

# Cahier des Charges

---

## Application de Gestion des Réservations

Salle de Réunion TechSpace Solutions

### 1. Synopsis

#### Contexte

TechSpace Solutions est un espace de coworking accueillant une cinquantaine de collaborateurs répartis en plusieurs équipes. L'entreprise dispose d'une salle de réunion partagée d'une capacité de 12 personnes.

#### Problématique

Actuellement, les réservations se font de manière artisanale (post-its, emails, tableau blanc), ce qui génère :

- Des conflits de double réservation (3 incidents le mois dernier)
- Des créneaux vides par manque de visibilité
- Une perte de temps et d'efficacité pour les équipes

#### Solution proposée

Une application web simple et efficace permettant aux collaborateurs de :

- Visualiser en temps réel le planning de la salle
- Réserver, modifier ou annuler leurs créneaux
- Éviter les conflits de réservation

### 2. Cahier des Charges Front-end

#### Fonctionnalités principales

##### Authentification et gestion des utilisateurs

- Création de compte, connexion, déconnexion, consultation du profil

##### Visualisation du planning

- Affichage hebdomadaire (lundi-vendredi, 8h-19h)

##### Gestion des réservations

- Création, modification, annulation de réservation

## EPIC 1 : Authentification

### US01 - Création de compte

En tant que **collaborateur** de TechSpace Solutions

Je veux **créer un compte avec mon email et un mot de passe**

Afin de **pouvoir accéder à l'application**

#### Contraintes :

- Email valide (format RFC 5322) et unique
- Mot de passe : minimum 8 caractères (1 majuscule, 1 minuscule, 1 chiffre)
- Champs obligatoires : nom, prénom, email, mot de passe
- Messages de confirmation/erreur explicites

### US02 - Connexion

En tant que **collaborateur inscrit**

Je veux **me connecter avec mon email et mot de passe**

Afin **d' accéder au planning de réservation**

#### Contraintes :

- Vérification des identifiants en base de données
- Création d'une session utilisateur (durée : 24h)
- Redirection vers le planning après connexion réussie

### US03 - Déconnexion

En tant que **collaborateur connecté**

Je veux **me déconnecter de l'application**

Afin **de sécuriser mon accès**

#### Contraintes :

- Bouton accessible depuis toutes les pages
- Destruction de la session et redirection vers connexion

## US04 - Consultation du profil

*En tant que collaborateur connecté*

*Je veux consulter mon profil utilisateur*

*Afin de vérifier mes informations personnelles*

### Contraintes :

- Affichage du nom, prénom, email (lecture seule)

## EPIC 2 : Visualisation du planning

### US05 - Visualisation du planning hebdomadaire

*En tant que collaborateur connecté*

*Je veux voir le planning de la semaine en cours*

*Afin d'identifier les créneaux disponibles*

### Contraintes :

- Affichage uniquement des jours ouvrés (lundi-vendredi)
- Horaires : 8h00 à 19h00, créneaux par tranches d'1 heure
- Différenciation visuelle : libres (vert), occupés (rouge), passés (gris)
- Affichage du nom du réservant et de l'objet sur les créneaux occupés

### US06 - Navigation entre les semaines

*En tant que collaborateur consultant le planning*

*Je veux naviguer entre les semaines*

*Afin de planifier les réunions à l'avance*

### Contraintes :

- Boutons 'Semaine précédente' / 'Semaine suivante'
- Impossible de revenir sur une semaine entièrement passée

## EPIC 3 : Gestion des réservations

### US07 - Réservation d'un créneau

En tant que **collaborateur connecté**

Je veux **réserver un créneau disponible**

Afin de **bloquer la salle pour ma réunion**

#### Contraintes :

- Créneau disponible uniquement, durée minimum 1 heure
- Objet de la réunion obligatoire (max 255 caractères)
- Impossible de réserver dans le passé
- Vérification de disponibilité en temps réel
- Code HTTP 201 (Created) / 409 (Conflict)

### US08 - Prévention des doubles réservations

En tant que **système**

Je veux **empêcher les doubles réservations**

Afin **d' éviter les conflits**

#### Contraintes :

- Vérification atomique de disponibilité avant insertion
- Lock optimiste sur le créneau pendant la réservation
- Message d'erreur explicite et rafraîchissement automatique

### US09 - Modification de ma réservation

En tant que **collaborateur ayant une réservation**

Je veux **modifier l'objet de la réunion**

Afin de **corriger ou préciser les informations**

#### Contraintes :

- Modification uniquement de ses propres réservations

- Modification de l'objet uniquement (pas du créneau horaire)
- Impossible si le créneau est passé
- Code HTTP 200 (OK) / 403 (Forbidden)

## US10 - Annulation de ma réservation

En tant que **collaborateur** ayant une réservation

Je veux **annuler ma réservation**

Afin de **libérer le créneau**

### Contraintes :

- Annulation uniquement de ses propres réservations
- Confirmation avant suppression (modale)
- Créneau redevient immédiatement disponible

## US11 - Restriction des réservations passées

En tant que **système**

Je veux **empêcher la réservation de créneaux passés**

Afin de **maintenir la cohérence du planning**

### Contraintes :

- Créneaux passés grisés et non cliquables
- Vérification côté client ET serveur
- Code HTTP 400 (Bad Request)

## Stack Technique Front-end

### Technologies

- Framework : React
- Langage : TypeScript
- Build : Vite
- Styling : Tailwind CSS / ou autres

## Routes de l'application

Route	Nom	Accès	Description
/	Home	Public	Page d'accueil avec redirection
/register	Register	Public	Création de compte

/login	Login	Public	Connexion
/dashboard	Dashboard	Protégé	Planning hebdomadaire
/profile	Profile	Protégé	Profil utilisateur
*	NotFound	Public	Page 404

### Contraintes de sécurité

- Middleware d'authentification sur toutes les routes protégées
- Vérification du token JWT dans le localStorage
- Expiration automatique de session après 24h d'inactivité
- Validation côté client avant envoi au serveur
- Sanitisation des inputs pour prévenir les injections XSS
- Toutes les communications en HTTPS uniquement

## 3. Cahier des Charges Backend

### Pic de consommation :

- 9h00-10h00 et 14h00-15h00 (horaires de réservation les plus courants)

### Endpoints API

#### Authentification

Méthode	Endpoint	Description	Codes
POST	/api/auth/register	Création de compte	201, 400
POST	/api/auth/login	Connexion (retourne JWT)	200, 401
GET	/api/auth/logout	Déconnexion (optionnel)	200
GET	/api/users/profile	Consultation profil (JWT)	200, 401

#### Planning et réservations

Méthode	Endpoint	Description	Codes
GET	/api/planning/week?date=...	Planning hebdomadaire	200
POST	/api/reservations	Créer réservation	201, 409
PUT	/api/reservations/:id	Modifier réservation	200, 403
DELETE	/api/reservations/:id	Annuler réservation	200, 403

### Stack Technique Backend

#### Technologies Backend

- Framework : [Express.js](#)
- Base de données : MySQL (mysql2)
- Authentification : JWT + bcrypt

## Modèle de données (MySQL)

Table Users

Colonne	Type	Contraintes
id	INT	PRIMARY KEY, AUTO INCREMENT
nom	VARCHAR(255)	NOT NULL
prenom	VARCHAR(255)	NOT NULL
email	VARCHAR(255)	UNIQUE, NOT NULL
password	VARCHAR(255)	NOT NULL (hashed)
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP

Table Reservations

Colonne	Type	Contraintes
id	INT	PRIMARY KEY, AUTO INCREMENT
user_id	INT	FOREIGN KEY → Users(id)
date	DATE	NOT NULL
start_time	TIME	NOT NULL
end_time	TIME	NOT NULL
object	VARCHAR(255)	NOT NULL
created_at	TIMESTAMP	DEFAULT CURRENT_TIMESTAMP

## Contraintes de sécurité Backend

- Protection contre SQL injection (prepared statements)
- Validation et sanitisation des inputs contre XSS
- HTTPS obligatoire en production
- JWT secret sécurisé (variables d'environnement)
- Pas de logs de données sensibles
- Hachage des mots de passe avec bcrypt

## Documentation

- Documentation API : Swagger/OpenAPI
- README : instructions de setup et variables d'environnement