérange-t'il de devoir créer un compte?
- Si vous aviez une application pour la planification, à quoi celle-ci ressemblerait?

6.4. Hypothèses à valider

ID	Hypothèse	Cible	Statut (après entretiens)
H1	Les planificateurs perdent un temps important à gérer les plannings manuellement	Planificateur	Confirmé
H2	Les agents souhaitent un accès mobile à leur planning personnel	Agent	Confirmé
H3	La majorité des refus sont communiqués par téléphone, ce qui complexifie le suivi	Planificateur	Confirmé
H4	Les agents accepteraient de créer un compte si cela facilite la gestion de leurs missions	Agent	Confirmé
H5	Une fonctionnalité de carte pour localiser l'événement est perçue comme utile	Agent	Confirmé
H6	Le planificateur souhaite être notifié en cas de refus ou d'absence d'agent	Planificateur	Confirmé

7. Analyse des retours du public cible

7.1. Retours des planificateurs

De manière générale, les planificateurs ont exprimé le besoin d'une solution centralisée pour gérer les disponibilités des agents. Ils ont souligné la complexité de la planification actuelle, notamment en raison de la communication par message. Ils souhaitent une application qui leur permette de visualiser facilement les disponibilités des agents et d'attribuer les plages horaires sans effort.

Je cite: "On passe notre temps à courir après les disponibilités des agents par messages. Un outil unique pour tout centraliser nous ferait gagner un temps fou." — Responsable planning, entreprise de sécurité à Marly

Nous avons également interviewé des planificateurs ayant déjà une application dédiée à la gestion des plannings. Dans l'ensemble, ils trouvent que leur application fonctionne très bien, mais qu'elle pourrait être améliorée en termes d'ergonomie et de fonctionnalités.

Je cite: "L'outil qu'on utilise est plutôt complet, mais l'interface est vieillissante et pas adaptée à tous les types de missions. On aimerait pouvoir mieux gérer les exceptions et les urgences."

"Je trouve que certaines applis oublient qu'on travaille souvent sur le terrain avec un téléphone. Si c'est trop lent ou confus, je reviens vite au papier."

7.2. Retours des agents

La majorité des agents ont dit être satisfaits de la manière dont ils transmettent leurs disponibilités au planificateur. En effet, ceux-ci n'ont pas de difficulté à communiquer avec le planificateur par message ou via l'application déjà existante. Nous en avons conclu que le problème ne se situe pas au niveau de la communication entre le planificateur et les agents, mais plutôt au niveau de la gestion des disponibilités par le planificateur.

Certains agents nous ont fait part de leur ressenti concernant les planificateurs. Selon eux, le problème n'est pas la gestion des disponibilités des agents, mais plutôt le nombre d'événements à planifier. Cela nous a amené à réfléchir à la possibilité d'intégrer une fonctionnalité de gestion des événements dans l'application.

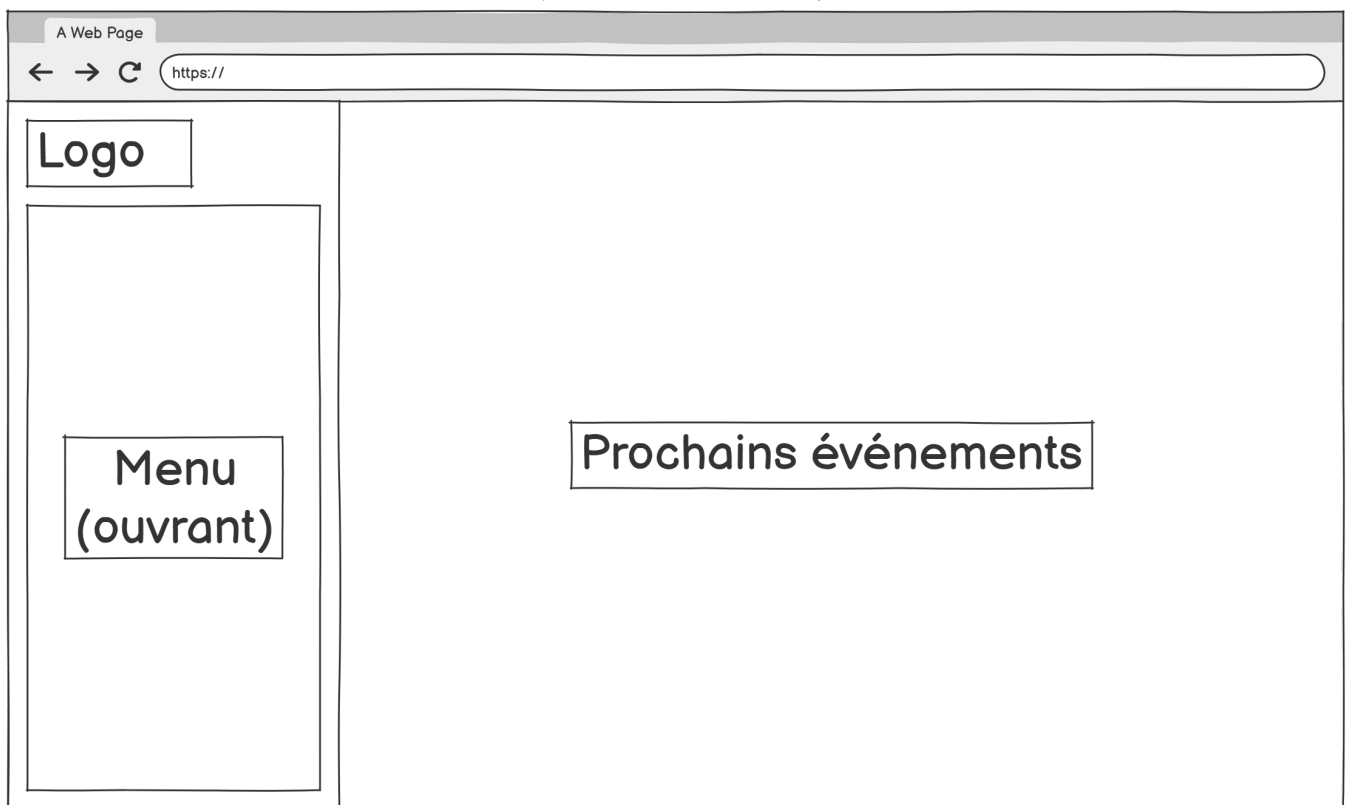
Je cite: "C'est pas tant les agents le problème, c'est qu'il y a trop d'événements, et ils doivent tous être calés en même temps." — Leonard K., agent de sécurité événementiel

8. Maquettes

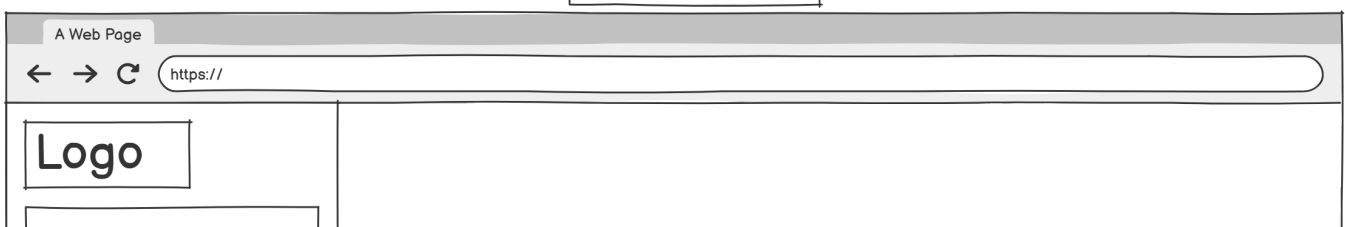
8.1. Wireframes

Nous avons réalisé quelques wireframes pour visualiser l'interface de l'application. Nous n'avons pas fait de wireframes pour chaque page car la plupart des pages sont similaires.

Home page

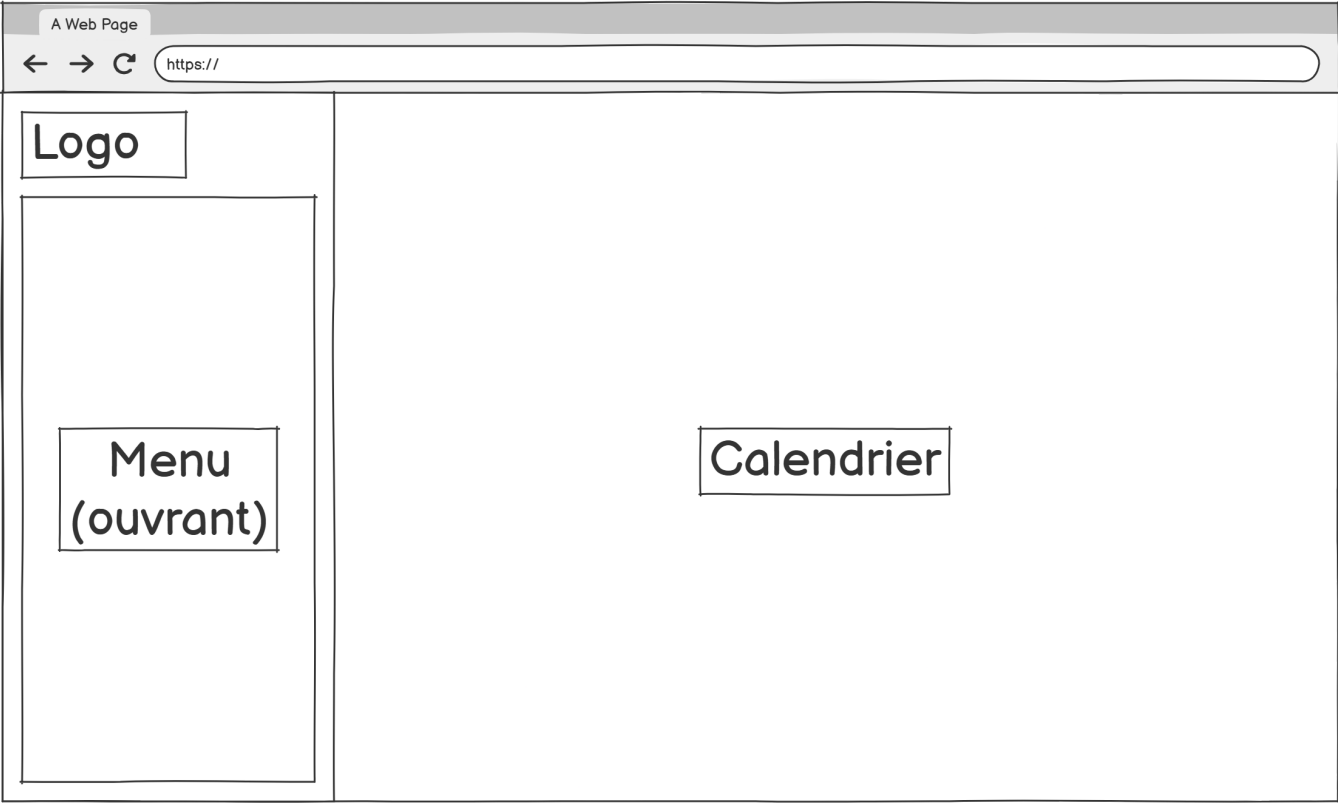


Page profil



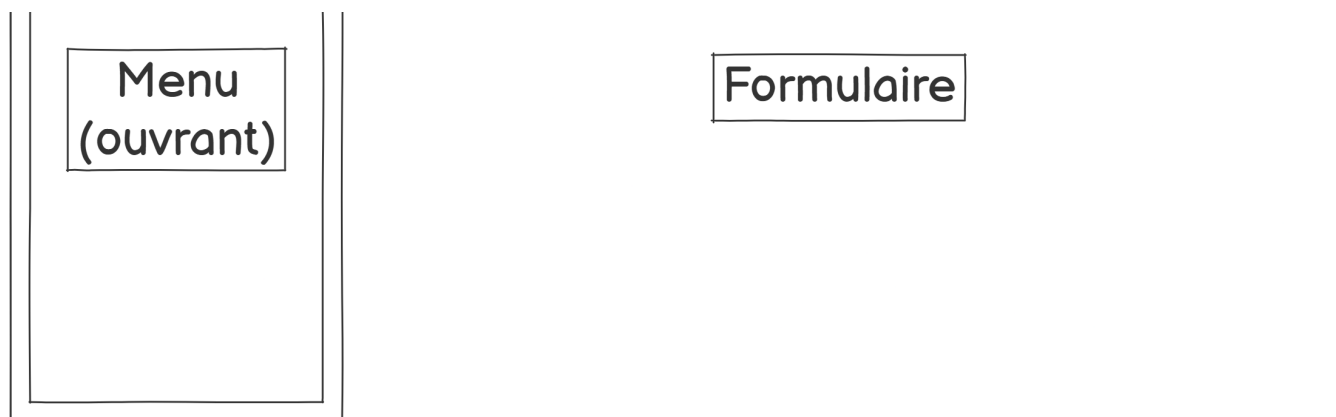


Page calendrier



Page disponibilités (agent)





8.1.1. Page home

Pour les planificateurs :

- La home page contient la liste des événements à venir.

Pour les agents :

- La home page contient la liste des événements à venir où l'agent doit intervenir.

8.1.2. Page profil

Pour les planificateurs :

- La page profil contient les informations de la personne.

Pour les agents :

- La page profil contient les informations de la personne et la possibilité d'accéder à la page des disponibilités.

8.1.3. Page calendrier

Pour les planificateurs :

- La page calendrier contient tous les événements planifiés sur plusieurs mois.

Pour les agents :

- La page calendrier contient tous les événements auxquels l'agent intervient.

8.1.4. Page disponibilités

Pour les agents :

- La page disponibilités permet aux agents de gérer leur disponibilités.

8.2. Mockups

Par manque de temps, nous n'avons pas pu réaliser de mockups pour l'application. Cependant, nous avons une idée précise de ce à quoi l'application devrait ressembler. Nous avons pris en compte les retours des

planificateurs et des agents pour concevoir une interface simple et intuitive.

D'autre part, nous avons prévu de réaliser l'application en utilisant des composants déjà existants, ce qui réduit le risque d'avoir une interface peu ergonomique et nous autorise à ne pas faire de mockups.

9. Choix technologiques

Nous avons choisi de développer l'application en utilisant le framework fullstack **Next.js**. Ce framework nous permet de créer une application web complète avec un backend et un frontend en utilisant le même langage (JavaScript). De plus, Next.js offre des fonctionnalités intéressantes telles que le rendu côté serveur, la génération de pages statiques et la gestion des routes. D'autre part, nous sommes familiers avec le langage JavaScript et le framework React, ce qui nous a évidemment orienté vers Next.js.

Nous avons également choisi d'utiliser **Prisma** comme ORM pour interagir avec la base de données. Prisma est un ORM moderne et performant qui nous permet de gérer facilement les migrations de la base de données et d'effectuer des requêtes complexes.

Nous avons opté pour une base de données **PostgreSQL** car elle est robuste et bien supportée par Prisma. De plus, PostgreSQL est une base de données relationnelle qui convient parfaitement à notre besoin de gérer des relations entre les agents, les événements et les disponibilités.

Concernant l'authentification, nous avons initialement prévu d'utiliser **NextAuth** pour gérer l'authentification des utilisateurs. Cependant, nous avons finalement décidé de ne pas implémenter cette fonctionnalité dans le MVP, car cela aurait nécessité un temps de développement supplémentaire. Nous avons donc opté pour une solution simple où les utilisateurs peuvent se connecter avec un nom d'utilisateur et un mot de passe ce qui, à l'aide de la librairie **Jose**, génère un token JWT stocké dans un cookie. Nous utilisons également **bcrypt** pour le hachage des mots de passe.

Pour le frontend, nous utilisons des composants React préfabriqués par la librairie **Material UI**. Cette librairie nous permet de créer une interface utilisateur moderne et responsive sans avoir à développer tous les composants nous-mêmes. De plus, Material UI est bien documentée et facile à utiliser.

9.1. Résumé

Pour résumer, voici les choix technologiques que nous avons faits :

- **Framework** : Next.js
- **ORM** : Prisma
- **Base de données** : PostgreSQL
- **Authentification** : JWT avec bcrypt
- **Librairie UI** : Material UI

10. Tests utilisateurs

Lors de la phase de tests utilisateurs, notre application n'était pas tout à fait terminée. Cependant, nous avons pu réaliser quelques tests avec des utilisateurs potentiels pour recueillir leurs retours sur l'ergonomie et les fonctionnalités de l'application. Les retours ont été globalement positifs, notamment concernant l'interface, mais nous nous sommes rendus compte d'un changement majeur qui pourrait être apporté à l'application.

10.1. Changement majeur

Au lieu de demander aux agents de transmettre leurs disponibilités, pourquoi ne pas leur demander de s'inscrire directement aux événements?

C'est la direction que nous avons décidé de prendre. En effet, cela évite au planificateur de devoir gérer les disponibilités des agents et permet aux agents de s'organiser sans avoir à attendre l'approbation du planificateur. Cette manière de faire simplifie grandement la gestion des plannings et permet au planificateur de gagner un temps considérable.

11. Pérennité du projet

Pour assurer la pérennité du projet, nous avons pris en compte plusieurs aspects :

- **Documentation** : Nous avons documenté le code et les choix techniques pour faciliter la prise en main du projet par d'autres développeurs. La documentation est disponible dans le dossier **docs** du dépôt GitHub.
- **Evolutivité** : Nous avons conçu l'application de manière modulaire pour faciliter l'ajout de nouvelles fonctionnalités à l'avenir. Par exemple, nous avons prévu une architecture qui permettrait d'intégrer facilement un système d'authentification plus complexe si nécessaire.

Cependant, il reste encore des points à améliorer pour assurer la pérennité du projet :

- **Tests automatisés** : Nous n'avons pas encore mis en place de tests automatisés pour l'application. Cela reste un point à améliorer pour garantir la stabilité du code et faciliter les évolutions futures.
- **Pipelines CI/CD** : Nous n'avons pas encore mis en place de pipelines d'intégration et de déploiement continu pour automatiser les tests et le déploiement de l'application. Cela reste un point à améliorer pour faciliter la gestion du projet à long terme.
- **Aspects financiers** : Nous n'avons pas encore réfléchi à un modèle économique pour monétiser l'application. Une solution pourrait être de proposer un abonnement mensuel/annuel pour les entreprises souhaitant utiliser l'application. Mais cela reste un point à approfondir pour assurer la viabilité financière du projet.

12. Difficultés rencontrées et solutions

12.1. Prise en main de Next.js

Nous voulions utiliser un framework fullstack populaire et moderne pour le développement de l'application. Next.js s'est imposé comme un choix évident, mais la prise en main a été un peu difficile au début. En effet, contrairement à d'autres frameworks, Next.js laisse une grande liberté de choix dans les librairies à utiliser notamment pour l'ORM, l'authentification, etc. Cela a rendu la phase de démarrage un peu plus longue que prévu.

Heureusement, Next.js étant très populaire, nous avons pu trouver de nombreux tutoriels et ressources en ligne pour nous aider à nous familiariser avec le framework. De plus, la documentation officielle est très complète et bien structurée, ce qui nous a permis de progresser rapidement.

12.2. Tentative d'implémentation de l'authentification

Nous avons initialement prévu d'utiliser NextAuth pour gérer l'authentification des utilisateurs. Cependant, nous avons rencontré quelques difficultés lors de la configuration de NextAuth avec Prisma et PostgreSQL. De plus, nous avons réalisé que l'implémentation de l'authentification aurait nécessité un temps de développement supplémentaire que nous n'avons pas.

Nous avons donc décidé de ne pas implémenter cette fonctionnalité dans le MVP et de simplement utiliser un système d'authentification basique avec des tokens JWT stockés dans des cookies.

13. Conclusion

Ce projet nous a permis de découvrir les étapes de lancement d'un MVP, de l'identification d'un problème concret au développement et aux tests utilisateurs. Nous avons appris l'importance de valider les hypothèses auprès des parties prenantes et de prioriser les fonctionnalités essentielles pour répondre aux besoins réels.

Grâce à ce processus, nous avons pu concevoir une solution qui n'est certes, pas terminée mais qui pourrait simplifier significativement la gestion des plannings pour les entreprises de sécurité, tout en offrant une expérience intuitive aux agents et aux planificateurs. Les retours des utilisateurs ont confirmé la pertinence de notre approche, notamment avec le pivot vers un système d'inscription directe aux événements, qui réduit la charge administrative.

L'aventure nous a également renforcés dans nos compétences techniques et notre capacité à travailler en équipe sur un projet concret. Enfin, elle a souligné l'importance de rester agile et à l'écoute des utilisateurs pour bâtir une solution pérenne et évolutive.