

Rapport de stage BTS SIO

2023

**03/01/2023 au** **10/02/2023**

**Raja France**

**Ecrit par Maxim Dailly Zagrebelnyy**



Une image contenant texte, clipart

Description générée automatiquement

Rapport de stage BTS SIO option SLAM seconde année

Stagiaire en tant que développeur web

Effectué par Maxim Dailly Zagrebelnyy

Du 03/01/2023 au 10/02/2023

**Tutrice de stage** : Beatrice Bernede

**Maître de stage :** Rayan Benhamida, Nicolas Techer

**Etablissement**: Lycée Jean Rostand – BTS SIO option SLAM (Solution Logicielle Application Métier)

**Entreprise d’accueil** : Raja France à Tremblay en France

**Remerciements**

En premier lieu, je voudrais remercier Rayan Benhamida, Nicolas Techer ainsi que Vincent Grondin. Je remercie Rayan pour son aide à la production du projet et de sa bienveillance, je remercie Vincent pour son expertise en développement et Nicolas pour les différentes discussions sur mes études futures ainsi que sur mon avenir professionnel.

Je remercie Grégory Loth pour les bons moments passés ensemble à travailler sur ce projet, les heures de discussions qui sont nécessaires au bon fonctionnement d’une équipe qui a pu créer un projet enrichissant humainement et techniquement parlant.

Je remercie les différents collègues de l’équipe de la Digital Factory pour m’avoir encadré ; j’ai pu partager de très bons moments avec eux.

Je remercie l’ensemble des professeurs qui m’ont permis d’en être à ce point dans mes études. Je remercie encore une fois Mme Béatrice Bernède qui m’a appris presque l’intégralité de ce que je sais en développement en informatique à l’aide de son enseignement et de sa sympathie.

Je remercie mes parents pour m’avoir accompagné lors de ce stage grâce à leur soutien inconditionnel, à la fois moral et économique ainsi que de leurs aides pour le futur de mes études.

Sommaire

[Introduction 5](#_Toc128239448)

[Informations sur l’entreprise 5](#_Toc128239449)

[I. Présentation de la mission 6](#_Toc128239450)

[A. La compréhension de l’objectif 6](#_Toc128239451)

[B. Les outils utilisés 6](#_Toc128239452)

[1) La partie front-end 6](#_Toc128239453)

[2) La partie back-end 7](#_Toc128239454)

[C. La répartition du travail 7](#_Toc128239455)

[II. Le développement du projet 8](#_Toc128239456)

[A. Mise en place du projet 8](#_Toc128239457)

[B. L’API (Application Programming Interface) 8](#_Toc128239458)

[C. Mon travail sur la partie front-end 9](#_Toc128239459)

[D. Les fonctionnalités apportées durant le développement 10](#_Toc128239460)

[1. L’ajout des mots de passe 11](#_Toc128239461)

[2. La page « Mes réservations » 11](#_Toc128239462)

[3. Les statistiques 11](#_Toc128239463)

[4. Automatisation des mails 11](#_Toc128239464)

[5. Les QR-codes 11](#_Toc128239465)

[III. La finalité du projet 12](#_Toc128239466)

[Conclusion 13](#_Toc128239467)

[Annexes 14](#_Toc128239468)

# Introduction

## Informations sur l’entreprise

Le groupe Raja est le plus grand groupe européen sur les marchés de la vente à distance de distribution d'emballages, de fournitures de bureau et d'équipements industriels pour les entreprises. En 2020, le groupe comprend 26 sociétés dans 19 pays européens, emploie 4 500 personnes et dispose de 14 centres logistiques de distribution. La même année, la moitié des ventes du groupe sont réalisées en ligne.

Premier sur ses marchés en Europe, le groupe a réalisé 1,2 milliard d'euros de chiffre d'affaires en 2021. Il est dirigé par Danièle Kapel-Marcovici.

En 2006, Danièle Kapel-Marcovici crée la *« Fondation Raja-Danièle Marcovici pour les femmes »* sous l'égide de la Fondation de France. En 2018, la fondation est dotée d'un budget d'environ 1 million d'euros. Raja est également le partenaire principal du Festival International de la Bande Dessinée d'Angoulême depuis 2018.

La Digital Factory est un groupe qui se sépare de la DSI (Direction du Système d’Information). La Digital Factory (plus communément DF) est un service comportant un ensemble de compétences et de métiers ciblés sur le e-commerce, les nouvelles fonctionnalités web et la relation client avec les problèmes techniques web. La DF contrôle le site e-commerce Raja, les sites dits « corporate » (sites que chaque filiale de Raja peut modifier à son souhait) et la digitalisation de l’entreprise.

*Cf annexe 1 pour la charte organisationnelle*

La Digital Factory a choisi d’être polyvalente et utilise tous types de logiciels ainsi que tous types de systèmes d’exploitation (Windows, MacOs et Linux). Dans ma tâche, j’ai utilisé Visual studio code en tant qu’éditeur de code, MongoDB en tant que SGBD. Les différents logiciels de bureautique sont de la suite office (Word, Outlook etc). Afin de mettre en pace le projet nous avons utilisé Team Foundation Server, aujourd'hui appelé Azure DevOps Server, qui est une forge logicielle éditée par Microsoft permettant la gestion des sources, la gestion des builds, le suivi des éléments de travail, la planification, la gestion de projet et l'analyse des performances.

Une image contenant texte

Description générée automatiquement

Figure : Azure Devops

# I. Présentation de la mission

### La compréhension de l’objectif

Ma principale mission lors de ce stage était de créer en collaboration avec Grégory Loth (stagiaire) une application web permettant de réserver des bureaux dans la Digital Factory. En effet, à la suite de l’augmentation des collaborateurs dans ce service, une organisation des places était donc nécessaire. Entre les internes, les travailleurs indépendants, les personnes venant au bureau et ceux en télétravail, l’open space est souvent chargé et les places sont souvent prises. Certaines places sont dites « fixe » et d’autres « flex ». Les places fixes sont des places où le collaborateur ne changera jamais de place et où le bureau ne peut être pris par quelqu’un d’autre. L’application aura un espace d’administration afin de changer les emplacements des bureaux, modifier les réservations, avoir des statistiques…

De ce fait, nos tuteurs Rayan Benhamida, Nicolas Techer et Vincent Grondin nous ont donné comme mission de trouver une solution face à ce problème à l’aide d’une API. Une API (application programming interface ou « interface de programmation d'application ») est une interface logicielle qui permet de « connecter » un logiciel ou un service à un autre logiciel ou service afin d'échanger des données et des fonctionnalités.

Au tout début de notre stage nous avions une liste de base à faire, mais plus notre stage avançait, plus des fonctionnalités ont été rajoutées pour avoir plus de possibilités et de fonctionnalités pour les utilisateurs.

Au début, nous avions pour but de :

* Pouvoir se connecter avec son propre compte
* Prendre une réservation
* Supprimer une réservation

Une page d’administration était aussi nécessaire afin de réorganiser les emplacements des bureaux, supprimer des réservations, réserver plusieurs places, changer un bureau de type (flex, fixe).

### Les outils utilisés

Discutant avec nos tuteurs, nous avions eu le choix de choisir les langages. Mais pour rester dans les éléments utilisés dans l’entreprise nous avons suivi leurs bases en programmation.

Les projets de ce type sont découpés en deux parties : une partie « front » et une partie « back ».

#### La partie front-end

La partie front-end ou le développement web frontal correspond aux productions HTML, CSS et JavaScript d’une page internet ou d’une application qu’un utilisateur peut voir et avec lesquelles il peut interagir directement.

Pour cette partie nous avons utilisé :

* HTML, CSS et Bootstrap
* Javascript afin d’interagir avec la page

JavaScript est un langage de programmation de scripts principalement employé dans les pages web interactives et à ce titre est une partie essentielle des applications web. Avec les langages HTML et CSS, JavaScript est au cœur des langages utilisés par les développeurs. Une grande majorité des sites web l'utilisent, et la majorité des navigateurs web disposent d'un moteur JavaScript pour l'interpréter. JavaScript est aussi employé pour les serveurs Web avec l'utilisation par exemple de Node.js.

#### La partie back-end

La partie back-end est un terme désignant un étage de sortie d'un logiciel devant produire un résultat. On l'oppose au front-end qui lui est la partie visible de l'iceberg.

Et pour celle-ci nous avons utilisé :

* NodeJS permettant d’accéder aux données
* MongoDB en tant que base de données NoSQL

MongoDB est un système de gestion de base de données orienté documents, répartissable sur un nombre quelconque d'ordinateurs et ne nécessitant pas de schéma prédéfini des données. Il est écrit en C++. De plus MongoDB fonctionne dans le cloud, les données transmises entre moi et Grégory seront donc identiques.

En informatique et en bases de données, NoSQL désigne une famille de systèmes de gestion de base de données (SGBD) qui s'écarte du paradigme classique des bases relationnelles. Dans notre cas les données seront stockées sous format JSON.

Durant notre formation au BTS nous n’avions pas tellement travaillé avec du JavaScript et ses extensions. Le NoSQL était tout aussi une nouvelle découverte.

### La répartition du travail

Grégory et moi avons des affinités et des préférences diverses qui ont permis de se diviser les tâches. Préférant l’algorithmie, j’ai proposé de faire la partie back-end ainsi que la mise en place des données. Grégory pour sa part, s’est occupé de la partie front-end pour mettre le plus en avant les différentes pages.

# II. Le développement du projet

Ayant un projet relativement vaste, je ne serai pas en mesure de détailler chaque fichier et chaque fonctionnalité.

### Mise en place du projet

Dès le début du projet, nous avons utilisé Figma pour faire le design de la page. Figma est un éditeur de graphiques vectoriels et un outil de prototypage. Il est utilisé pour faire des designs de pages et est aussi de travailler en groupe en direct. *Vous pouvez voir dans l’annexe 2 le premier design à l’aide Figma.*

Pour mettre en place les collections (groupement de données pour la base de données), il faut créer des schémas permettant de donner les indications nécessaires de la structure des informations.



Figure - Schéma de la collection des utilisateurs

Dans cet exemple nous pouvons voir le schéma de la collection « User ». Cela permet de donner les caractéristiques d’un utilisateur comme on pourrait faire avec une table SQL.

### L’API (Application Programming Interface)

Dans le backend, l’API possède un fichier de requête web et un fichier interagissant avec la base de données.

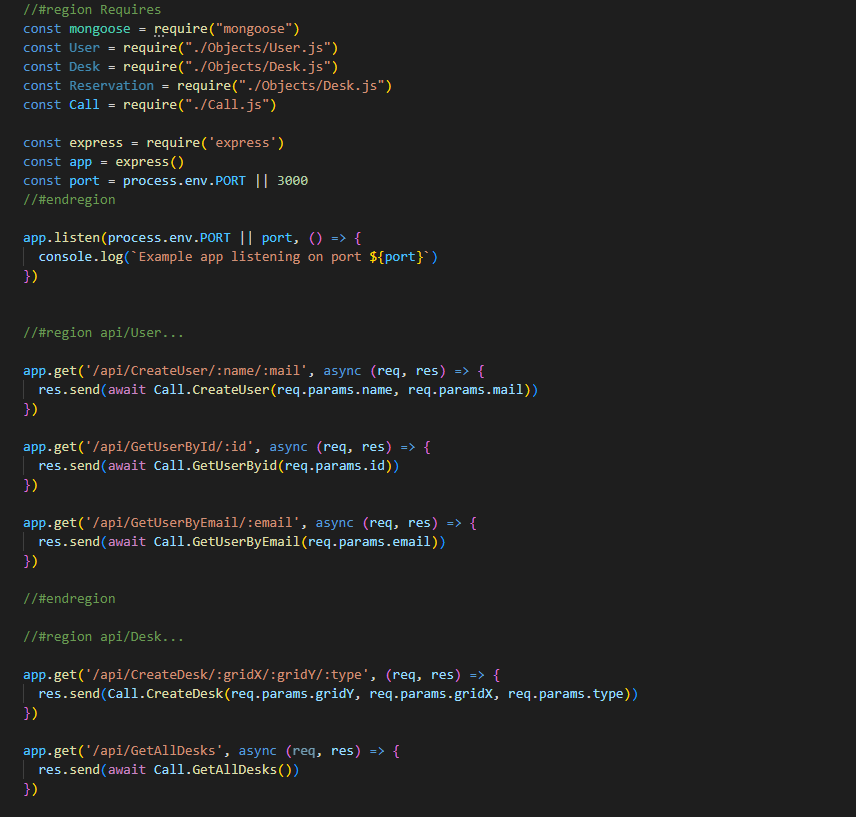


Figure 4 - Un exemple de fonction du fichier interagissant avec la base de données

Figure 3 - Le fichier de requête API

Nous pouvons donc voir sur le fichier appelant l’API les différente routes (par exemple <http://localhost/api/CreateUser/:name/:mail>) ayant des interactions avec la base. Dans l’exemple précèdent, si nous renseignons cet URL dans notre navigateur avec de bons paramètres (nom et mail), un nouvel utilisateur sera donc créé.

En effet, en faisant cette requête nous allons appeler la fonction nécessaire à la création de l’utilisateur.

Dans cette requête interagissant avec la base de données, nous pouvons observer une requête asynchrone (return Promise dans la fonction). La programmation asynchrone est une technique qui permet à un programme de démarrer une tâche à l'exécution potentiellement longue et, au lieu d'avoir à attendre la fin de la tâche, de pouvoir continuer à réagir aux autres évènements pendant l'exécution de cette tâche. Une fois la tâche terminée, le programme en reçoit le résultat. L'objet Promise (pour « promesse ») est utilisé pour réaliser des traitements de façon asynchrone. Une promesse représente une valeur qui peut être disponible maintenant, dans le futur voire jamais. Grace à cela nous pouvons demander des données tout en continuant différents processus en même temps, cela gagne du temps lorsque nous avons une grande quantité de données à recevoir.

Dans la fonction nous recherchons donc toutes les réservations suivant une date donnée. La fonction resolve permettra de recevoir les données recherchées sous format JSON et la fonction reject permettra de rendre une erreur en cas de problème.

### Mon travail sur la partie front-end

Pour pouvoir visualiser, utiliser et interpréter les données, il faut aussi recevoir les informations du côté front-end. Comme nous pouvons le voir sur la figure 5, la page d’accueil a besoin de recevoir les informations concernant la réservation (ou non) du bureau. A l’aide de la fonction *fetch*, nous pouvons appeler l’API pour qu’il nous envoie ses données. Dans l’exemple ci-dessous, nous appelons l’API pour connaître toutes les réservations à l’aide d’une date. Nous recevons ces informations et les interprétons pour changer le visuel en fonction de si le bureau est réservé. Le visuel change si la réservation nous appartient (couleur bleue) ou si elle appartient à quelqu’un d’autre (couleur rouge).

Une image contenant texte

Description générée automatiquement

Figure 5 - Fonction front-end pour changer la couleur du bureau

Sur la partie front-end, je me suis également occupé de créer les fichiers Javascript d’administration : modification, ajout et suppression des bureaux et des utilisateurs.

### Les fonctionnalités apportées durant le développement

Pour donner suite à notre avancée rapide du projet, nos tuteurs ont décidé de rajouter certaines fonctionnalités. Nous avons également fait part à nos tuteurs de certaines modifications qui seraient utiles, ce qui nous a permis d’ajouter encore plus de fonctionnalités.

#### L’ajout des mots de passe

Au départ, nous ne voulions pas mettre de mots de passe pour éviter une contrainte inutile pour les utilisateurs. Après réflexion, nous avons vu que certaines fonctionnalités de l’administration avaient « de trop gros pouvoirs », et qu’il était plus prudent de rajouter des éléments de protection. Les utilisateurs ont également la possibilité de changer leur mots de passe via la page « Mon profil ».

#### La page « Mes réservations »

Cette page permet de visualiser toutes les réservations que nous avons pris afin de faciliter le planning des utilisateurs. Les réservations sont visibles à l’aide d’un agenda.

#### Les statistiques

Afin d’avoir un meilleur visuel des réservations au sein de l’open-space, il était préférable pour les administrateurs d’avoir des statistiques concernant : les jours où les bureaux sont le plus réservés, le taux de remplissage des réservations par semaine. Cela permet de mieux gérer le flux de personnes dans la DF.

#### Automatisation des mails

A l’aide du module NodeMailer, NodeJS peut automatiser l’envoi de courriels aux utilisateurs lorsqu’ils réservent un bureau, lorsqu’ils suppriment une réservation ou lorsqu’ils changent de mots de passe. Un mail de bienvenue est également envoyé lorsque les utilisateurs créent leur compte. Les utilisateurs ont bien entendu la possibilité de donner leur accord ou non pour la notification par mail. Ce choix s’opère sur la page « Mon profil ». N’ayant pas eu le temps d’optimiser cette partie du code, un mail est envoyé à chaque réservation, ce qui peut être assez contraignant.

#### Les QR-codes

Pour faciliter la réservation dans l’open-space, des QR-codes seront placés sur chaque bureau. En scannant le QR-code, les utilisateurs visualisent si la place est réservée, et par qui elle l’est, et ont la possibilité de la réserver si elle n’est pas occupée.

# III. La finalité du projet

Pour mettre en avant le projet terminé voici les différentes pages de celles-ci :

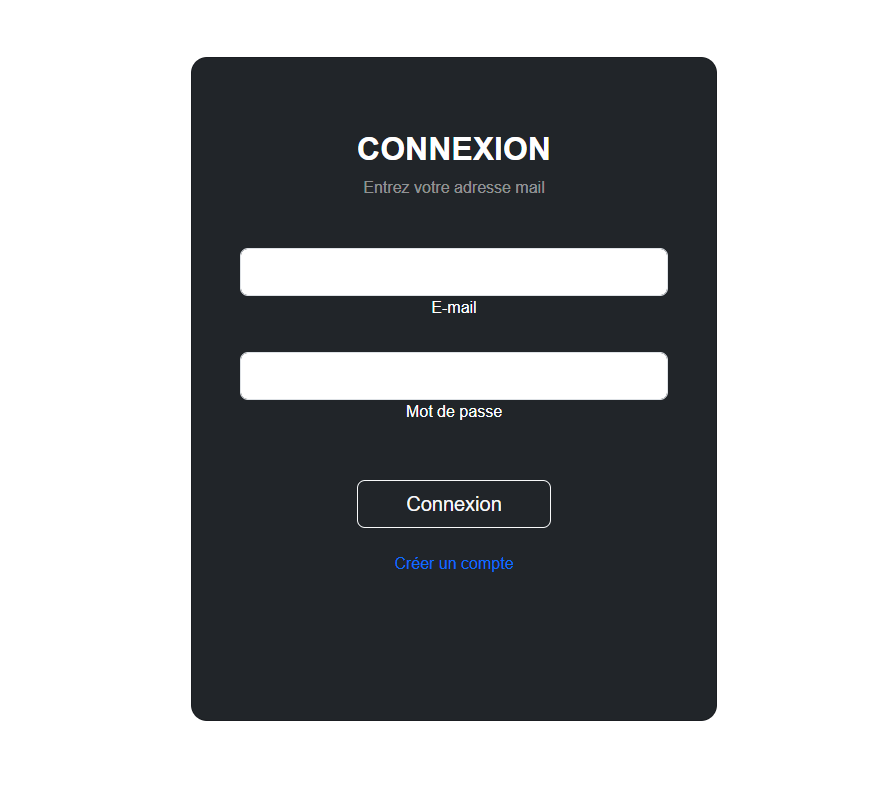


Figure 6 - Page de connexion

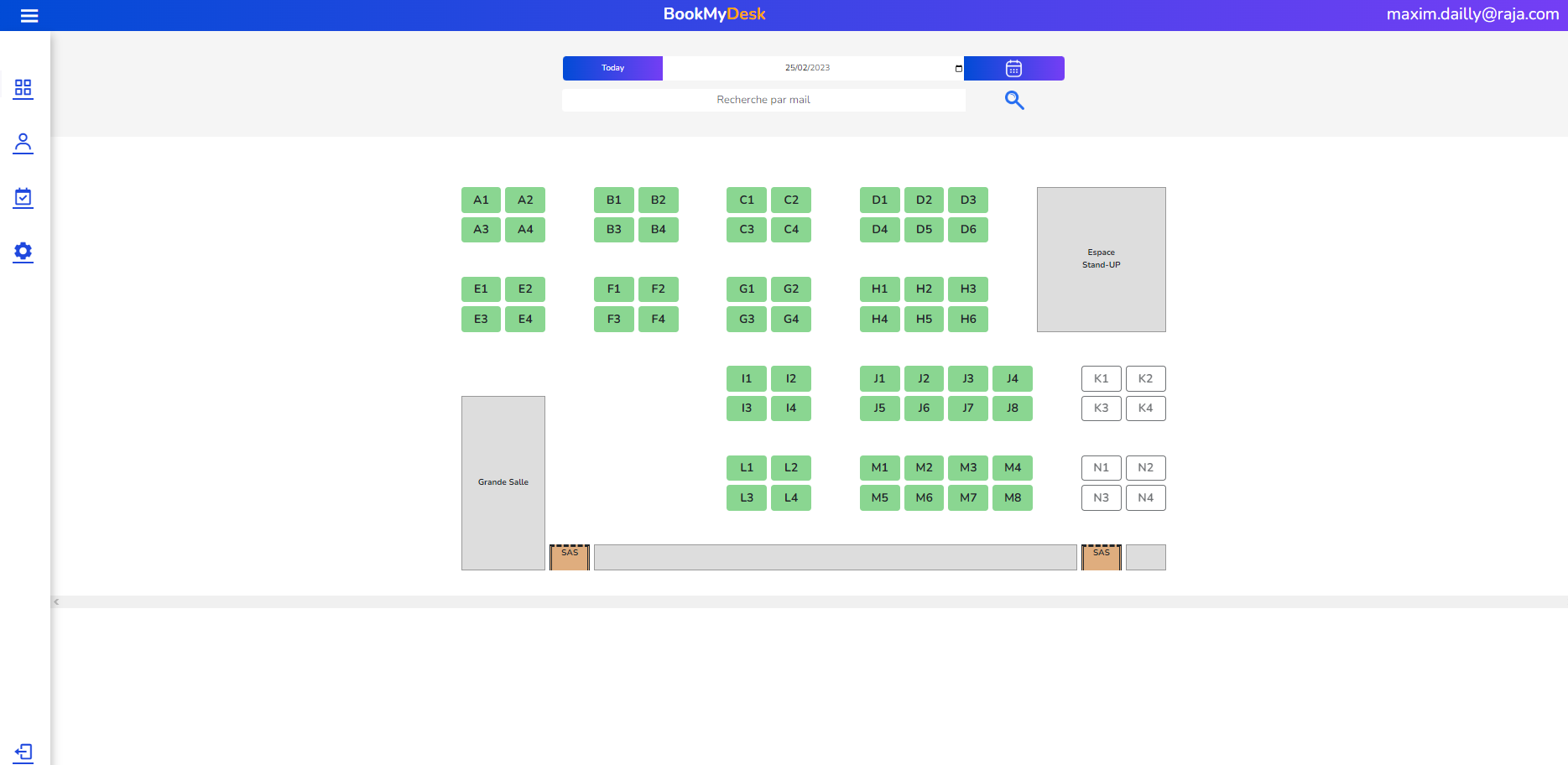


Figure 7 - Page d'accueil

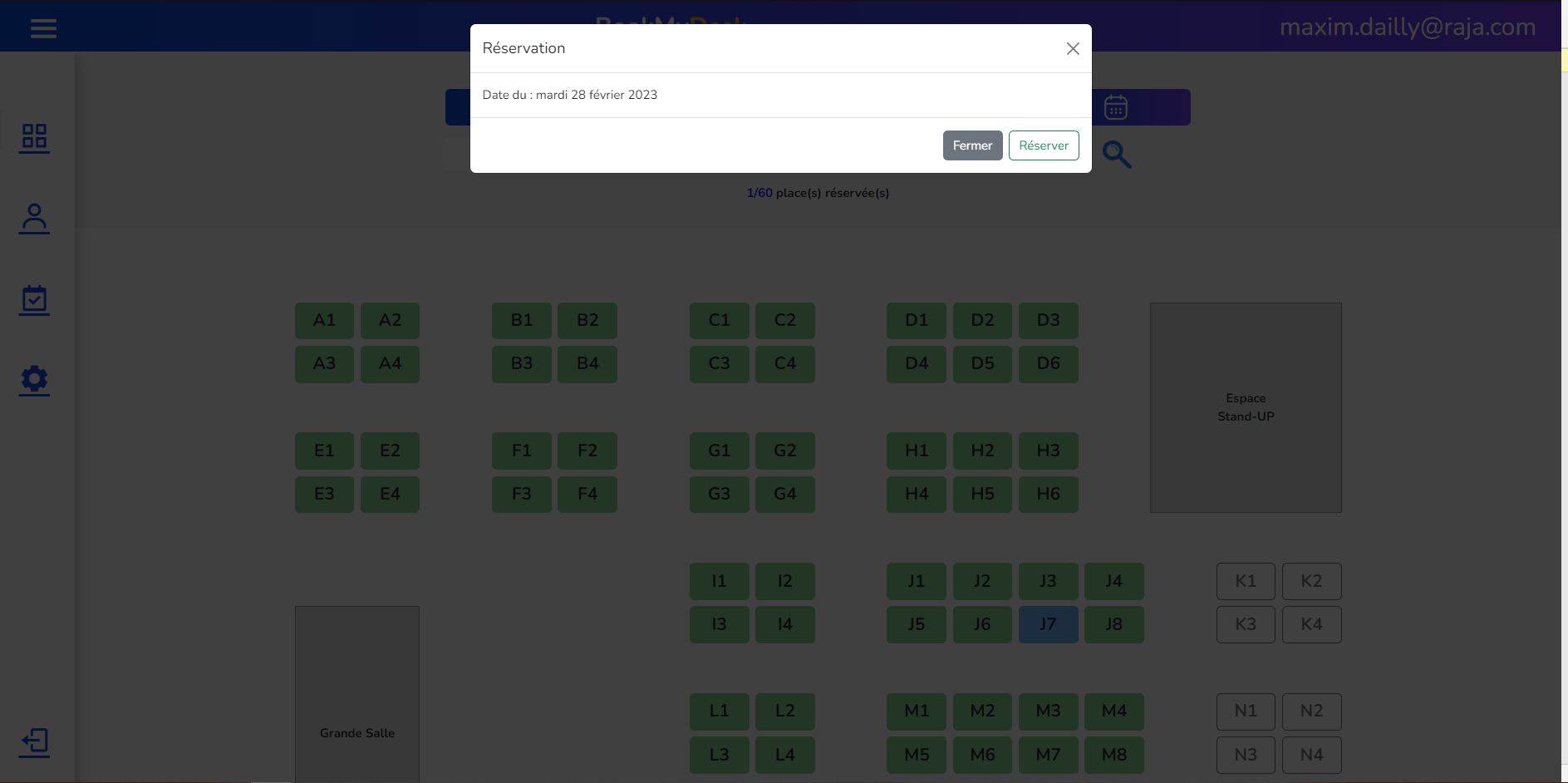


Figure 8 - Exemple de pop-up

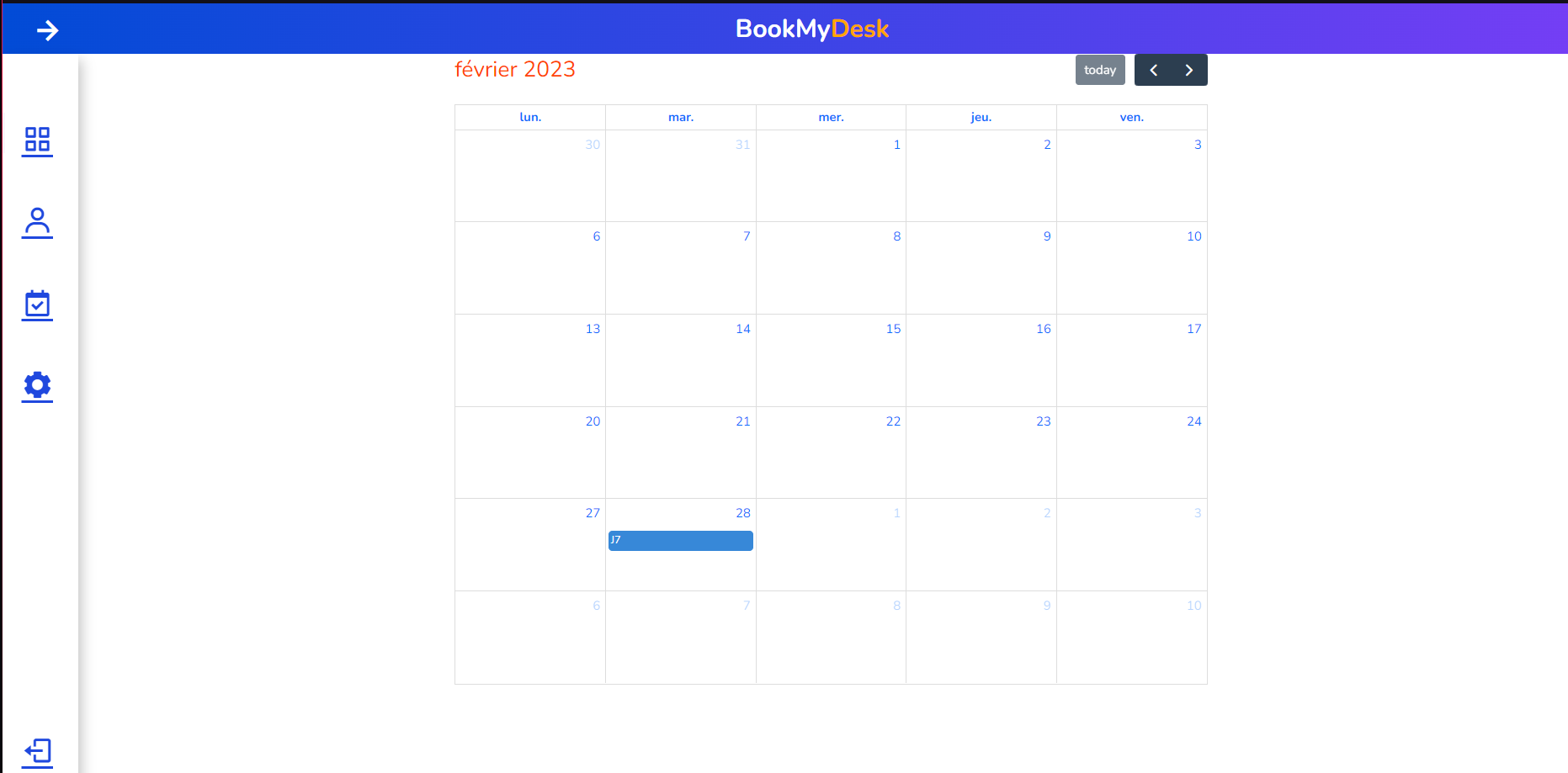


Figure 9 - Page mes Réservations

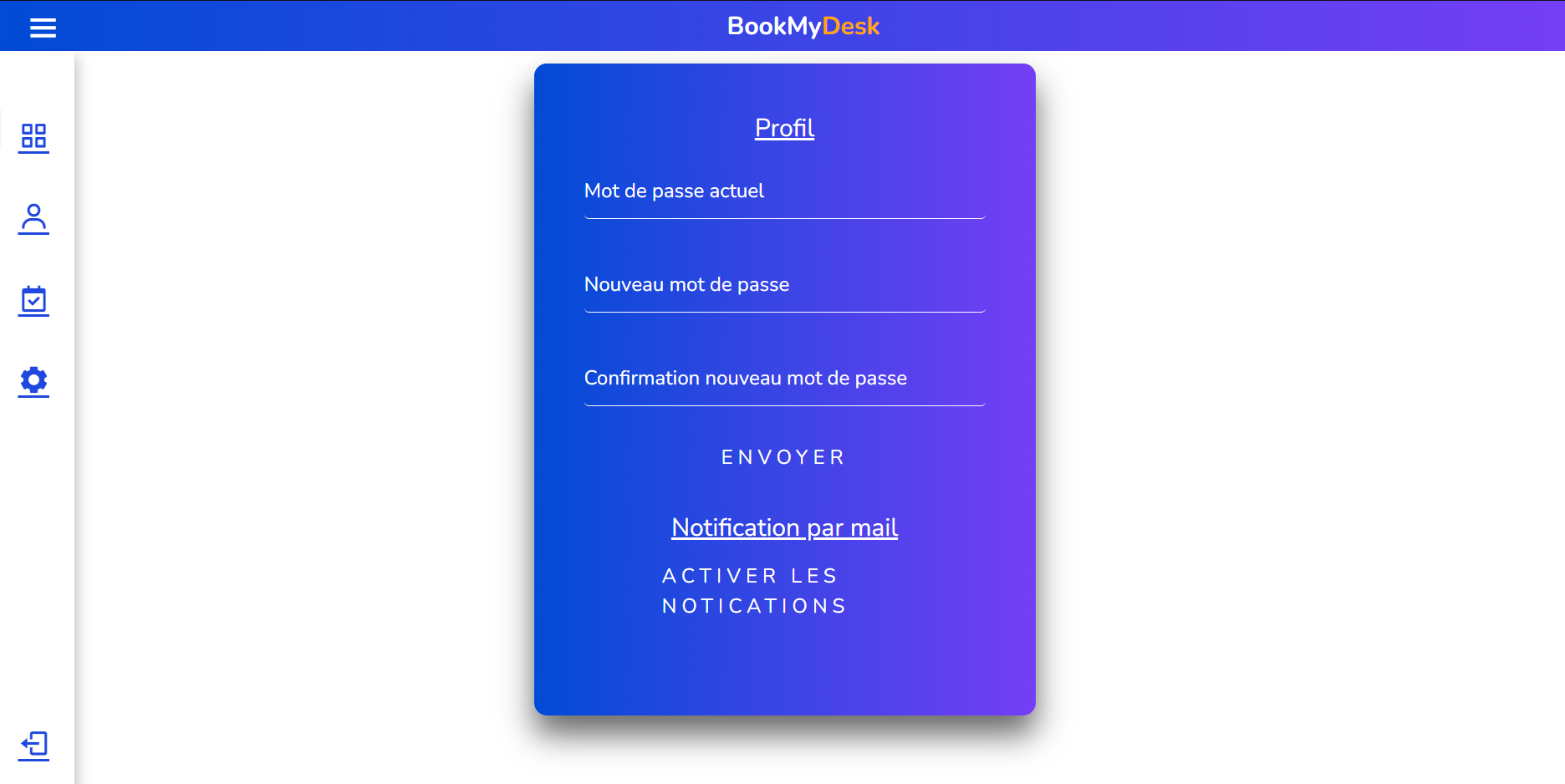


Figure 10 - Page mon Profil



Figure 11 - Page Administration des bureaux

# Conclusion

Pour conclure ce rapport, ce stage m’a beaucoup enrichi et m’a permis de découvrir en contexte réel le secteur informatique et plus précisément celui du développement. En participant à la vie de l’entreprise au quotidien et en interagissant avec les différents acteurs, j’ai pu appréhender les différents enjeux soulevés au sein du monde de l’entreprise.

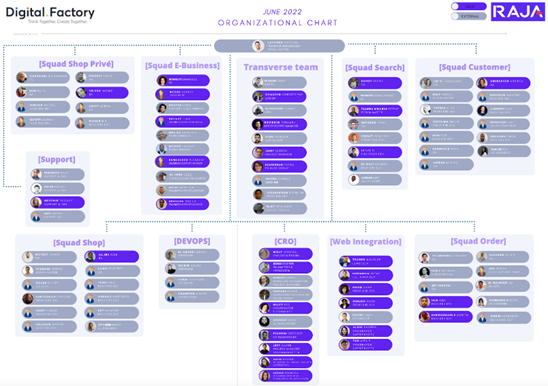
J’ai pu acquérir diverses compétences comme le développement en JavaScript, l’utilisation de NodeJs ainsi que le fonctionnement de MongoDB. Cela m’a aussi permis de comprendre que le développement est un élément fondamental à la croissance d’une entreprise. Mon intérêt de travailler dans une entreprise a augmenté, ainsi que celui de développer des solutions utiles pour les autres collaborateurs et pour ses clients.

L’informatique est un milieu vaste et fort intéressant, faire ce stage m’a permis de me rendre compte encore plus des enjeux des nouvelles technologies dans notre avenir. Fort de cette expérience et ayant désormais une idée plus claire de la réalité du métier de développeur, j’aimerais dorénavant continuer à m’orienter vers le développement et plus particulièrement vers l’Intelligence Artificielle, qui est un atout crucial entre les différentes entreprises.

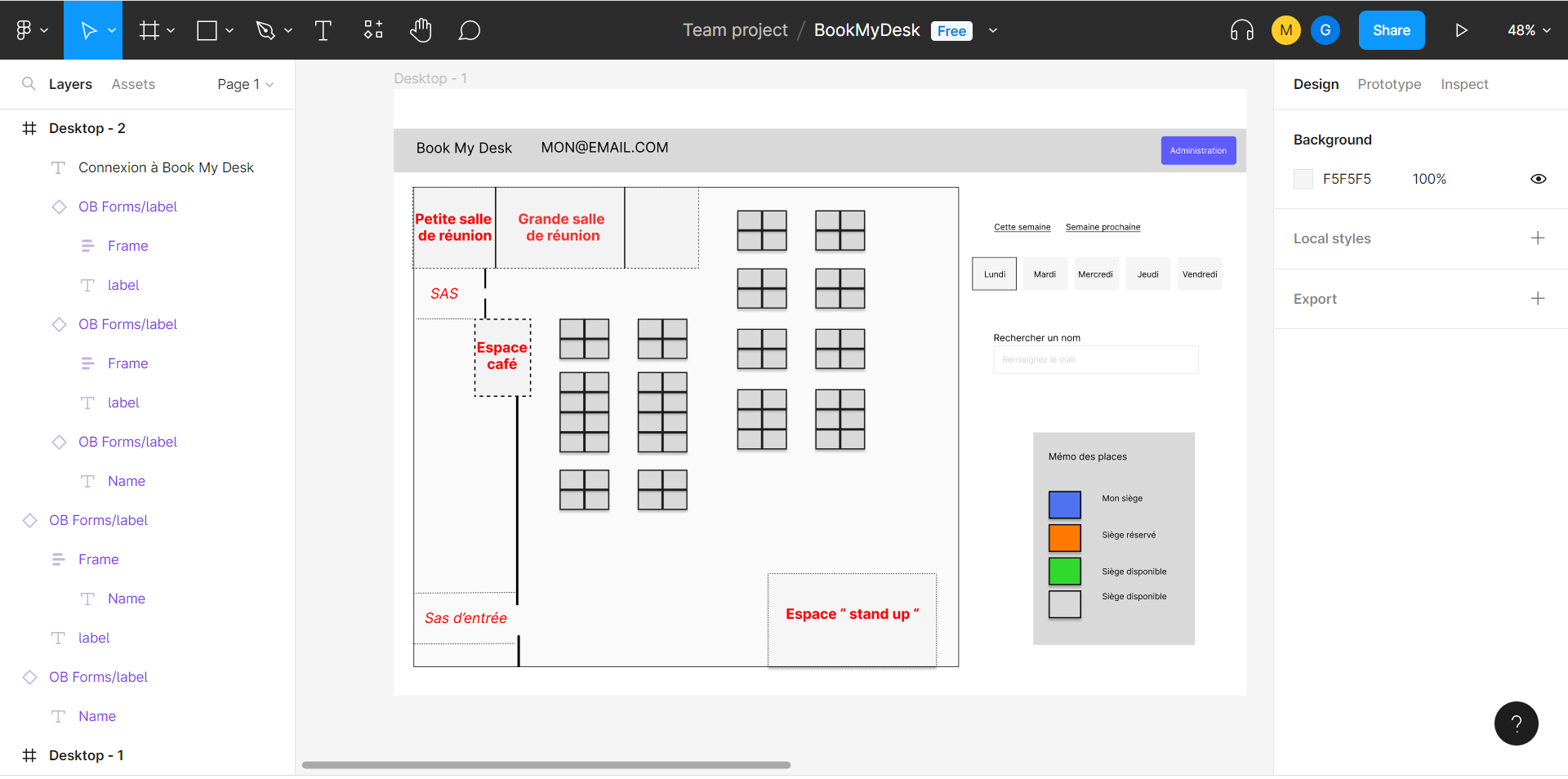
Je remercie Raja et mes tuteurs de m’avoir appris toutes ces différentes technologies et méthodes de travail.

# Annexes

Annexe 1*: Charte organisationnelle de la digital factory*

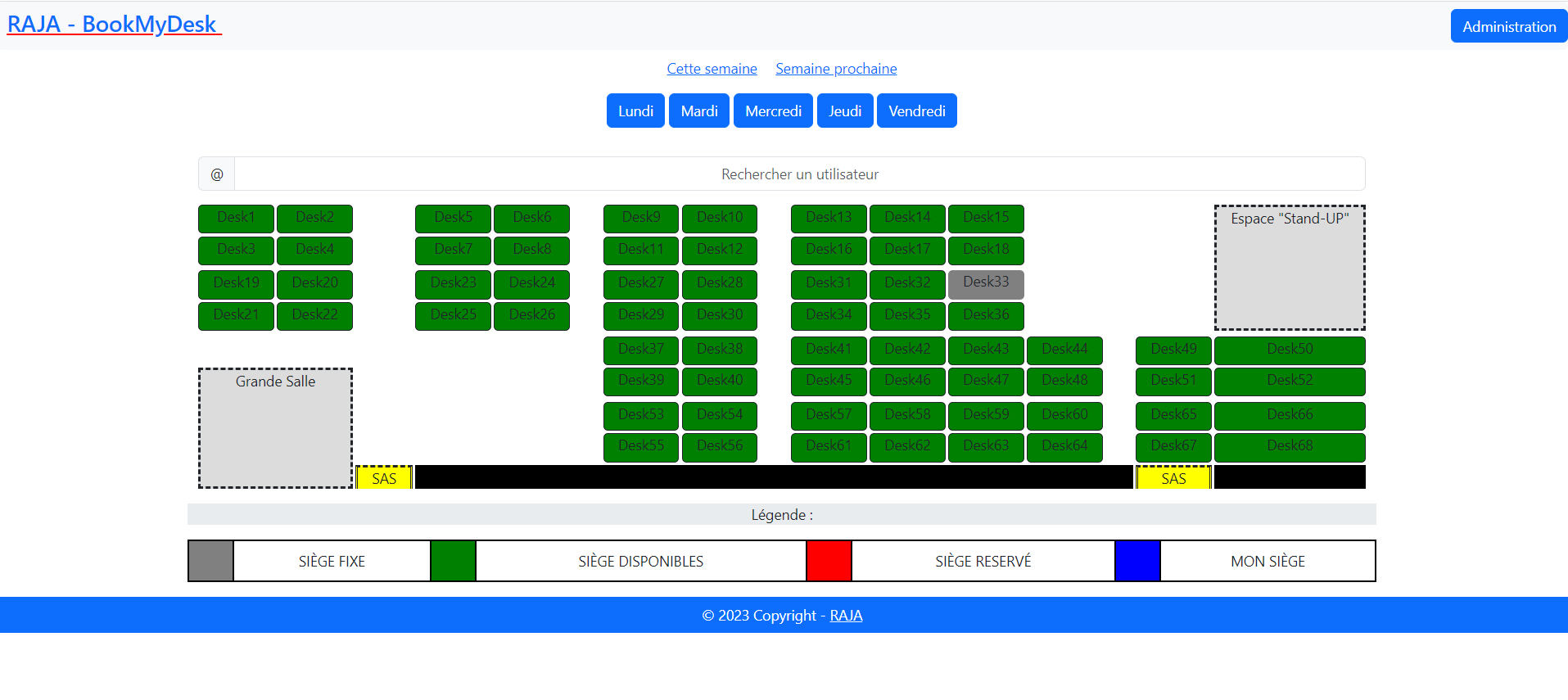


Annexe 2 *: Premier Design de BookMyDesk à l’aide de Figma*



Annexe 3 *: Evolution de la page d’acceuil*









Une image contenant texte, moniteur, calculatrice, capture d’écran

Description générée automatiquement